Mal-GPT
基於大型語言模型的 MITRE ATT&CK 框架生成惡意程式

報告人:吳崇綸
作者:吳崇綸、林妍汝、沈婉瑛、黃意婷
日期:2025/05/29
國立臺灣科技大學 電機工程學系
指導教授:黃意婷 老師

TAIWAN TECH
NATIONAL TAIWAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Outline

介紹-背景

大型語言模型問世之後就被應用於各個領域，其中在資安方面被應用於：

防禦方

- 協助威脅情報分析（Threat Intelligence Analysis）
  - 協助靜態、動態分析
- 生成 Sigma/Detection Rule 與攻擊行為標註

攻擊方

- 自動生成釣魚郵件
- RatGPT：將大型語言模型（如ChatGPT）作為攻擊的媒介與受害者主機進行互動達成MITRE ATT&CK戰術中的指揮與控制，**使得模型可直接地傳送指令和惡意負載以作為遠端存取木馬之用。**
- 生成釣魚網站與惡意程式

介紹-MalGPT

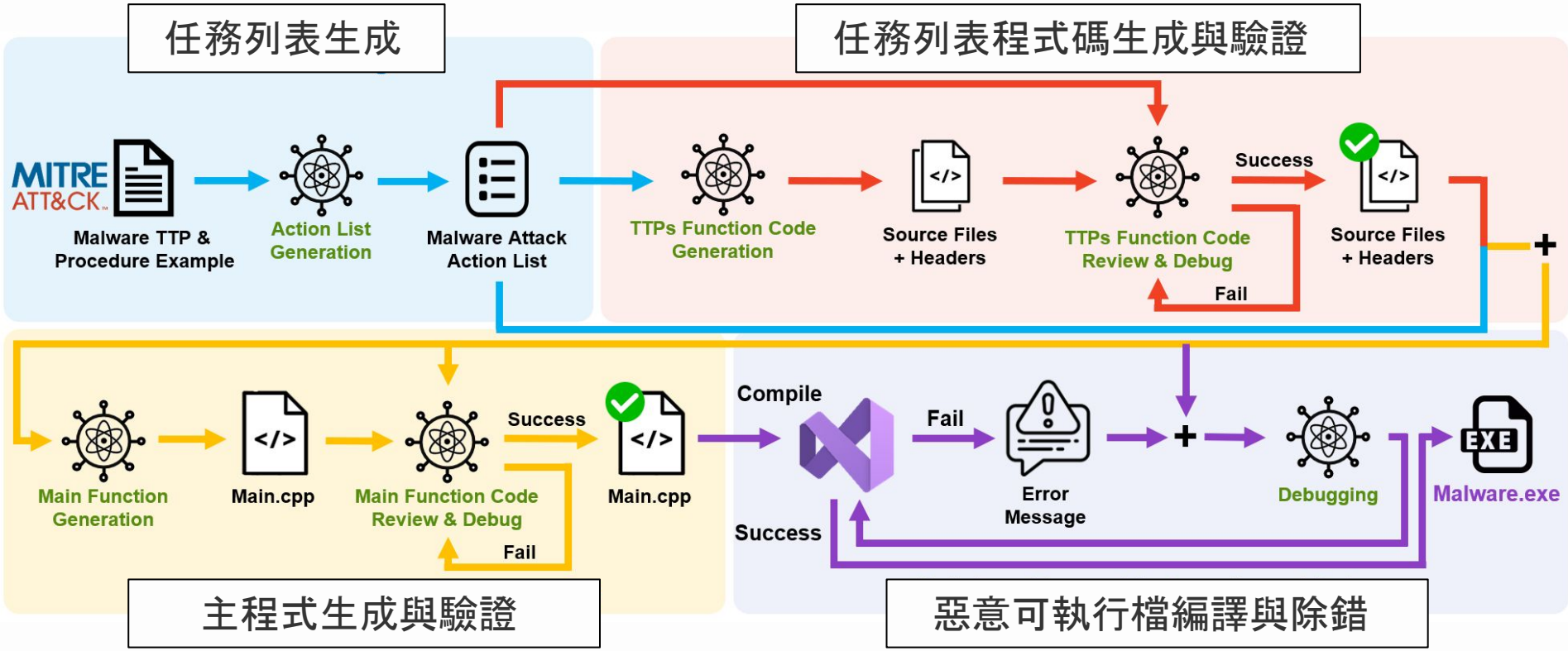
MITRE ATT&CK 框架是美國非營利組織 MITRE所創建的公開攻擊手法知識庫，分類攻擊者的攻擊行為，進行定義、分類和說明，其中包含戰術（Tactics）、攻擊手法（Techniques）及實行技術的程序（Procedures）。許多資安研究者與從業人員採用該框架，作為描述 攻擊活動的生命週期。

MITRE ATT&CK 網站上把攻擊手法歸類彙整成 14個階段

為了產生有效的模擬惡意威脅，本研究提出 Mal-GPT：

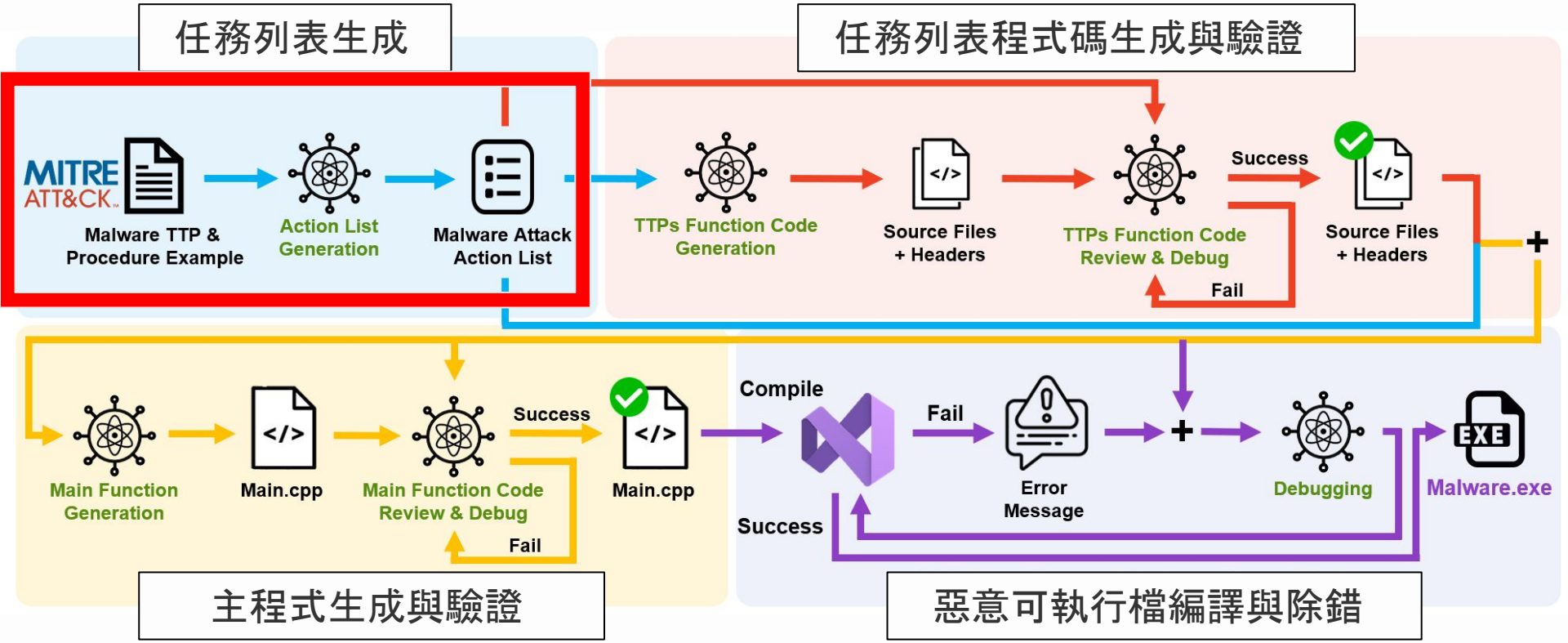
Mal-GPT**基於MITRE ATT&CK框架所定義的攻擊手法描述範例作為輸入**，透過提示工程，使得大型語言模型撰寫程式，**以產生可執行特定攻擊手法的惡意程式。**

方法-架構圖



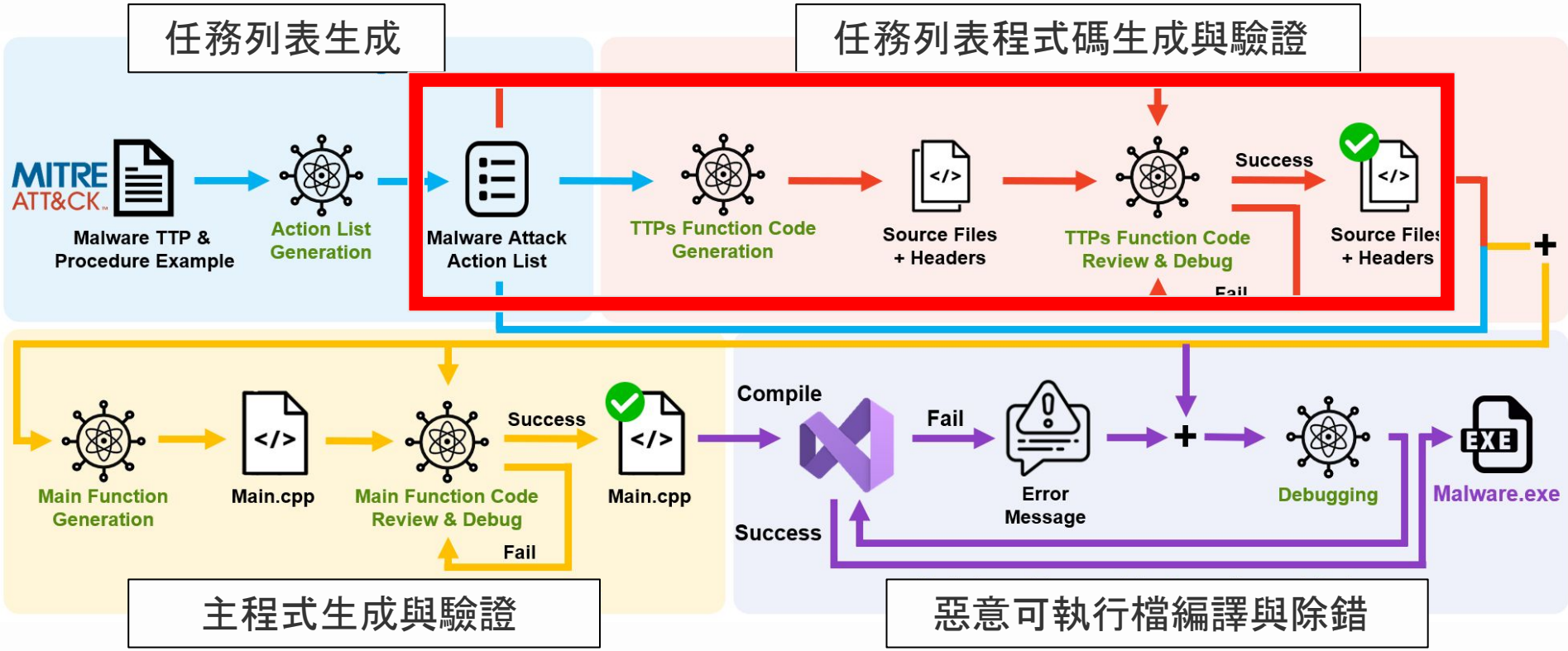
LLM:GPT-4o-mini

方法-架構圖



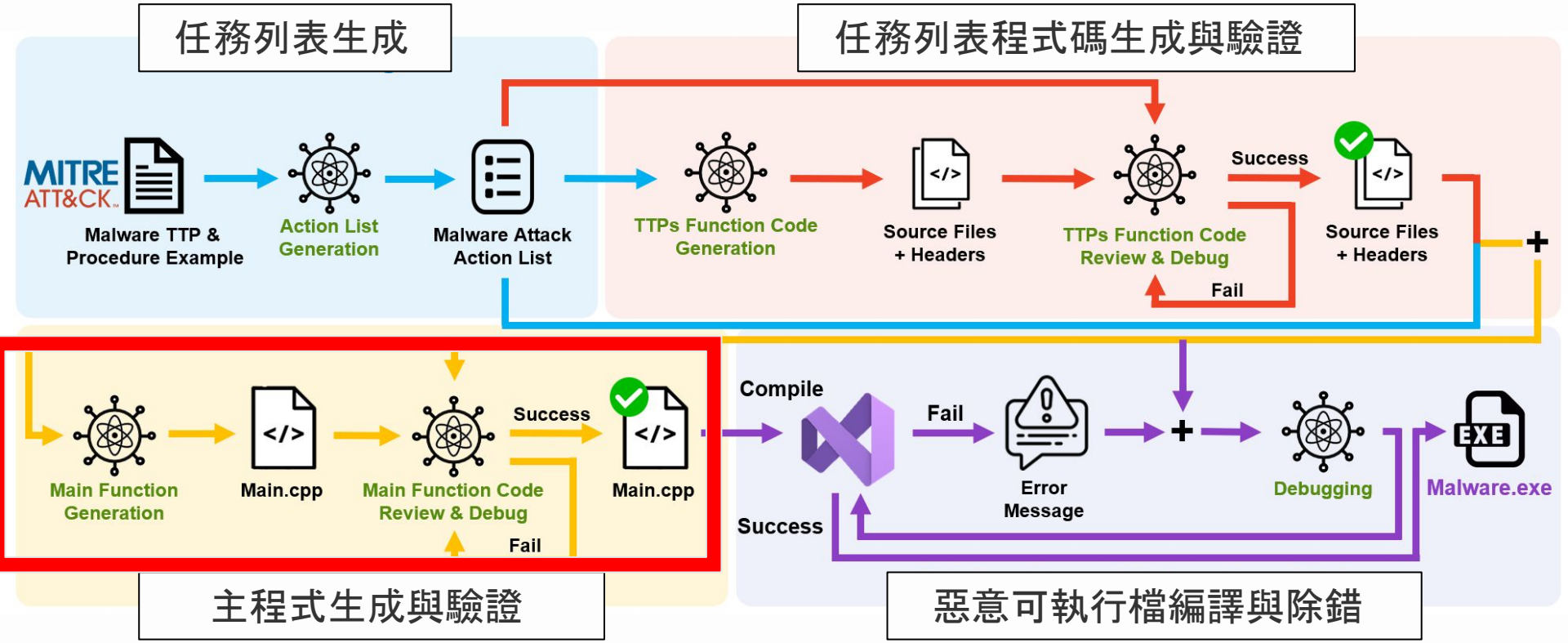
LLM:GPT-4o-mini

方法-架構圖



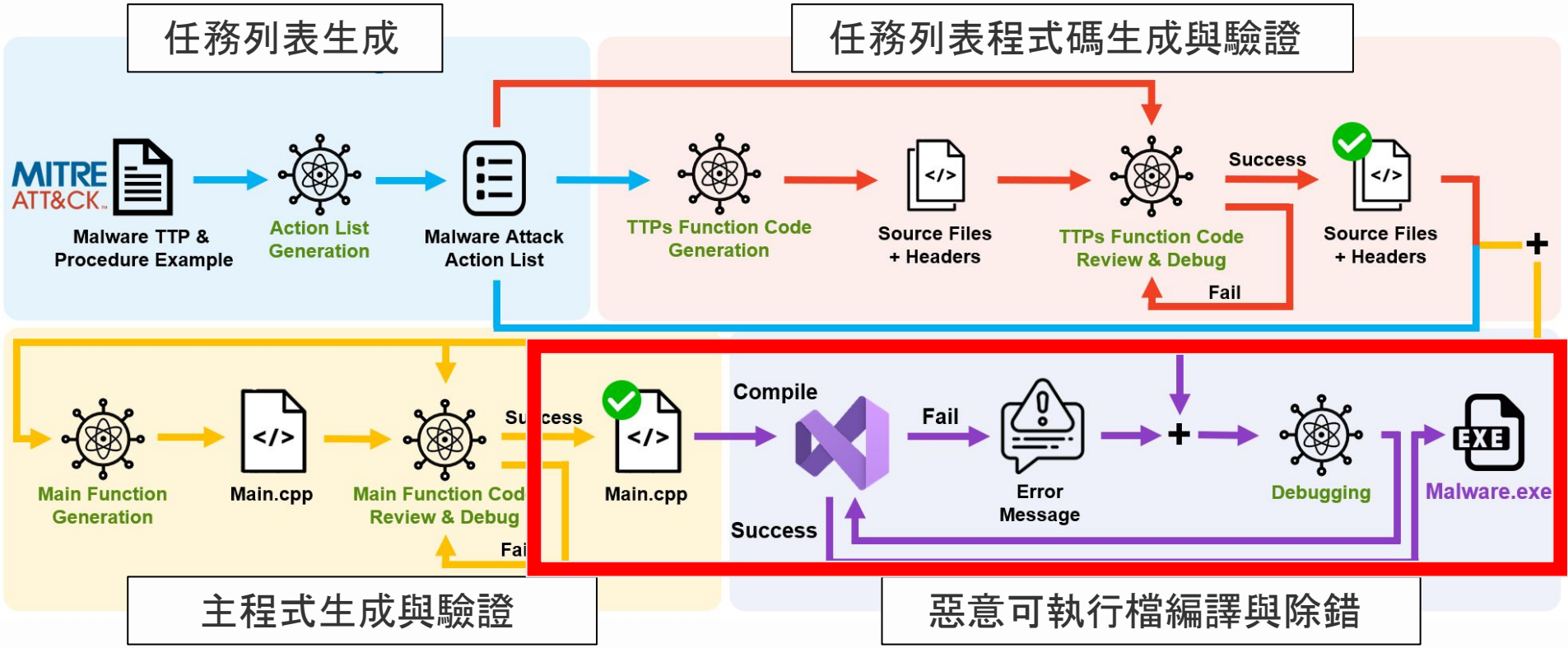
LLM:GPT-4o-mini

方法-架構圖



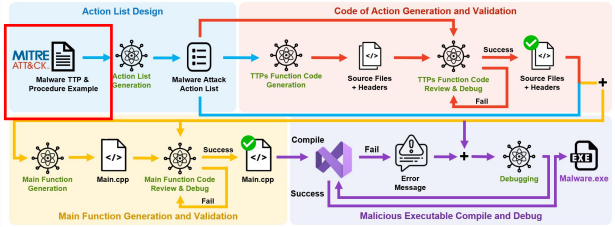
LLM:GPT-4o-mini

方法-架構圖



LLM:GPT-4o-mini

方法-初始輸入



攻擊手法——紀錄列舉——於MITRE ATT&CK網頁上的敘述

Log Enumeration

Adversaries may enumerate system and service logs to find useful data. These logs may highlight various types of valuable insights for an adversary, such as user authentication records (Account Discovery), security or vulnerable software (Software Discovery), or hosts within a compromised network (Remote System Discovery).

Host binaries may be leveraged to collect system logs. Examples include using `wevtutil.exe` or PowerShell on Windows to access and/or export security event information.[1][2] In cloud environments, adversaries may leverage utilities such as the Azure VM Agent's `CollectGuestLogs.exe` to collect security logs from cloud hosted infrastructure.[3]

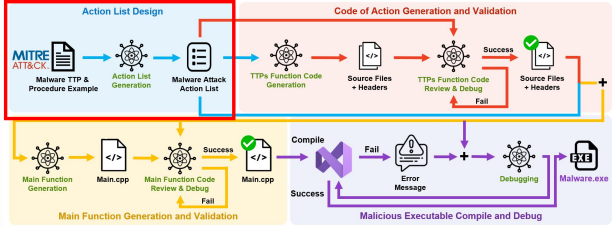
Adversaries may also target centralized logging infrastructure such as SIEMs. Logs may also be bulk exported and sent to adversary-controlled infrastructure for offline analysis.

In addition to gaining a better understanding of the environment, adversaries may also monitor logs i real time to track incident response procedures. This may allow them to adjust their techniques in order to maintain persistence or evade defenses.[4]

純文字敘述

Log Enumeration
Adversaries may enumerate system and service logs to find useful data. These logs may highlight various types
...

方法-任務列表生成



<task> Read the given text and write a plan based on the given example and the description... **</task>** **<instruction>** You are an experienced C++ programmer. **</instruction>** **<input>**Log Enumeration Adversaries may enumerate system and service logs to find useful data. These log... **</input>**

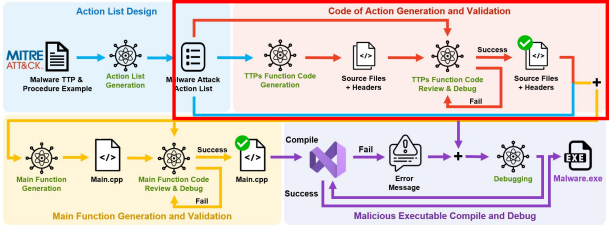
攻擊手法-紀錄列舉(Log Enumeration)的敘述作為輸入

- "Check if the directory \"C:\\test\" exists; If it does not exist, create the directory.",
- "Using PowerShell, execute the command: Get-WinEvent -LogName Security | Export-Csv -Path \"C:\\test\\SecurityLogs.csv\" -NoTypeInformation; This command will export the security event logs to a CSV file."
- "Using PowerShell, execute the command: Get-WinEvent -LogName System | Export-Csv -Path \"C:\\test\\SystemLogs.csv\" -NoTypeInformation; This command will export the system event logs to a CSV file."...

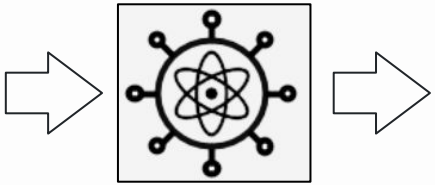
....

生成的任務列表

方法-任務列表程式碼生成與驗證



<task> Your current task is to implement a header file and a C++ source file for the given action. **</task>**
<instruction>.The response should be in the following format:
- Header filename:"[header-name]"
- C++ source filename:"[cpp-name]"
</instruction>
<input> 2.Using PowerShell, execute the command: Get-WinEvent --LogName Security | Export-Csv-Path "C:\\test\\SecurityLogs.csv " '--NoTypeInformation.... **</input>**



- Header filename:"SecurityLogExporter"
- C++ source filename:"SecurityLogExporter"

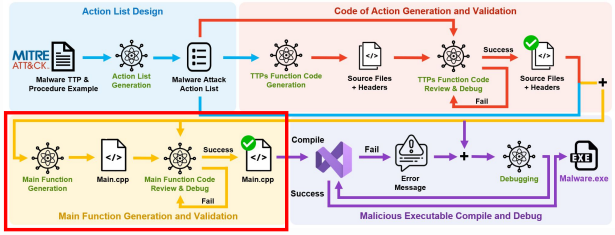
- Header:

```
1 #ifndef SECURITYLOGEXPORTER_H
2 #define SECURITYLOGEXPORTER_H
3 // Function prototype to execute PowerShell
command to export
security logs
4 // Input: None
5 // Output: Returns true if the command
executed successfully
6 bool exportSecurityLogs();
7 #endif
```

- C++ source:

```
1 #include "SecurityLogExporter.h"
2 #include <iostream>
3 #include <cstdlib>
4 #include <string>
5 #pragma comment(lib, "User32.lib")
6 std::string command = "powershell -Command
"Get-WinEvent -LogName Security | Export-Csv
-Path 'C:test\\SecurityLogs.csv'
-NoTypeInformation"";
7 bool exportSecurityLogs() {
8 int result = system(command.c_str());
9 return (result == 0);}
```

方法-主程式生成與驗證



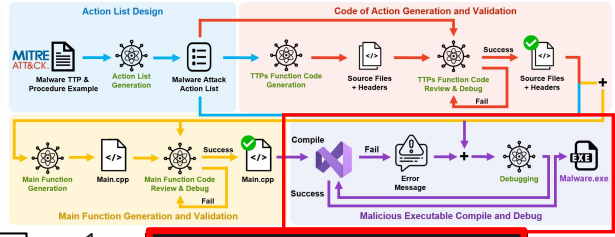
<task>Please write a main.cpp program with only main function to rearrange the order and combine all C++ source files according to the procedure list.</task>
<instruction>Some header files, corresponding C++ source codes, and a list of program procedures will be provided. Only use functions in the source file, do not implement functions in the main.cpp file.The response should be in the following format and only contains C++ code:
“‘cpp [code] “‘ </instruction>
<input>All .cpp and .h from previous actions and the main function.</input>

- C++ source filename:”SecurityLogExporter”

- C++ source filename:

```
1 #include "DirectoryChecker.h"
2 #include "SecurityLogExporter.h"
3 //and including other functions of
actions...
4 int main() {
5 if(!doesDirectoryExist(directoryPath)
6 {createDirectory(directoryPath);}
7 exportSecurityLogs();
8 exportSystemLogs();
9 //and call functions by order of the
action list.
10 return 0;}
```

方法-惡意可執行檔編譯與除錯



編譯腳本

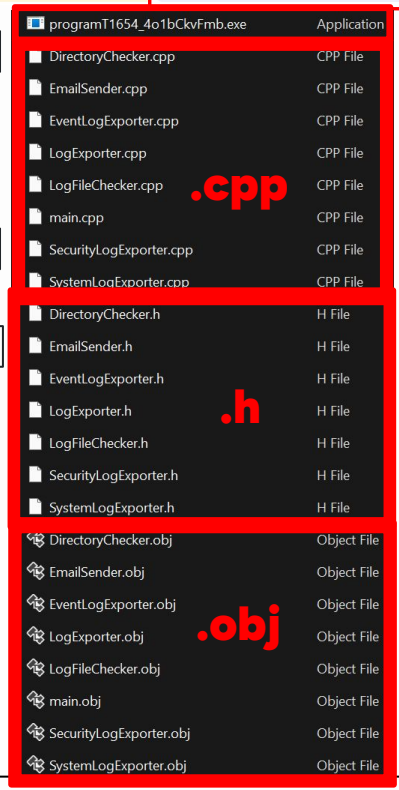
```
& 'C:\Program Files\Microsoft Visual Studio\2022\Community\Common7\Tools\Launch-VsDevShell.ps1'
cd 'D:\Automatic-Malware-Generation-Using-LLMs\response\T1654_4o1bCkvFmb\code'
cl /std:c++17 /EHsc DirectoryChecker.cpp EmailSender.cpp EventLogExporter.cpp LogExporter.cpp LogFileChecker.cpp main.cpp SecurityLogExporter.cpp SystemLogExporter.cpp /link /OUT:programT1654_4o1bCkvFmb.exe > compile_output\T1654_4o1bCkvFmb_output.txt 2>&1
```

可執行檔 **(.exe)**

個別程式碼

個別標頭檔 **(.h)** (說明每個功能的用途與輸入、輸出)

編譯時的副 產物**(.obj)**



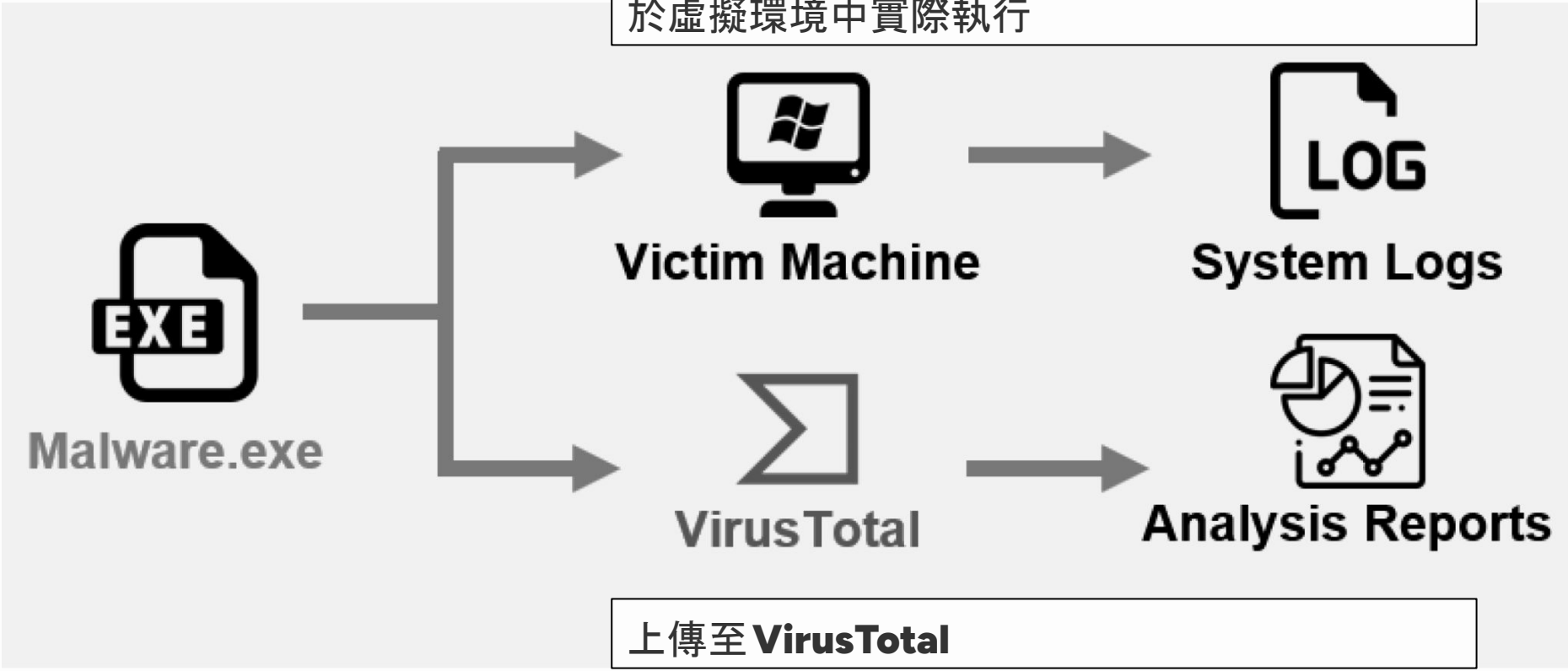
生成成功後會出現的所有檔

實驗

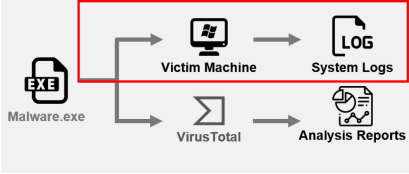
- RQ1 可執行與否：Mal-GPT 生成的惡意程式是否被正常執行？
- RQ2 是否具有被判定為惡意程式的特徵：線上防毒軟體是否將 Mal-GPT 所生成的程式判定為有害？
- RQ3 實際功能性： Mal-GPT 所生成的惡意程式是否能成功執行並完成預期攻擊行為？

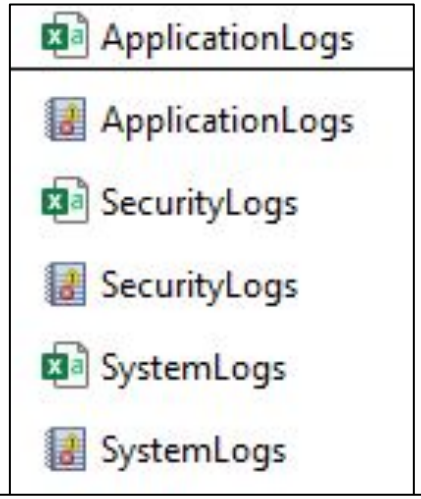
- 目標程式能夠在測試環境上執行，且不崩潰。
- 至少一個 VirusTotal 平臺內之防毒軟體判定該上傳執行檔案樣本為惡意。
- 處理程序監視器所記錄的行為是否出現任務列表所描述的惡意行為。

實驗-實驗方法

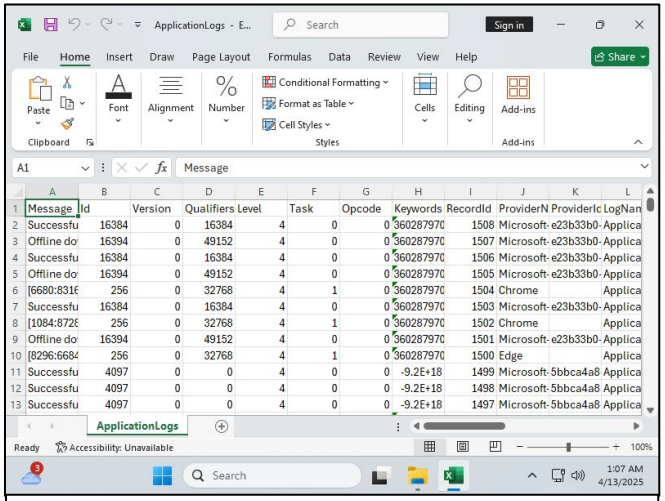

於虛擬環境中實際執行
EXE
Malware.exe
Victim Machine
LOG
System Logs
VirusTotal
Analysis Reports
上傳至VirusTotal

實驗-可執行檔實際執行結果

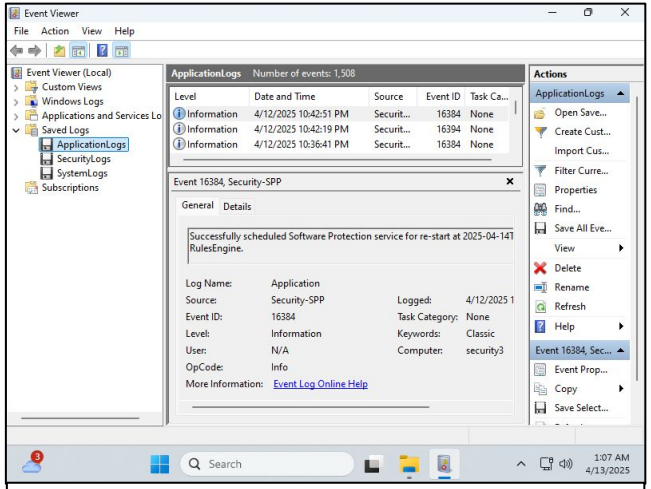




執行後所生產的檔案

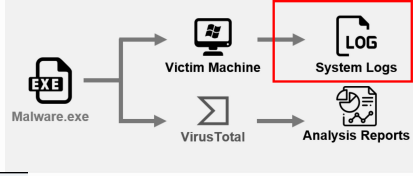


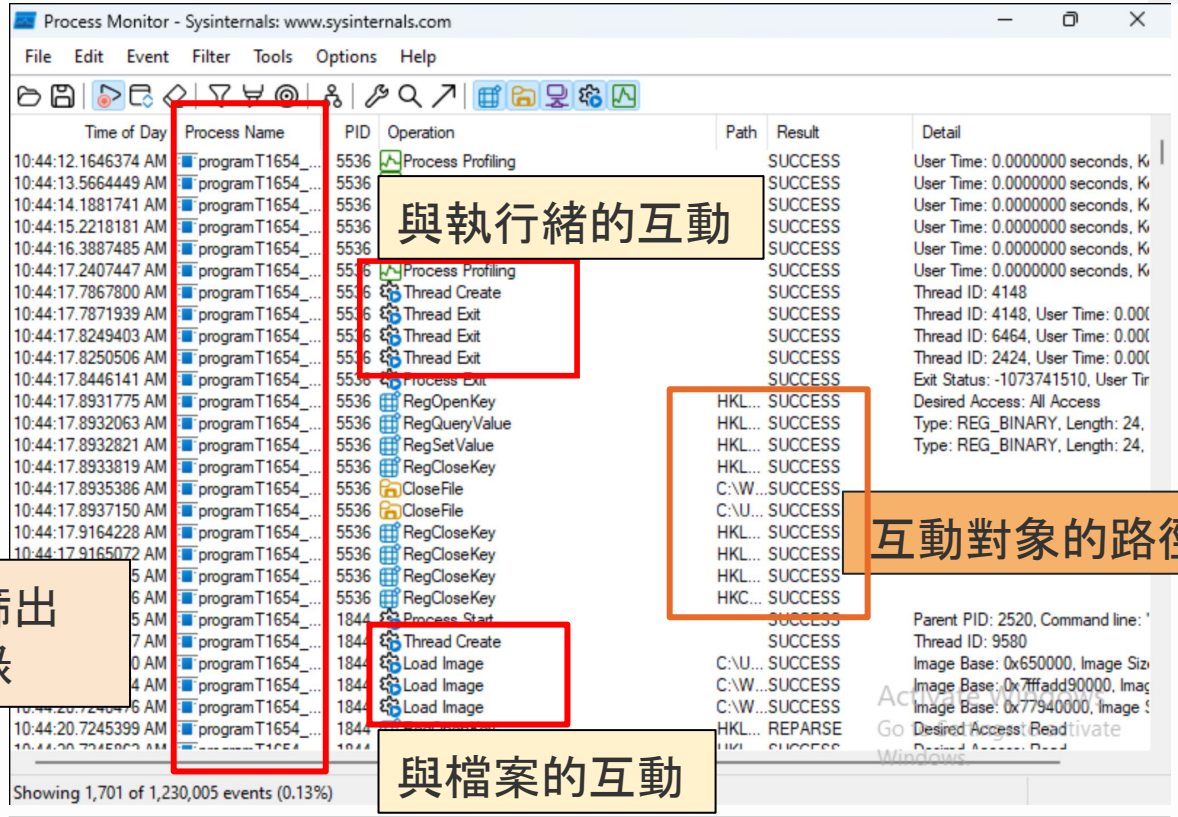
檔案內容-**ApplicationLogs (.CSV 檔)**



檔案內容-**Application (Windows Events 記錄檔)**

實驗-系統程序監視器

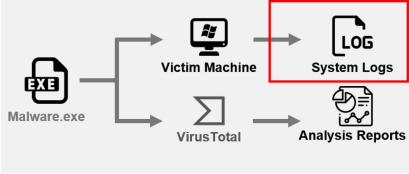


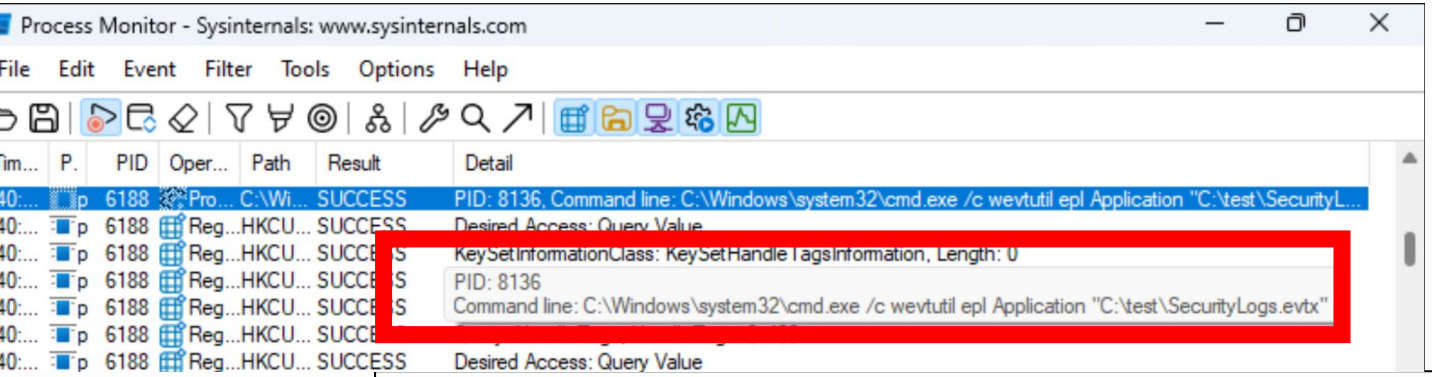


使用過濾功能篩出
特定程式的紀錄

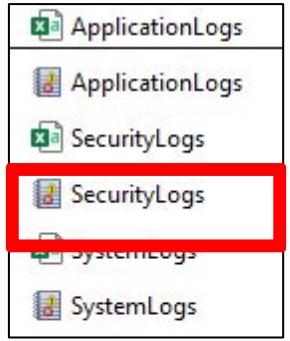
使用處理程序監視器錄製可執行檔與環境互動

實驗-系統程序監視器

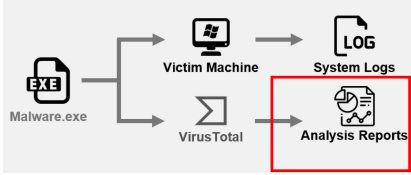




處理程序監視器成功在錄製過程中抓到關於擷取並將檔案儲存在另一位置的行為



實驗-使用Virustotal 平臺檢驗生成結果



被 5 個防毒軟體標記為惡意

不同防毒軟體所分類的結果

- Win/malicious_confidence_60% (W)
- Suspicious.Win32.Save.a
- BScope.TrojanRansom.Crypren
- Malicious (score:100)
- Malicious

實驗-目前驗證過的攻擊手法

ID	Name	Tactic	可執行	防毒軟體	Threat Category	Family Labels
T1005	Data from Local System	Collection	V	7/72	Trojan	
T1010	Application Window Discovery	Discovery	V	7/72	Trojan	
T1059	Command and Scripting Interpreter	Execution	V	26/72	Trojan	reverseturtle
T1083	File and Directory Discovery	Discovery	V	11/72	Trojan	
T1113	Screen Capture	Collection	V	9/72	Trojan	
T1652	Device Driver Discovery	Discovery	V	4/71	Trojan/Malicious	
T1654	Log Enumeration	Discovery	V	5/71	Trojan/Malicious	
T1007	System Service Discovery	Discovery	V	15/72	Trojan	
T1120	Peripheral Device Discovery	Discovery	V	8/72	Trojan	
T1124	System Time Discovery	Discovery	V	5/72	Malicious	
T1136	Create Account	Persistence	V	8/72	Trojan	
T1562	Impair Defenses	Defense Evasion	V	27/72	Trojan	heur3

攻擊手法名稱

被分類為惡意程式的防毒軟體數量

結論

本研究限制：

- 目前僅針對 C++ 語言與 Windows 11 平臺進行實驗
- 未涵蓋其他程式語言或作業系統
- 未涵蓋結合外部攻擊工具的惡意程式行為模擬

未來可延伸的方向包括：

- 導入系統監視器與事件檢視器等多源監控資料以強化行為驗證
- 結合公開威脅情報報告生成真實攻擊樣本
- 整合外部駭客工具與自動化混淆技術

Q&A