Threat Hunting

HW04

Name: 劉定睿 ID: M11409313

Date: 2025-10-13

目錄

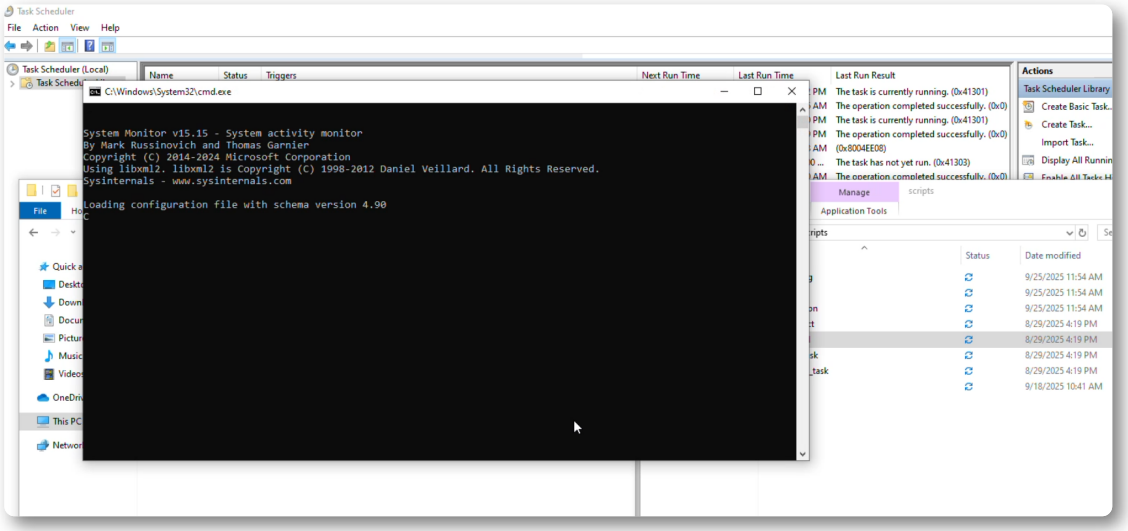
Part I

1. What workflow or process did the automated system demonstrate?

   1.1 The first video is show that threat hunter is going to extract the log on the computer.

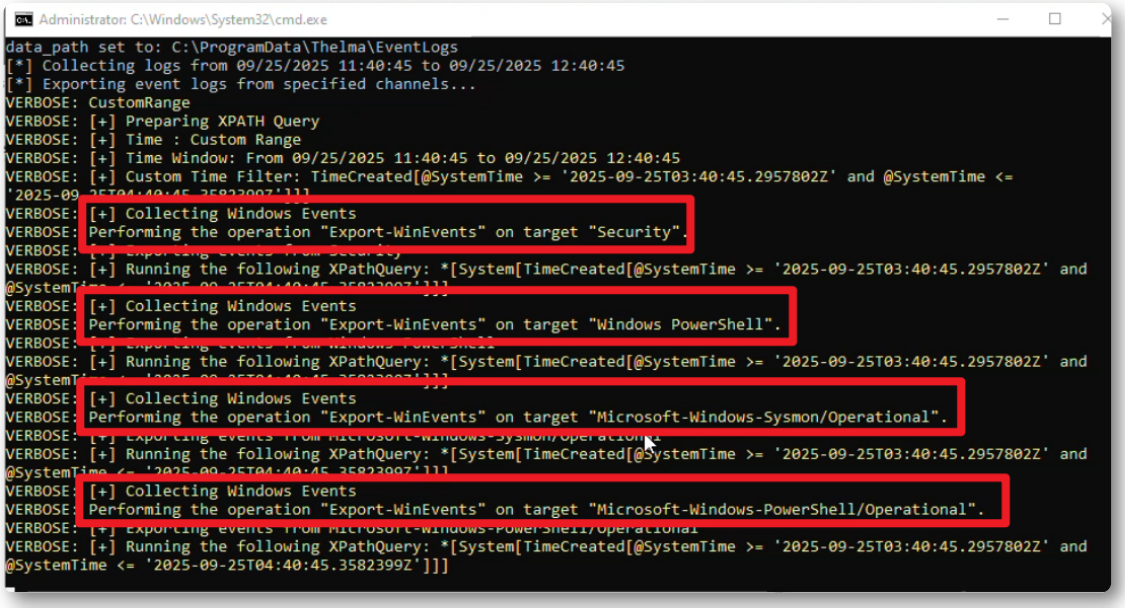
   - First step:

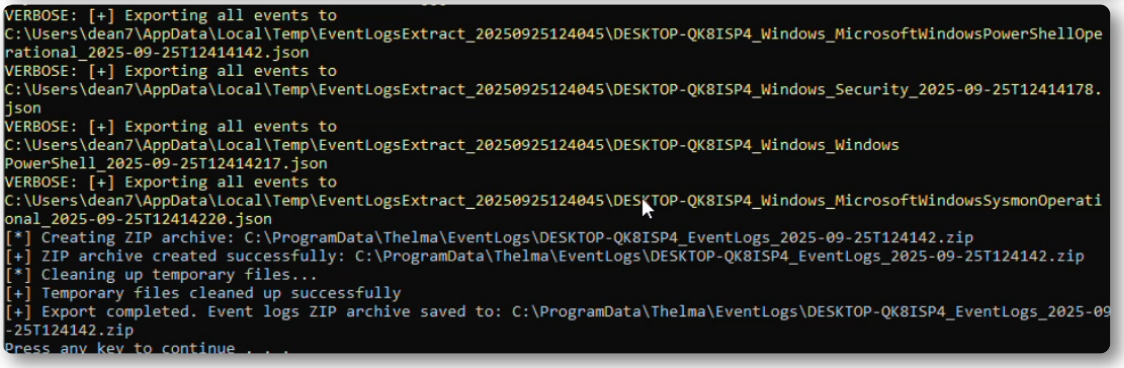
     Threat hunter try to run the specific script to install the software in the computer.

   - second step:

     Threat hunter export the log of `Security`, `Windows Powershell`, `Microsoft-Windows-Sysmon/Opertional` and `Microsoft-Windows-Powershell/Opertional` from computer. And also store the each log data to a zip file.

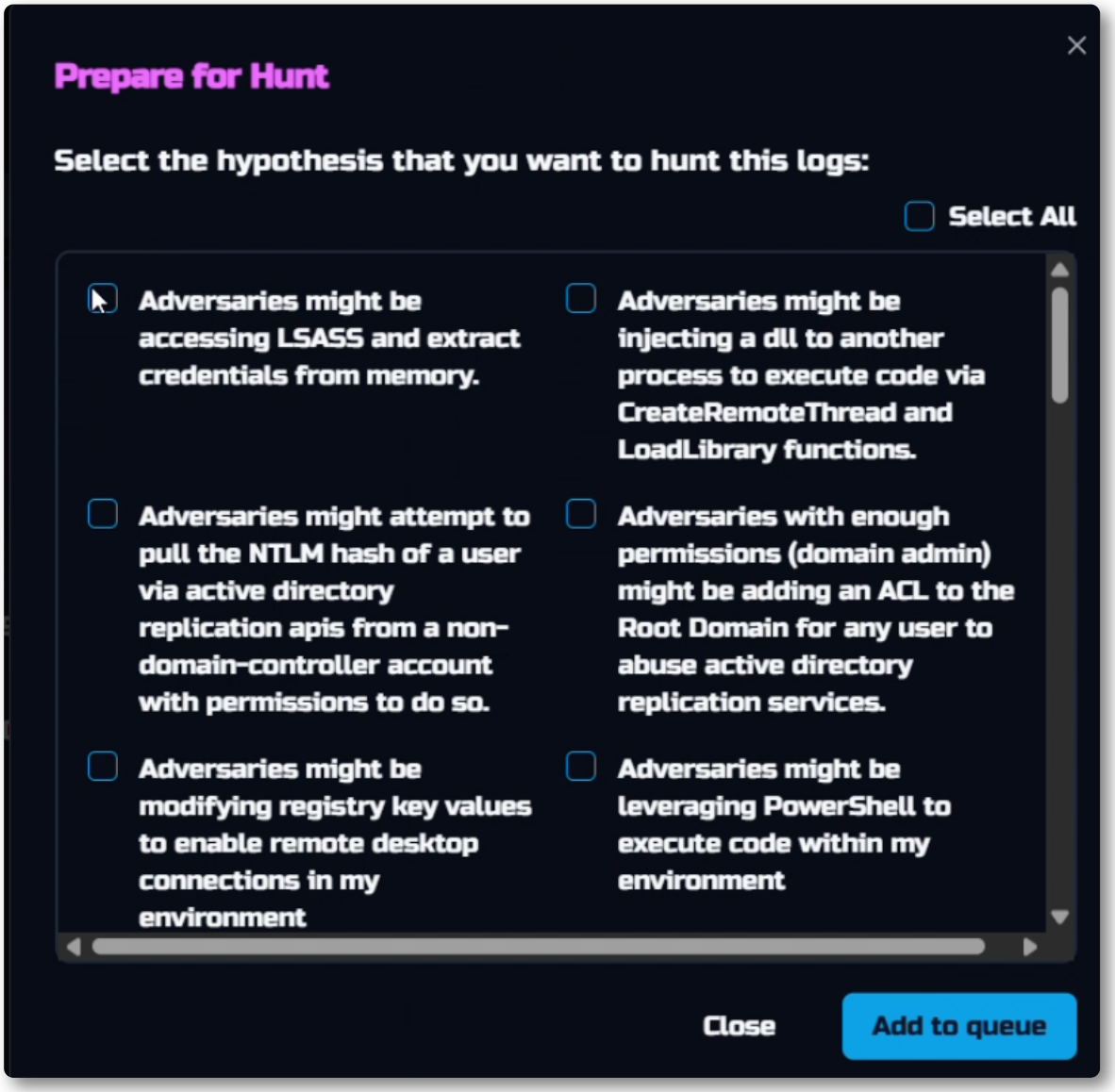




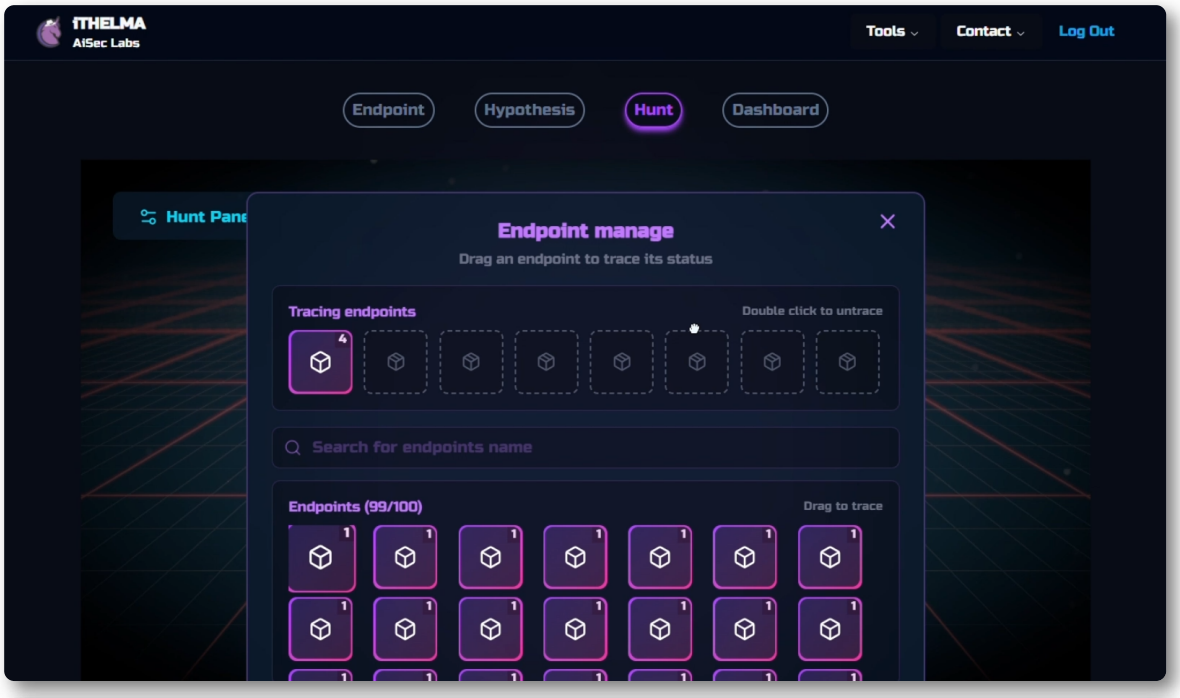
- Third step:
  Eventually threat hunter use `rm_task` script to remove the task which create by thelma.

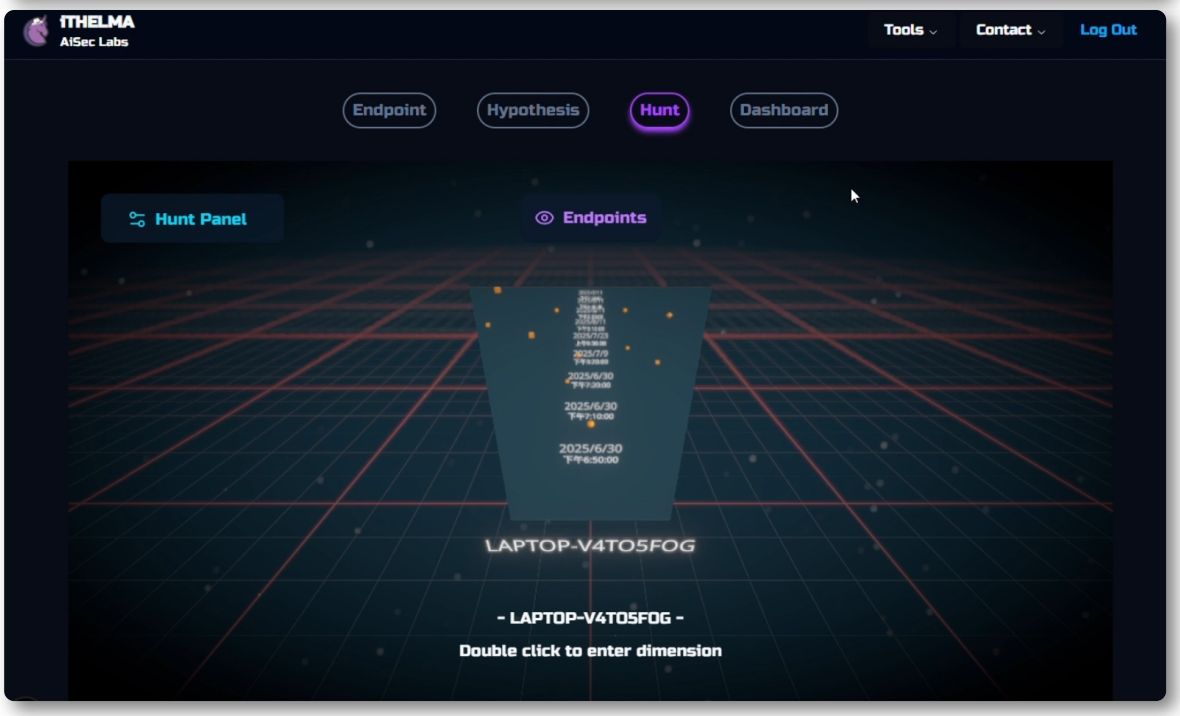
1.2 In the second video. Threat hunter is trying to import the zip file to thelma software and automatically hunting the suspicious threat.
Threat hunter can select the hypothesis threat. And software will searching the

pattern base on hunter selected.

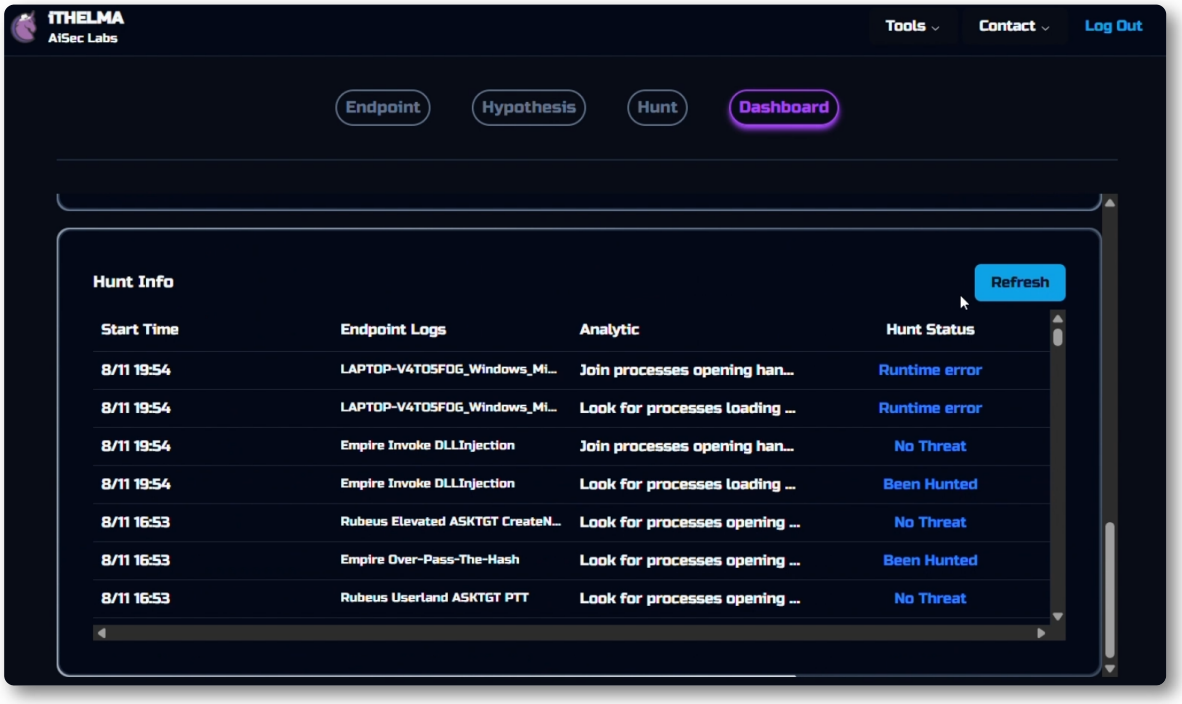


1.3 The third one video shows that thelema can classify different kind of functions. Threat hunter select the `Hunt` function. Thelma will process the log which import by before. After process the log. Thelma can list the timeline and endpoint. Threat hunter can choose their preference and investigate the case.


ITHELMA
AiSec Labs
Tools
Contact
Log Out
Endpoint
Hypothesis
Hunt
Dashboard
Hunt Pane
Endpoint manage
Drag an endpoint to trace its status
Tracing endpoints
Double click to untrace
Search for endpoints name
Endpoints (99/100)
Drag to trace


ITHELMA
AiSec Labs
Tools
Contact
Log Out
Endpoint
Hypothesis
Hunt
Dashboard
Hunt Panel
Endpoints
2025/6/30
2025/6/30
2025/6/30
LAPTOP-V4TO5FOG
- LAPTOP-V4TO5FOG -
Double click to enter dimension

Later, Thelma can output the whole information on the dashboard.



2. How did the system interpret the playbook logic and produce threat reports?

Threat Hunter Playbook expresses detection *logic* as executable, ATT&CK-mapped Jupyter notebooks (queries + code + expected outputs). A system (human analyst, SIEM/EDR, or automation runner) interprets that logic by executing the notebooks (or translating their queries into SIEM rules), correlating the notebook outputs with telemetry/datasets, enriching results, scoring/triaging, and finally exporting the findings into a threat report (notebook output → HTML/PDF/alert payload).

- Executive summary (threat, severity)
- Evidence & timeline (events, counts, timestamps)
- IOCs (files, hashes, domains, JA3s)
- TTP mapping (MITRE ATT&CK techniques observed)
- Recommended containment/remediation actions (isolate, credential reset, block IOCs)
- Confidence / artifacts to support triage (memory dump path, EDR snapshot id)

3. What aspects of the process still required human intervention or judgment?

Phase	Automated by System	Requires Human Judgment
Hypothesis generation	Partially (based on ATT&CK mapping)	Context choice, environment awareness
Data acquisition	Automated (log pulls, queries)	Schema validation, source reliability

Detection logic execution	Fully automated	
Result triage	Partially (basic scoring)	Interpret ambiguous findings
Reporting	Automated export	Narrative synthesis, risk assessment
Feedback / tuning	Basic metrics	Decide thresholds, adapt to ops context

4. How might such systems improve efficiency or consistency in real-world SOC operations?

Category	Efficiency Gain	Consistency Gain
Execution	Automates queries	Identical workflows across analysts
Decision process	Embedded scoring logic	Common escalation criteria
Documentation	Auto-generated reports	Uniform report structure
Knowledge	Central repository of hunts	Shared vocabulary and ATT&CK mapping

Part II

1. The conceptual and architectural foundations of the automation framework.

Leveraging Large Language Models (LLMs) to automate the traditionally labor-intensive process of Cyber Threat Hunting. The foundational concept is that threat hunting knowledge can be extracted from human-authored playbooks and converted by an AI agent into executable detection scripts (Hunt Scripts).

2. The advantages and limitations of automated threat hunting.

- Advantages
    - Overcoming Environmental Heterogeneity:
      Traditional threat hunting tools and playbooks are often difficult to apply universally due to incompatibilities in different environments (such as varying log formats and query languages)LLM-based frameworks can dynamically generate scripts adapted to specific environments, eliminating the need for cumbersome data standardization.

    - Improving Efficiency and Scale:
      Automation significantly reduces the extensive time and effort analysts spend on manually writing and testing queries. iThelma's co-occurrence scheduling strategy was proven to detect more threats faster than relying solely on LLM-based predictions.

    - Knowledge Generalization and Recovery:
      Even with incomplete playbooks, iThelma can autonomously generate and successfully recover 22 out of 26 removed scripts based on threat descriptions, hunting tips, and related scripts This demonstrates its ability to generalize from partial knowledge.

    - Continuous Self-Improvement:
      iThelma's self-adaptation mechanism allows it to learn from both successful and failed experiences, continuously optimizing its script generation and threat analysis capabilities.

- Limitations
    - Heavy Reliance on Input Quality:
      The quality of scripts generated by the LLM is highly dependent on the quality of

the input playbooks and hunting tips. If the provided tips are insufficient or vague, the accuracy of the generated scripts drops significantly . iThelma also found that script recovery tasks were prone to failure when dealing with playbooks containing poor information.

- Unreliability of LLM Reasoning Capabilities: The evaluation of Thelma showed that its LLM B had extremely low accuracy (around 10%) in predicting the next threat, indicating that relying solely on an LLM for complex causal reasoning is unreliable. This is why iThelma shifted to using a data-driven co-occurrence matrix instead.

- Semantic Correctness Challenge:
  Although LLMs excel at generating syntactically correct code, ensuring its **semantics (logic)** fully aligns with the hunting intent remains a challenge, especially when procedural details are missing.

- Limitations Against Emerging Threats:
  The framework's knowledge is derived from existing playbooks. Its detection capability may be limited when faced with entirely new threats for which no prior knowledge is available for reference.

3. The changing role of human hunters in an increasingly automated SOC.

- From Script Writing to Knowledge Curation:
  An analyst's primary job is no longer to manually write and execute every query, but rather to provide and manage high-quality, structured threat hunting playbooks. They act as "teachers" for the AI agent, responsible for supplying high-quality "training materials."

- From Manual Operation to Exception Handling:
  Analysts monitor the agent's execution status via a GUI. When the agent encounters a problem it cannot solve—such as a script that still fails after multiple auto-debugging attempts or when consensus cannot be reached among the detection results of several scripts—the system flags it for human review. Human expertise is then applied to handle edge cases that the machine cannot resolve.

- Focusing on Higher-Level Strategy:
  With the tedious "last-mile delivery" tasks automated, human analysts can devote more energy to more strategic work, such as analyzing macro-level attack trends, researching new attack techniques, and designing more creative hunting strategies.

- Trainers in Human-Machine Collaboration:
  The future development path clearly points toward human-in-the-loop refinement and reinforcement learning based on analyst feedback. Human hunters will play an active role in continuously improving the AI agent's performance through this feedback.

4. Your critical reflection on whether full autonomy is desirable or dangerous in this context.

The motivation for pursuing autonomy is clear and rational. Faced with increasingly complex and rapid cyberattacks, human speed and scale have become a bottleneck. An autonomous system capable of learning, adapting, and executing hunting tasks 24/7 can theoretically provide defensive coverage and reaction speeds far exceeding those of human teams.

- The Danger
  - Magnification of Flawed Decisions:
    The evaluation of Thelma showed that a flawed reasoning core (LLM B) would make a large number of incorrect judgments. If a fully autonomous system were to take action based on such flawed reasoning (for example, dedicating significant computational resources in the wrong direction or, worse, executing an incorrect response action), the consequences could be catastrophic.

  - Risks from Lack of Validation:
    Thelma's initial design lacked a robust script validation mechanism. Allowing an autonomous agent to execute unverified code in a production environment is extremely dangerous. A minor error in syntax or logic could lead to system crashes, data corruption, or a flood of false positives.

  - Vulnerability to Adversarial Attacks:
    A fully LLM-reliant autonomous system could become a target for attackers. Through methods like "Prompt Injection" or data poisoning, an attacker could potentially induce the autonomous agent to generate malicious scripts or ignore genuine threats.

  - Lack of Common Sense and Contextual Understanding:
    Current AI is still vulnerable when handling ambiguity, incomplete information, and situations requiring deep domain knowledge. In critical decision-making moments, complete reliance on a machine could lead to disastrous misjudgments.