AIS3

教育部先進資通安全實務人才培育計畫

113年度新形態資安實務暑期課程

主題：AI 資安情報員 – 但好像少了些什麼

組別：情資運用及防禦 B1
姓名：陸紹祺、莊舒涵、張簡雲翔、蕭愷宸

目錄

研究動機

01

利用AI技術統整最新的時事

02

惡意檔案查詢功能與攻擊技術

03

結合最新資訊提供相關的建議、防禦方式

相關實作 - HackBot

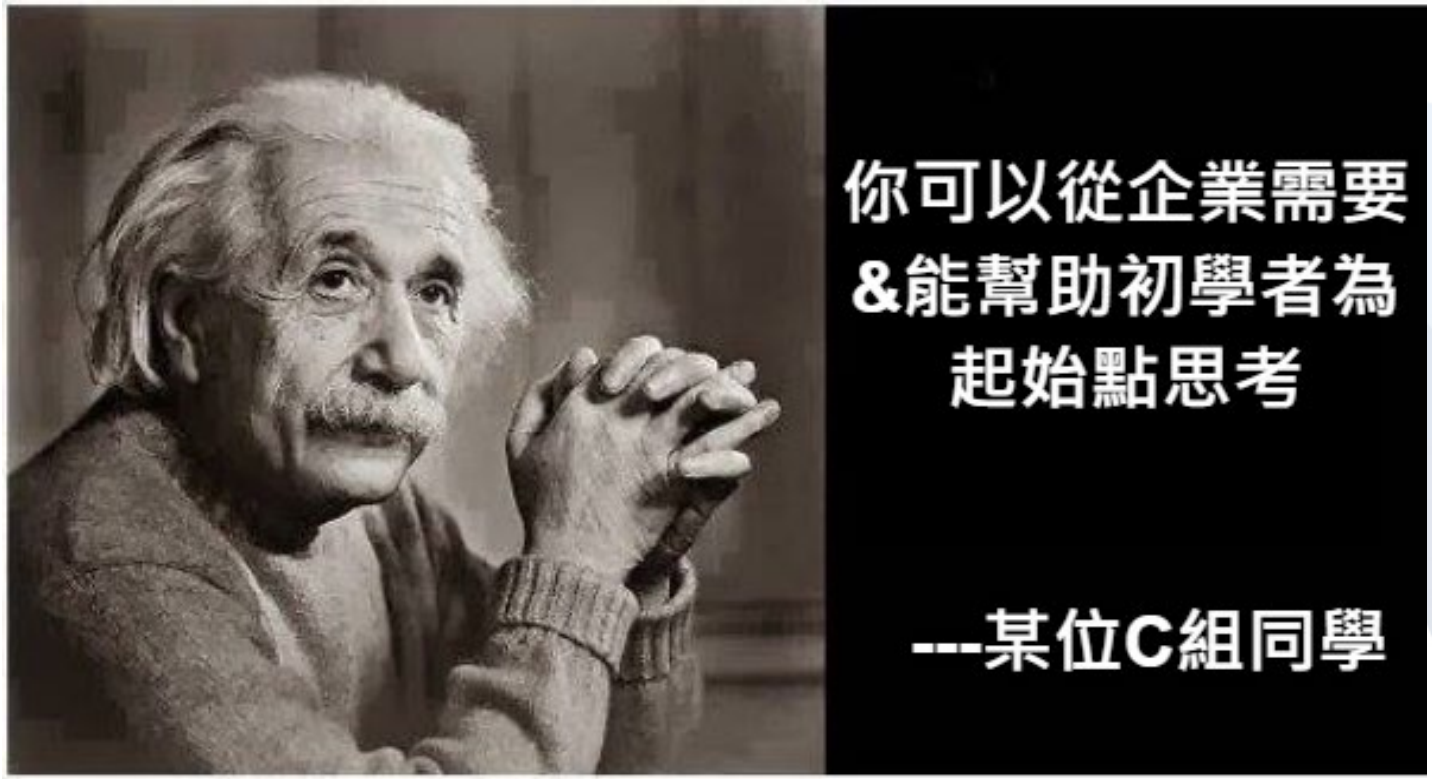
- https://github.com/morpheuslord/HackBot
- 使用之 LLM 為 Llama2
- 資料集來自 CVE 資料庫

前情提要

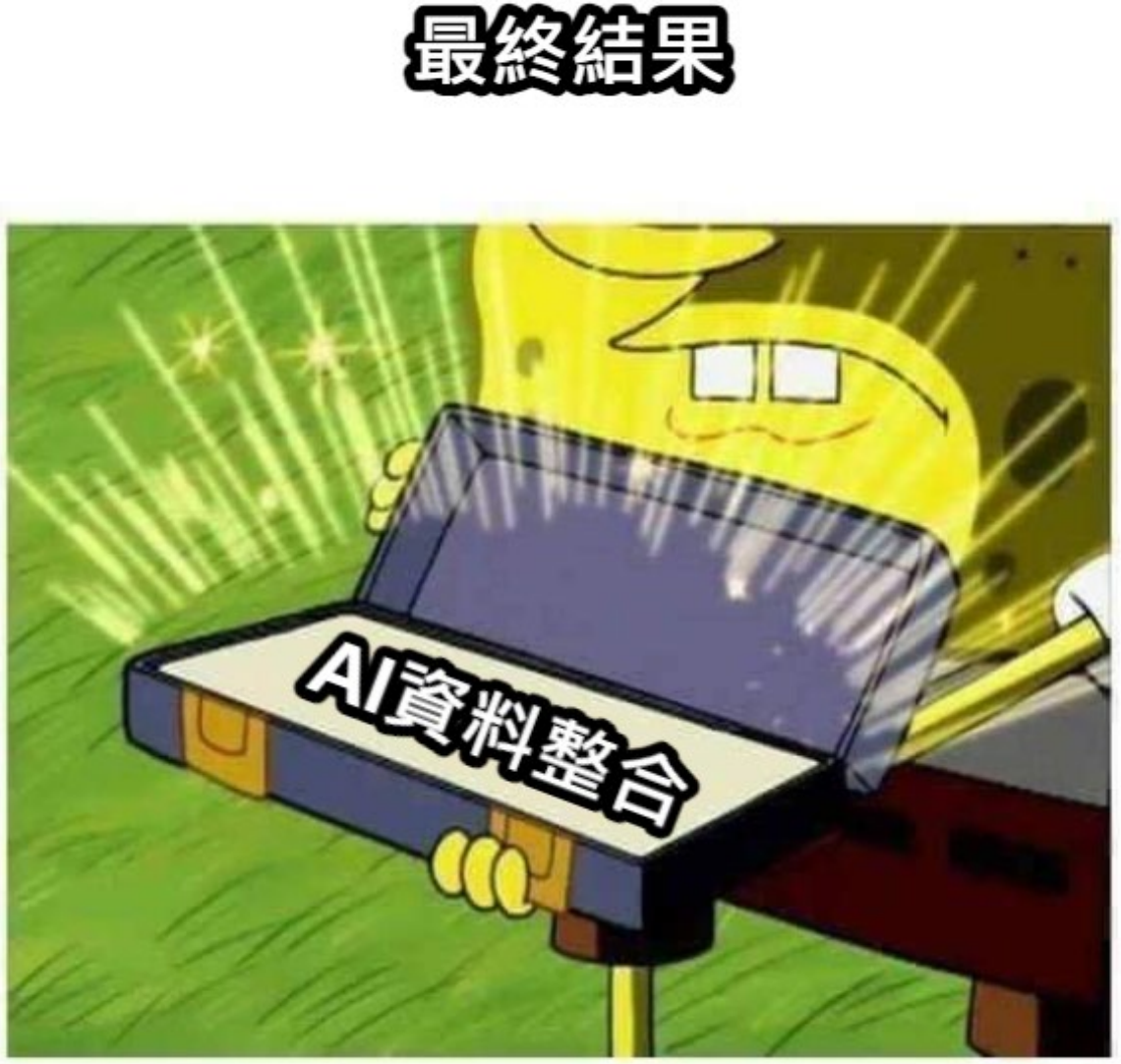
前情提要

- 在專題決定的時候...

一籌莫展時...

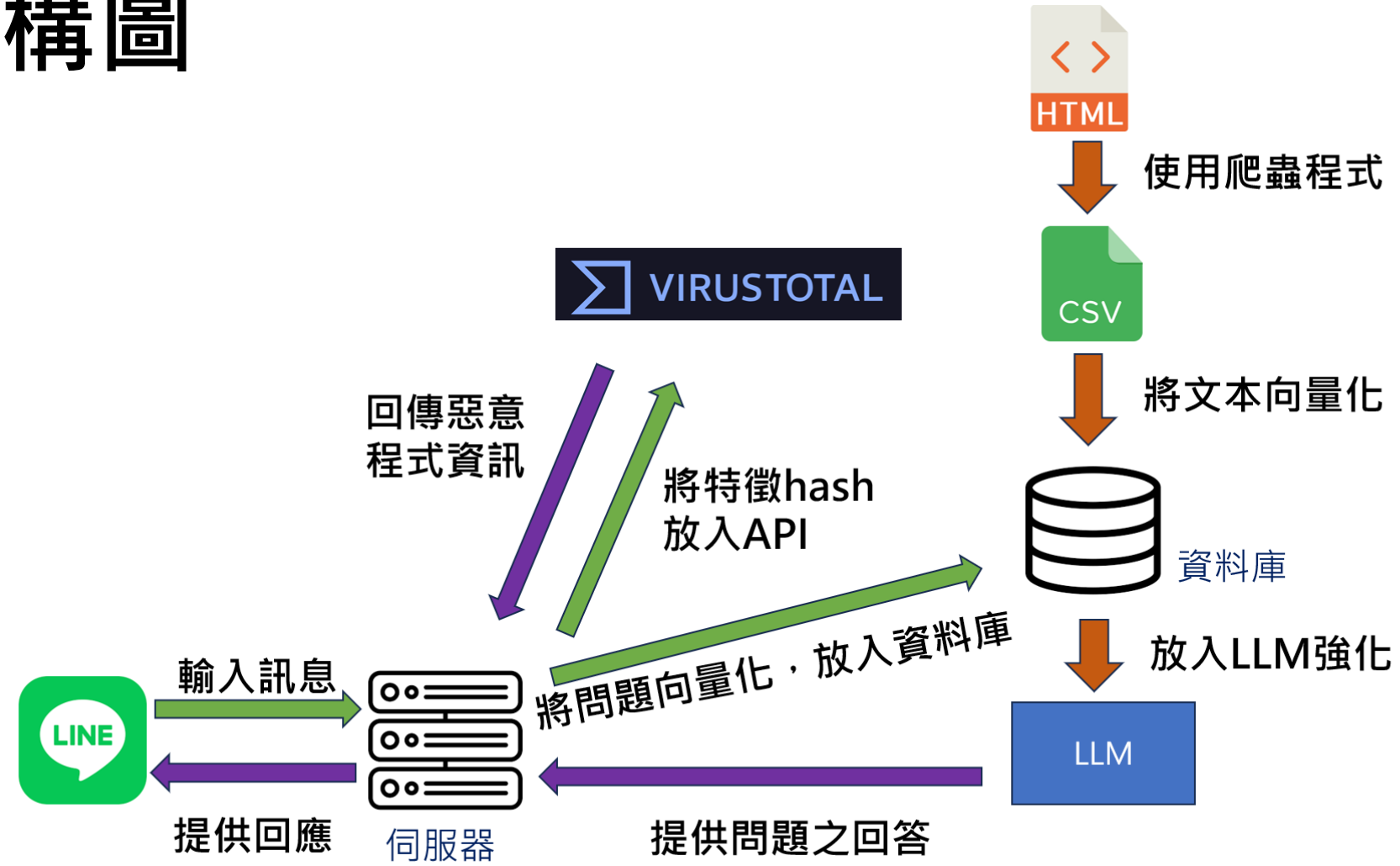


7/31的晚上...



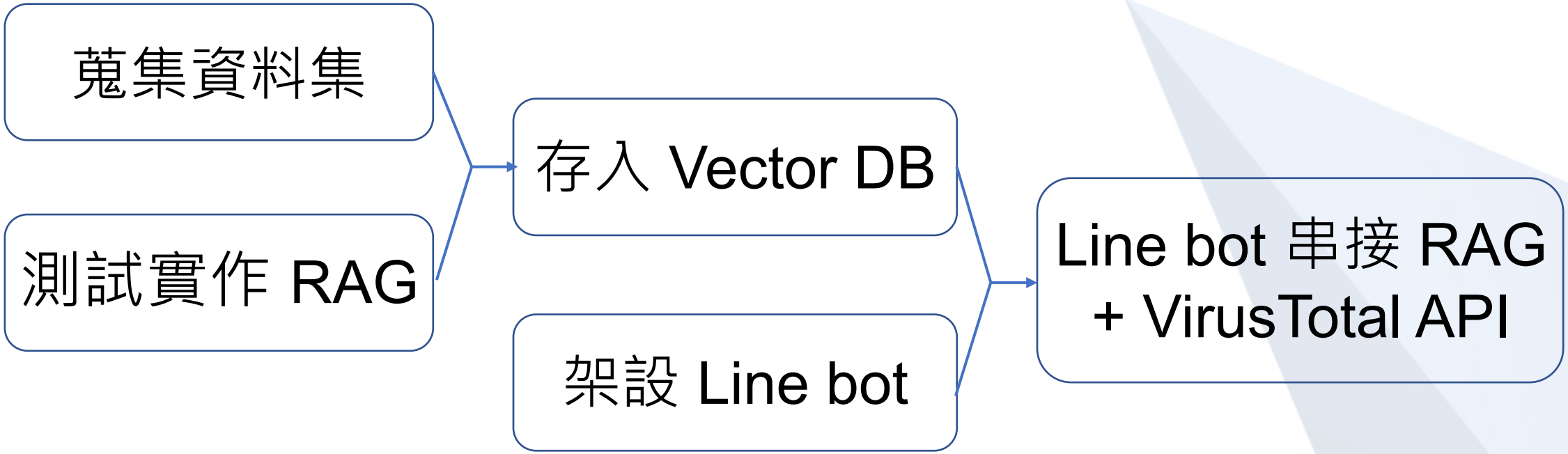
系統架構圖

系統架構圖


HTML
使用爬蟲程式
CSV
將文本向量化
VIRUSTOTAL
回傳惡意
程式資訊
將特徵hash
放入API
資料庫
放入LLM強化
LINE
輸入訊息
將問題向量化，放入資料庫
LLM
提供回應
伺服器
提供問題之回答

實作方法&遇到困難

實驗架構圖


蒐集資料集
測試實作 RAG
存入 Vector DB
架設 Line bot
Line bot 串接 RAG
+ VirusTotal API

1. 蒐集資料集

- 由資安新聞網站爬下新聞標題 + 內文
  - Broadcom.com
  - darkreading.com
  - cyware.com
- 使用自動爬蟲軟體 EasyScraper 將資料擷取下來
- 自動轉檔為 .csv 格式

1. 蒐集資料集

資料範例

SARA Android Ransomware Targets Vietnamese Mobile Users in Fake App Scheme , Android lockers and ransomware were prevalent a couple of years ago, especially during the RansomLock craze. Today, while they remain in the mobile threat landscape, their prevalence has dwindled. These threats typically lock users out of their devices and display a ransom message, demanding payment to regain access with an unlock code. Many variants exist, but one particular variant with screen locking and file encryption capabilities, SARA, appears to have gained traction.

2. 使用少量資料測試實作 RAG

- 主要使用 Langchain 套件實作
- 向量資料庫：Chroma
- LLM：Llama3

2. 使用少量資料測試實作 RAG

```
docs = [(資料 list 略)]

text_splitter = CharacterTextSplitter(chunk_size=20, chunk_overlap=5)
documents = text_splitter.split_documents(docs)

embeddings = OllamaEmbeddings()
```

設定文本分割器將資料分割，chunk_size 是分割的大小，

chunk_overlap 是重疊的部分

2. 使用少量資料測試實作 RAG

```
persist_directory = 'db'
vectordb = Chroma.from_documents(documents=documents,
embedding=embeddings, persist_directory=persist_directory)
```

使用 Chroma 建立向量資料庫

2. 使用少量資料測試實作 RAG

```
retriever = vectordb.as_retriever()
```

Vector Database 設為 retriever

```
document_chain = create_stuff_documents_chain(llm, prompt)

retrieval_chain = create_retrieval_chain(retriever, document_chain)
```

建立 retrieval_chain，將 vectordb 搜索後的資訊與 LLM 相連

2. 使用少量資料測試實作 RAG

```
context = []
input_text = input('>>> ')
while input_text.lower() != 'bye':
    response = retrieval_chain.invoke({
        'input': input_text,
        'context': context
    })
    print(response['answer'])
    context = response['context']
    input_text = input('>>> ')
```

讓使用者輸入問題並取得回應

2. 使用少量資料測試實作 RAG

- 使用 3 筆資料測試
- 能成功依照資料回應，但前處理及回應時間極長

```
>>> SMS Stealer
According to my previous output, SMS Stealer is a type of Android malware that has been distributed through an extensive campaign tar
geting victims in 113 countries since at least 2022. It primarily spreads via malvertising and Telegram bots, leading users to malici
ous webpages mimicking the Google Play store. The malware has data theft capabilities focused on stealing SMS messages and 2FA One-ti
me passwords (OTPs), which could aid attackers in further fraudulent activities or phishing campaigns.
```

2. 使用少量資料測試實作 RAG

Problem：

每次會重新將所有資料做 embedding 存入資料庫一次，耗時長

⇒先將所有資料存入資料庫，欲使用時直接載入

3. 資料存入 Vector Database

- 將所有資料 embed 之後放入 vector database
- ~~因為有 batch size 限制，每次約放入 100 筆資料~~
- ~~1000+ 筆資料~~

3. 資料存入 Vector Database

```
docs = []
loader = CSVLoader('.\data\\thehackernews-2024-07-31.csv')
docs = loader.load_and_split()

documents = text_splitter.split_documents(docs)
vectordb = Chroma.from_documents(documents=documents,
embedding=embeddings, persist_directory=persist_directory)

vectordb.persist()
```

首次存入使用 from_documents 建立 db

儲存資料庫

3. 資料存入 Vector Database

```
vectordb = Chroma.add_documents(documents=documents,
embedding=embeddings,
persist_directory=persist_directory)

vectordb.persist()
```

第二次後存入以 add_documents 新增資料

儲存資料庫

4. 使用 Django 及 Ngrok 架設 Line bot

- 以 Django 架構建置 Line bot 專案
- Ngrok 讓本機運行的服務能從公網存取

5. Line bot 回應串接 RAG

- Line bot server 跑起來時也讓 LLM 跑起來
- 在 Line bot 的回應部分 code 加入呼叫 LLM 的部分

```
context = []
response = retrieval_chain.invoke({
    'input': input_text,
    'context': context
})
line_bot_api.reply_message(
    event.reply_token,
    TextSendMessage(text=response['answer'])
)
```

6. 加入 VirusTotal API

- 讓使用者可以利用「VirusTotal <hash_value>」的格式直接查詢檔案在 VirusTotal 上的資訊
- 此處因為時間不足只有做到傳送檔案 hash 查詢

6. 加入 VirusTotal API

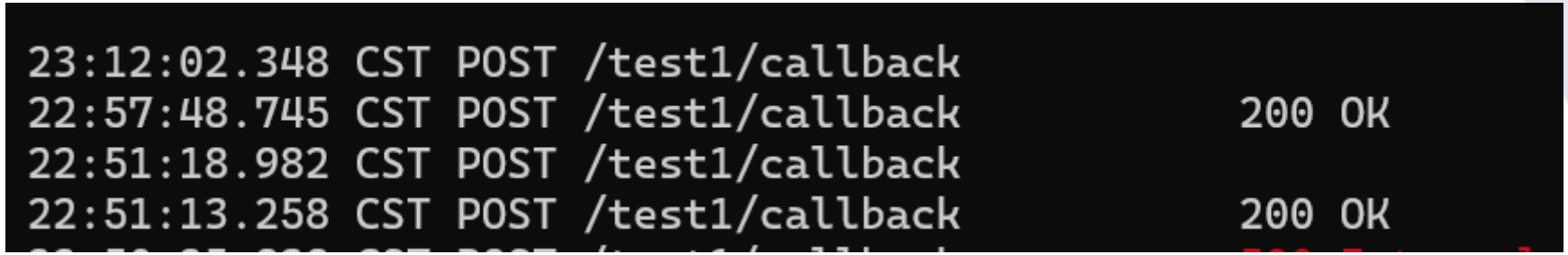
```
if input_text[:10] == "VirusTotal":
    file = client.get_object("/files/"+input_text[11:])
    line_bot_api.reply_message(
        event.reply_token,
        TextSendMessage(text=file.last_analysis_stats)
    )
```

呼叫 VirusTotal api 查詢此 hash 的相關資訊
last_analysis_stats 為近期掃描所得資料

範例格式：{malicious': 68, 'suspicious': 0, 'undetected': 4, 'harmless': 0, 'timeout': 0, 'confirmed-timeout': 0, 'failure': 0, 'type-unsupported': 7}

遇到的困難

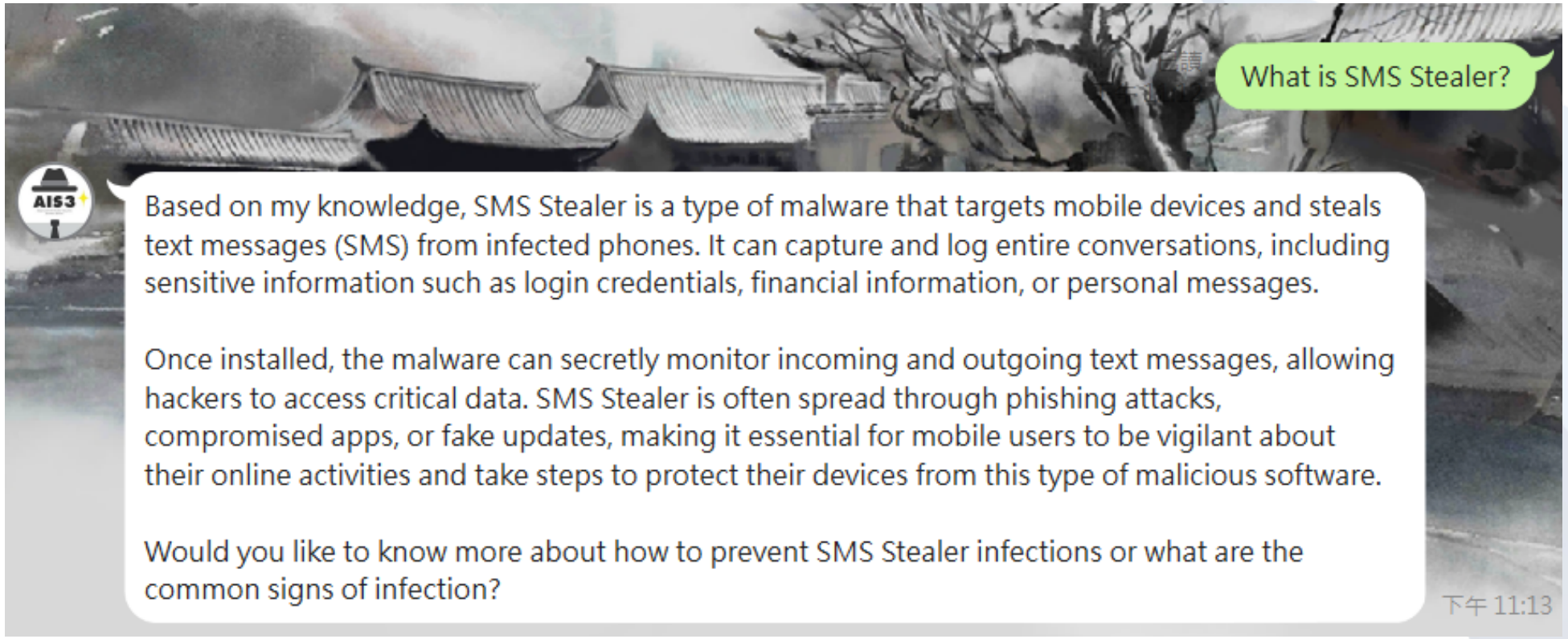
- **算力不足：我們的運算資源是一張 MX450（2GB VRAM）**
- 文件上寫 from_documents 會把以前的資料覆蓋，但實作發現如果沒有刪除的話會影響結果使搜尋時完全搜尋不到剛放進去的資料
- 回應久到 ngrok 沒抓到 HTTP status code



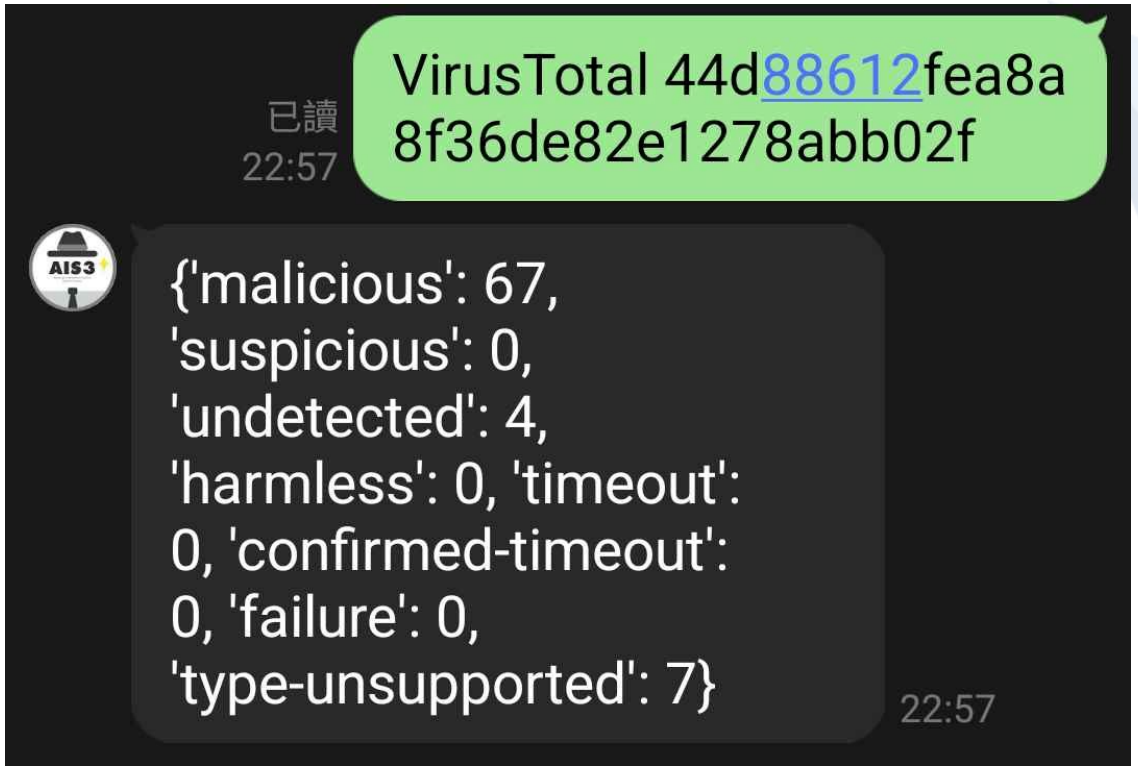
DEMO

問與資料集相關的資訊

※ 這是 3 筆資料的成果



VirusTotal API 查詢



結論

References

- https://medium.com/@cch.chichieh/rag%E5%AF%A6%E4%BD%9C%E6%95%99%E5%AD%B8-langchain-llama2-%E5%89%B5%E9%80%A0%E4%BD%A0%E7%9A%84%E5%80%8B%E4%BA%BAllm-d6838febf8c4
- https://medium.com/@weiberson/langchain-rag%E5%AF%A6%E4%BD%9C%E6%95%99%E5%AD%B8-%E7%B5%90%E5%90%88llama3%E8%AE%93llm%E5%8F%AF%E4%BB%A5%E8%AE%80%E5%8F%96pdf%E5%92%8Cdoc%E6%96%87%E4%BB%B6-%E4%B8%A6%E7%94%A2%E7%94%9F%E5%9B%9E%E6%87%89-2e7a0b2aacc1