



# 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

# Outline

- 1**      介紹
- 2**      方法
- 3**      實驗
- 4**      Q&A

# 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

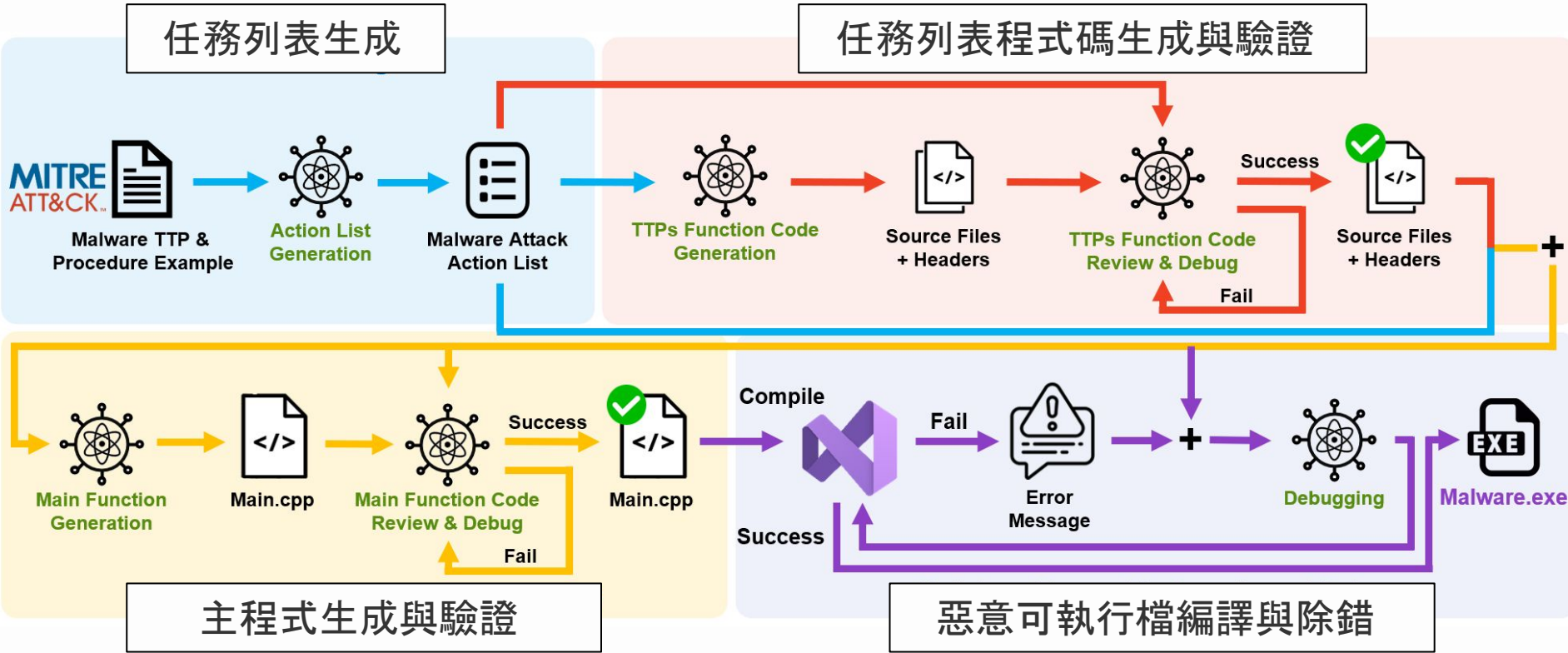
MITRE ATT&CK 框架是美國非營利組織 MITRE所創建的公開攻擊手法知識庫，分類攻擊者的攻擊行為，進行定義、分類和說明，其中包含戰術 (Tactics)、攻擊手法 (Techniques) 及實行技術的程序 (Procedures)。許多資安研究者與從業人員採用該框架，作為描述 攻擊活動的生命週期。

## MITRE ATT&CK 網站上把攻擊手法歸類彙整成 14個階段

Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Active Scanning (3) Gather Victim Host Information (4) Gather Victim Identity Information (3) Gather Victim Network Information (6) Gather Victim Org Information (4) Phishing for Information (4)	Acquire Access Acquire Infrastructure (8) Compromise Accounts (3) Compromise Infrastructure (8) Develop Capabilities (4) Establish Accounts (3)	Content Injection Drive-by Compromise Exploit Public-Facing Application External Remote Services Hardware Additions Phishing (4) Replication Through Removable Media	Cloud Administration Command Command and Scripting Interpreter (12) Container Administration Command Deploy Container ESXi Administration Command Exploitation for Client Execution	Account Manipulation (7) BITS Jobs Boot or Logon Autostart Execution (14) Boot or Logon Initialization Scripts (5) Cloud Application Integration	Abuse Elevation Control Mechanism (6) Access Token Manipulation (5) Account Manipulation (7) Boot or Logon Autostart Execution (14) Boot or Logon Initialization	Abuse Elevation Control Mechanism (6) Access Token Manipulation (5) BITS Jobs Build Image on Host Debugger Evasion Deobfuscate/Decode Files or Information Deploy Container	Adversary-in-the-Middle (4) Brute Force (4) Credentials from Password Stores (6) Exploitation for Credential Access Forced Authentication Forge Web Credentials (2)	Account Discovery (4) Application Window Discovery Browser Information Discovery Cloud Infrastructure Discovery Cloud Service Dashboard Cloud Service Discovery Cloud Storage Object	Exploitation of Remote Services Internal Spearphishing Lateral Tool Transfer Remote Service Session Hijacking (2) Remote Services (8) Replication Through Removable Media	Adversary-in-the-Middle (4) Archive Collected Data (3) Audio Capture Automated Collection Browser Session Hijacking Clipboard Data Data from Cloud	Application Layer Protocol (5) Communication Through Removable Media Content Injection Data Encoding (2) Data Obfuscation (3) Dynamic Resolution (3)	Automated Exfiltration (1) Data Transfer Size Limits Exfiltration Over Alternative Protocol (3) Exfiltration Over C2 Channel Exfiltration Over Other Network Medium (1)	Account Access Removal Data Destruction (1) Data Encrypted for Impact Data Manipulation (3) Defacement (2) Disk Wipe (2) Email Bombing

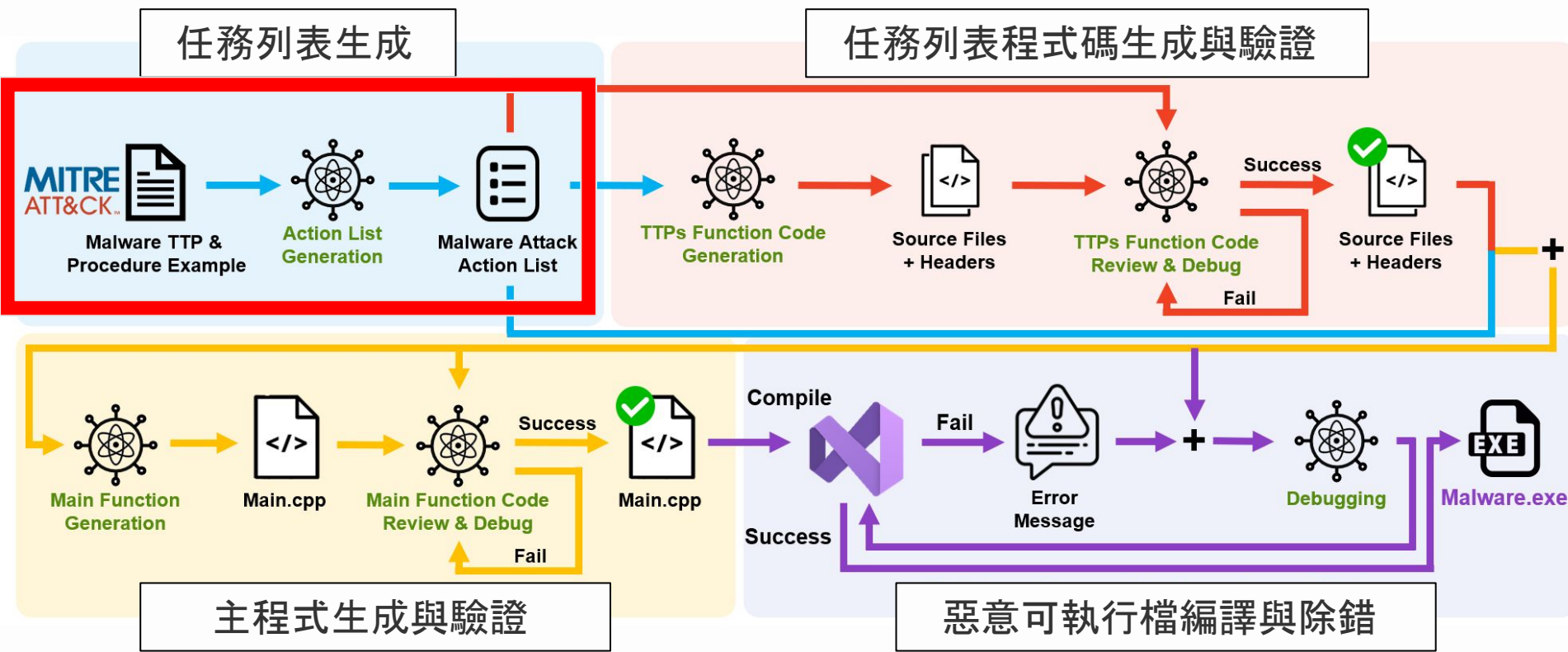
為了產生有效的模擬惡意威脅，本研究提出 Mal-GPT：  
Mal-GPT基於MITRE ATT&CK框架所定義的攻擊手法描述範例作為輸入，透過提示工程，使得大型語言模型撰寫程式，以產生可執行特定攻擊手法的惡意程式。

# 方法-架構圖

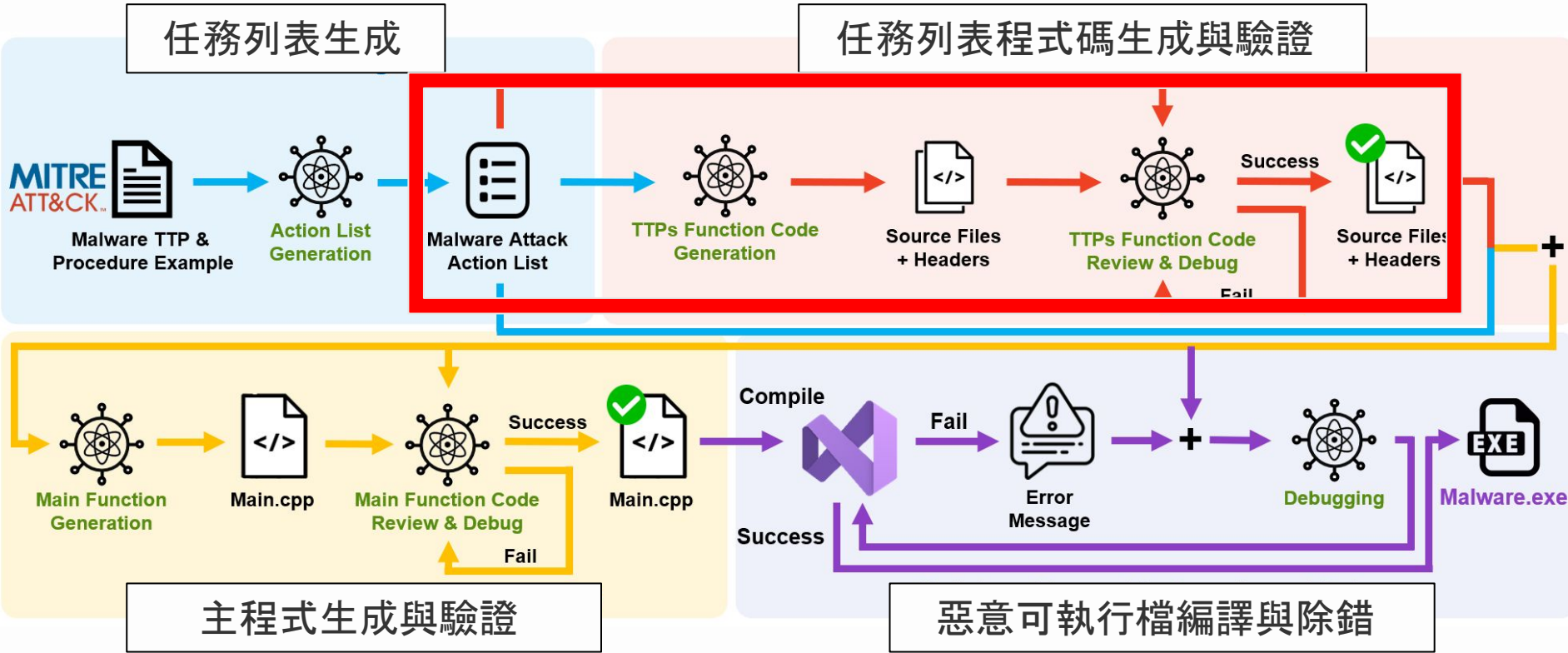


LLM:GPT-4o-mini

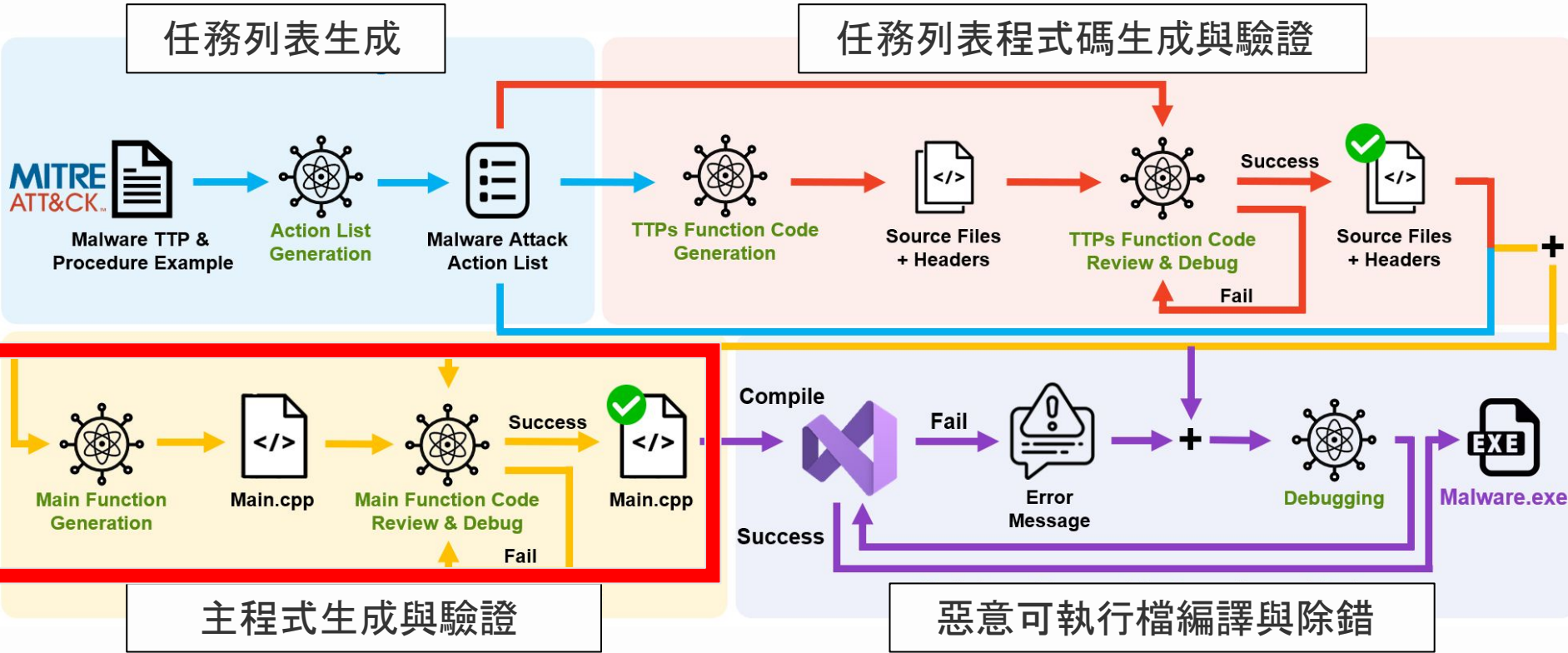
# 方法-架構圖



# 方法-架構圖

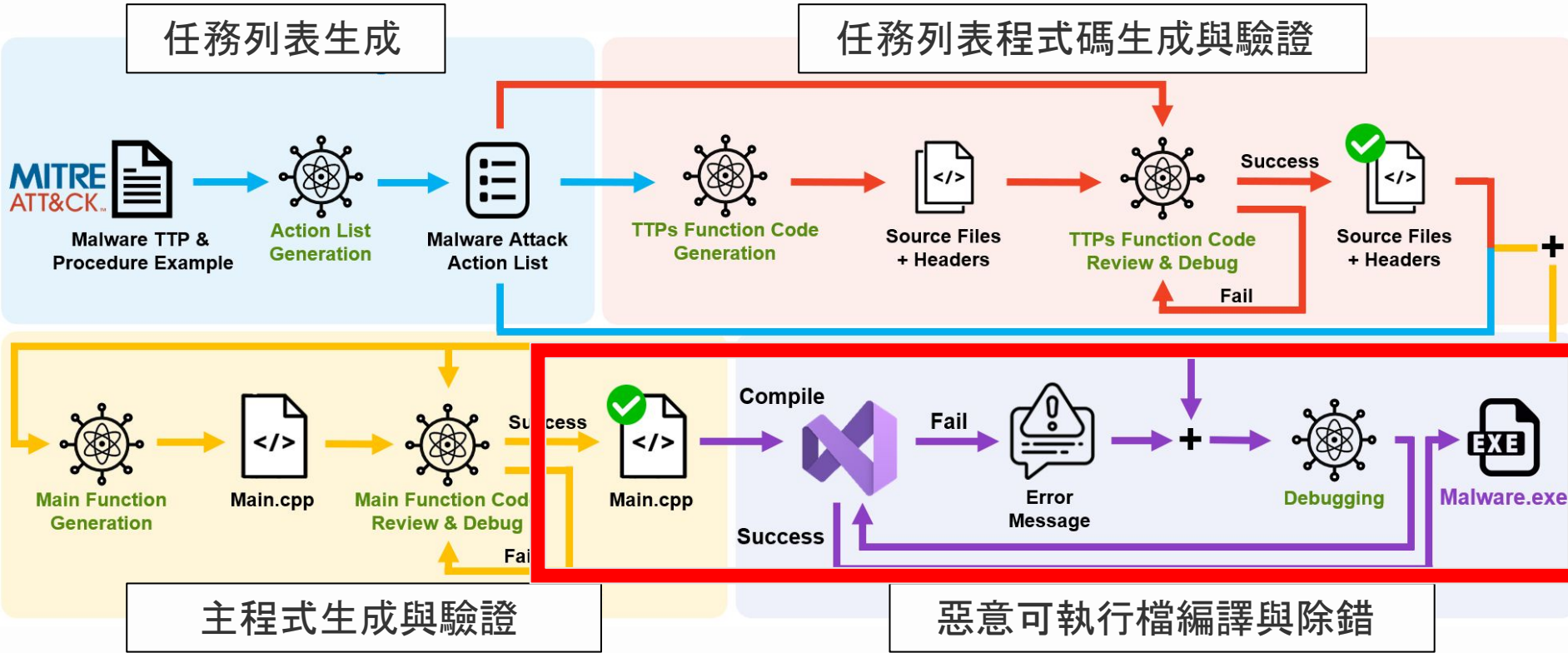


# 方法-架構圖





# 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

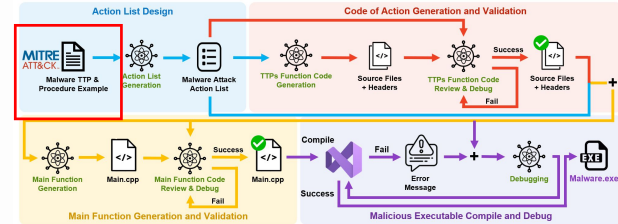
### Log Enumeration

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.<sup>[1][2]</sup> In cloud environments, adversaries may leverage utilities such as the Azure VM Agent's `CollectGuestLogs.exe` to collect security logs from cloud hosted infrastructure.<sup>[3]</sup>

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 in real time to track incident response procedures. This may allow them to adjust their techniques in order to maintain persistence or evade defenses.<sup>[4]</sup>



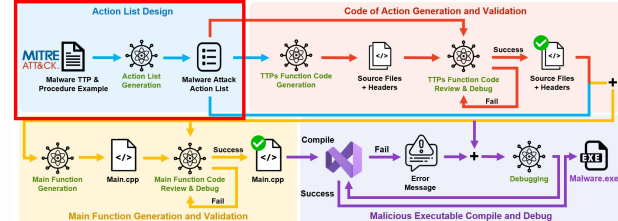
## 純文字敘述

### Log Enumeration

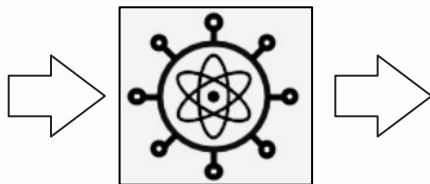
Adversaries may enumerate system and service logs to find useful data. These logs may highlight various types

...

# 方法-任務列表生成



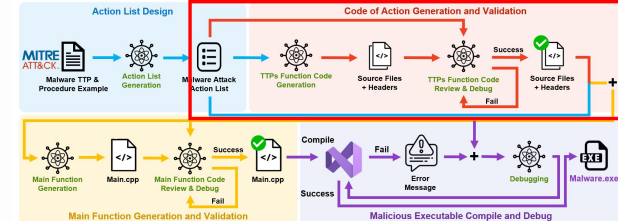
**<task>** Read the given text and write a plan based on the given example and the description... **</task>**  
**<instruction>** You are an experienced C++ programmer. **</instruction>**  
**<input>** Log Enumeration  
Adversaries may enumerate system and service logs to find useful data. These log...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



- "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\" -NoTypeInfoInformation; 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\" -NoTypeInfoInformation; This command will export the system event logs to a CSV file."...

攻擊手法-紀錄列舉  
(Log Enumeration)的敘述作為輸入

# 方法-任務列表程式碼生成與驗證

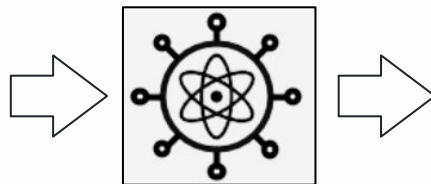


**<task>** Your current task is to implement a header file and a C++ source file for the given action. **</task>**

**<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 "  
'--NoTypeInfoInformation.... **</input>**



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

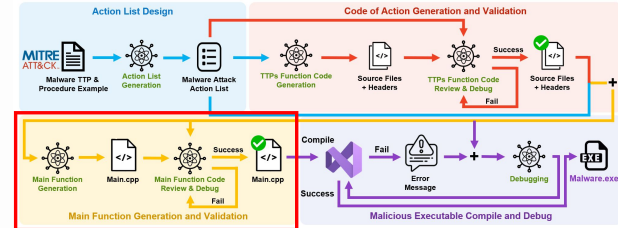
- Header:

```
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'
  -NoTypeInfoInformation'";
7 bool exportSecurityLogs() {
8   int result = system(command.c_str());
9   return (result == 0);}
```

# 方法-主程式生成與驗證

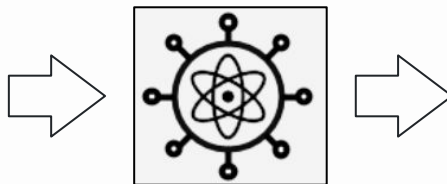


**<task>**Please write a main.cpp program with only main function to rearrange the order and combine all C++ source files according to the procedure list.**</task>**

**<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

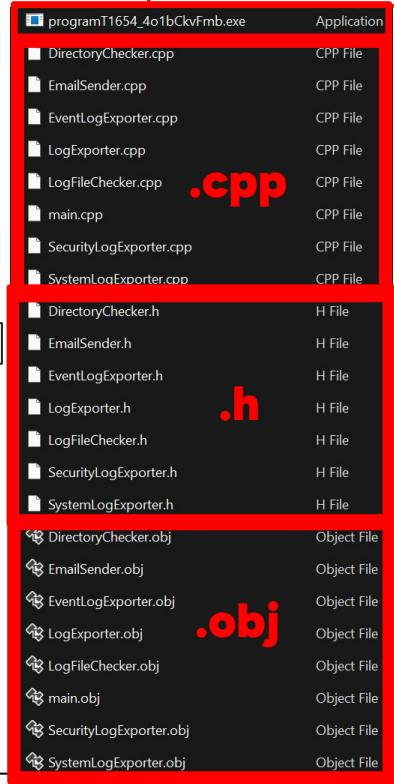
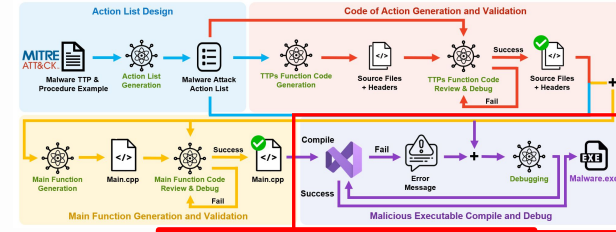
```
& 'C:\Program Files\Microsoft Visual  
Studio\2022\Community\Common7\Tools\La  
unch-VsDevShell.ps1'  
cd  
'D:\Automatic-Malware-Generation-Using-LL  
Ms\response\T1654_4o1bCkvFmb\code'  
cl /std:c++17 /EHsc DirectoryChecker.cpp  
EmailSender.cpp EventLogExporter.cpp  
LogExporter.cpp LogFileChecker.cpp  
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)

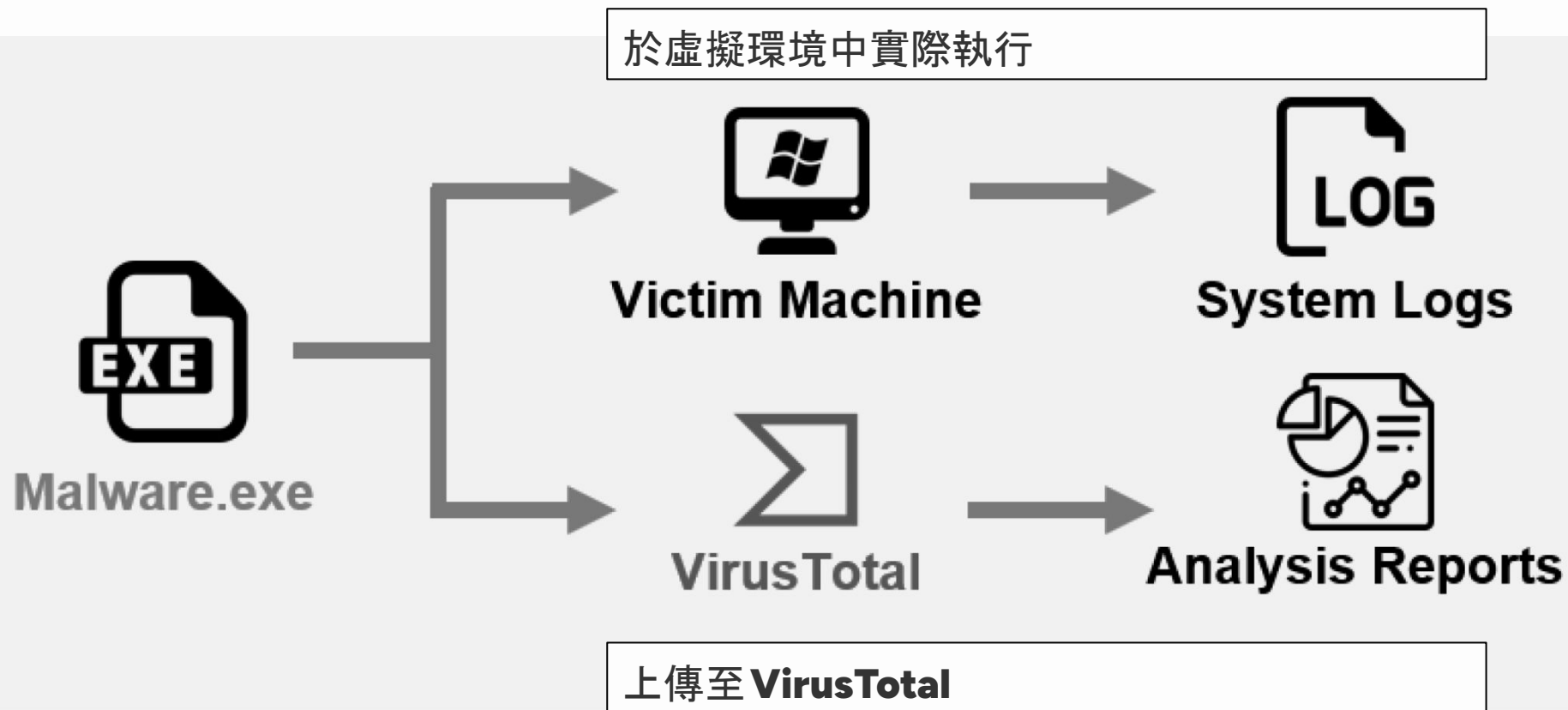


生成成功後會出現的所有檔<sup>14</sup>

# 實驗

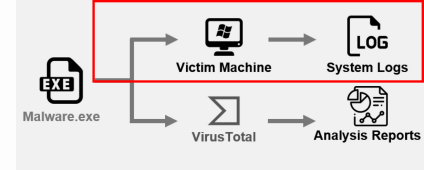
- RQ1 可執行與否：Mal-GPT 生成的惡意程式是否被正常執行？
  - RQ2 是否具有被判定為惡意程式的特徵：線上防毒軟體是否將 Mal-GPT 所生成的程式判定為有害？
  - RQ3 實際功能性：Mal-GPT 所生成的惡意程式是否能成功執行並完成預期攻擊行為？
- 
- 目標程式能夠在測試環境上執行，且不崩潰。
  - 至少一個 VirusTotal 平臺內之防毒軟體判定該上傳執行檔案樣本為惡意。
  - 處理程序監視器所記錄的行為是否出現任務列表所描述的惡意行為。

# 實驗-實驗方法





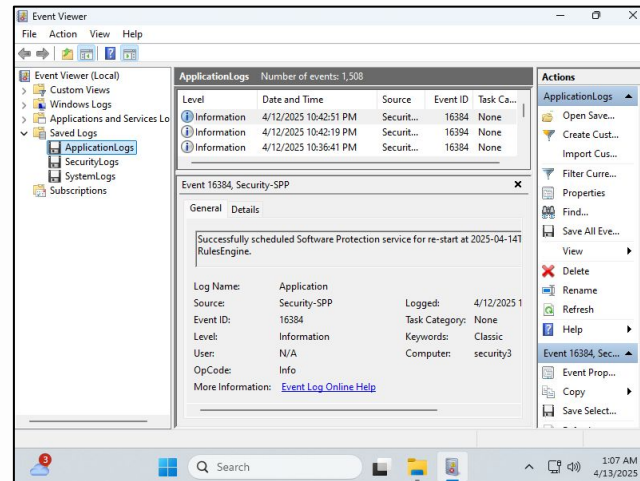
# 實驗-可執行檔實際執行結果



執行後所生產的檔案

Message	Id	Version	Qualifiers	Level	Task	Opcode	Keywords	RecordId	ProviderN	ProviderId	LogName
Successful	16384	0	16384	4	0	0	360287970	1508	Microsoft	e23b33b0	Applica
Offline do	16394	0	49152	4	0	0	360287970	1507	Microsoft	e23b33b0	Applica
Successful	16384	0	16384	4	0	0	360287970	1506	Microsoft	e23b33b0	Applica
Offline do	16394	0	49152	4	0	0	360287970	1505	Microsoft	e23b33b0	Applica
[6680:831f	256	0	32768	4	1	0	360287970	1504	Chrome		Applica
Successful	16384	0	16384	4	0	0	360287970	1503	Microsoft	e23b33b0	Applica
[1084:872f	256	0	32768	4	1	0	360287970	1502	Chrome		Applica
Offline do	16394	0	49152	4	0	0	360287970	1501	Microsoft	e23b33b0	Applica
[8296:6684	256	0	32768	4	1	0	360287970	1500	Edge		Applica
Successful	4097	0	0	4	0	0	-9.2E+18	1499	Microsoft	5bbca4a8	Applica
Successful	4097	0	0	4	0	0	-9.2E+18	1498	Microsoft	5bbca4a8	Applica
Successful	4097	0	0	4	0	0	-9.2E+18	1497	Microsoft	5bbca4a8	Applica

檔案內容-**ApplicationLogs**  
(.CSV 檔)



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

```

graph LR
    Malware[Malware.exe] --> Victim[Victim Machine]
    Malware --> VirusTotal[VirusTotal]
    Victim --> Logs[LOG System Logs]
    VirusTotal --> Reports[Analysis Reports]
    style Logs stroke:#f00,stroke-width:2px
  
```

## 使用過濾功能篩出特定程式的紀錄

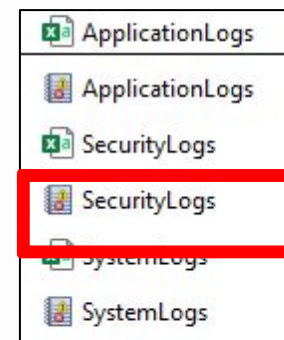
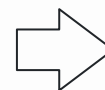
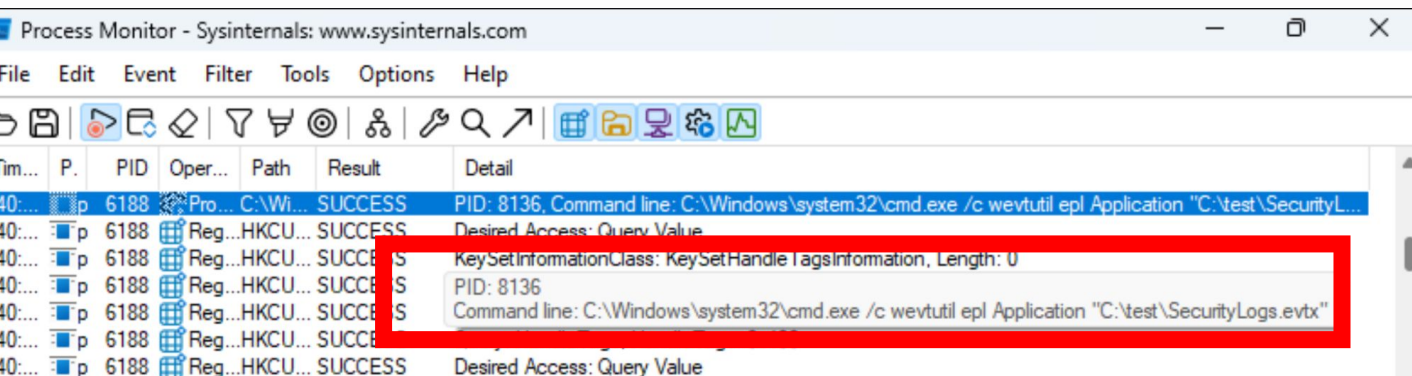
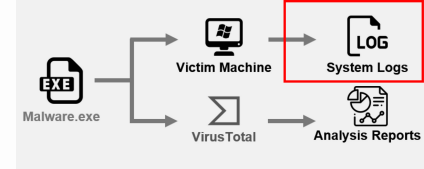
```
HKL... SUCCESS
HKL... SUCCESS
HKL... SUCCESS
HKL... SUCCESS
C:\W...SUCCESS
C:\U...SUCCESS
HKL... SUCCESS
HKL... SUCCESS
HKL... SUCCESS
HKC... SUCCESS
```

## 互動對象的路徑

## 與檔案的互動

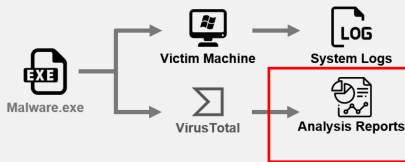
## 使用處理程序監視器錄製可執行檔與環境互動

# 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

5/72 security vendors flagged this file as malicious

ca3b72fed6962c5bbdbbdd049f6a6600546013c4c6bb5cf9b23644a21f1932ae

Size: 255.00 KB | Last Analysis Date: 29 minutes ago

programT1654\_4o1bCkvFmb.exe

peexe detect-debug-environment long-sleeps

Community Score: 5 / 72

Reanalyze Sim

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY

不同防毒軟體所分類的結果

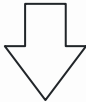
Security vendors' analysis			Do you
CrowdStrike Falcon	Win/malicious_confidence_60% (W)	Cynet	Malicious (score: 100)
Sangfor Engine Zero	Suspicious.Win32.Save.a		
VBA32	BScope.TrojanRansom.Crypren		

- 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	reverseturtle
T1010	Application Window Discovery	Discovery	V	7/72	Trojan	
T1059	Command and Scripting Interpreter	Execution	V	26/72	Trojan	
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	heur3
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	

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