Communications of the CCISA
Vol. 31 No. 2 May 2025

基於模型上下文協定與大型語言模型之自動化資安檢測框架

陳緯杰[1]、廖翊廷[2]、黃賜欽[3]、王銘宏[4*]
[1234] 國立中正大學資訊工程學系
[1] j38583172@gmail.com、[2] yitingliao2003@gmail.com、[3] mr.47discord@gmail.com、
[4] tonymhwang@gmail.com

摘要

本研究提出一個整合大型語言模型 (LLM) 與模型上下文協定 (MCP) 的智慧型資安檢測架構，其目標為降低傳統資安工具的操作複雜性並提升資安任務的自動化與報告的可讀性。結合 MCP 架構，Agent 可主動調用外部資安工具，執行威脅偵測，並自動生成結構化、易於理解的資安分析報告。在框架中設計中包含兩階段安全性評估機制，利用外部大型語言模型進行多面向的風險評估，有效識別並阻擋高風險或惡意的任務指令，保證系統安全性。

本架構具有以下特點：(1) 自然語言操作介面，降低使用者的技術門檻；(2) 強化自動化能力，能迅速執行即時偵測任務；(3) 提供結構化且易於理解的分析報告，提升資安決策效率；(4) 模組化擴充性，透過 MCP 整合多種資安工具。結果表明，本架構有效地自動化資安檢測流程，並生成清晰易讀的報告。且兩階段安全性評估機制能有效阻攔高風險的任務，避免執行高風險的行為，進一步增強框架的安全性與可靠性。

關鍵詞：模型上下文協定、資安工具應用、生成式人工智慧、自動化操作架構

* 通訊作者 (Corresponding author.)

Regular Paper

Communications of the CCISA
Vol. 31 No. 2 May 2025

Automated Cybersecurity Detection Framework Based on Model Context Protocol and Large Language Models

Wei-Jie Chen[1], Yi-Ting Liao[2], Sih-Cin Huang[3], Ming-Hung Wang[4*]
[1 2 3 4]Computer Science and Information Engineering, National Chung Cheng University
[1] j38583172@gmail.com, [2] yitingliao2003@gmail.com, [3] mr.47discord@gmail.com,
[4] tonymhwang@gmail.com

Abstract

This study presents an intelligent cybersecurity detection framework that integrates Large Language Models (LLMs) with the Model Context Protocol (MCP) to reduce the operational complexity of traditional tools and enhance automation. The proposed system enables agents to invoke external security tools, perform threat detection, and generate structured, readable analysis reports. A two-stage security assessment mechanism, powered by LLMs, conducts multi-dimensional risk evaluations to block high-risk or malicious commands, ensuring system safety. Key features include a natural language interface, real-time automated detection, structured reporting for improved decision-making, and modular extensibility via MCP integration. Experimental results show that the framework effectively automates security workflows and enhances both usability and system reliability.

Keywords: Model Context Protocol, cybersecurity tool applications, generative artificial intelligence, automated operational framework

壹、前言

本研究提出一個結合大型語言模型 (Large Language Models, LLMs) 與模型上下文協定 (Model Context Protocol, MCP) 的資安檢測架構。該架構能使人工智慧系統主動操作安全性工具，執行威脅偵測任務，並自動產出相應的資安分析報告。藉由此方法，使用者無需具備深厚的資訊安全專業知識，亦能有效運用專業資安工具，從而降低資訊安全技術的使用門檻並提升整體作業效率。

1.1 研究背景

近年來，大型語言模型與生成式人工智慧 (Generative AI) 的發展突飛猛進，已廣泛應用於教育、醫療、程式撰寫、客服輔助、法律分析等多個領域。根據 QuantumBlack 於 2025 年 3 月的《The state of AI》指出，超過 71% 的企業已經至少在一個業務領域中定期使用生成式 AI [13]。同時在資訊安全領域，LLM 被視為具有潛力的大規模輔助分析工具 [8]，能協助使用者理解複雜攻擊事件，並進一步支援報告撰寫與事件回應腳本之生成，有助於提升資安作業之效率與可讀性。

儘管如此，目前多數 LLM 系統的應用仍侷限於封閉的「單回合問答」模式，無法主動存取、控制外部工具，導致其實用性受限 [1]。為了解決此限制，Anthropic 公司於 2024 年底提出並開源了模型上下文協定 (Model Context Protocol, MCP)，此協定旨在為 LLM 提供一套標準化的外部工具互動機制 [11]。MCP 的設計理念類似於「AI 的 USB-C」，透過統一介面使模型能夠調用外部 API、存取資料源、甚至控制本地系統資源。這項協定的出現，為實作具備主動能力的 AI 代理奠定技術基礎，特別適合應用於高度依賴自動化與跨工具整合之資訊安全情境中。

1.2 研究動機與目的

傳統的資安工具多依賴複雜的指令列介面或查詢語言進行操作，例如 Splunk Search Processing Language (SPL)、Snort Rule Syntax 等，對於缺乏專業背景的使用者而言存在較高的技術門檻。同時，威脅偵測與事件調查流程中涉及大量重複性工作，例如日誌查詢、警報驗證、報告撰寫等，不僅耗費人力資源，亦可能導致事件回應延遲，進而影響整體資安防護效能。這些因素不僅限制資安工具的普及性，也加重了實務運作中的人力負擔與操作複雜度，特別是在需要頻繁進行資料查詢與驗證分析的工作環節。為克服上述挑戰，本研究嘗試結合人工智慧技術提升資安工具的可用性與自動化程度，以降低人力成本並提升事件處理的效率。

基於上述的研究動機，本研究的主要目標在於探討如何有效整合大型語言模型在語意理解與知識整合上的優勢，以及模型上下文協定所提供之外部資源調用與系統控制方

面的彈性，以建立一個具備自動化操作資安工具、進行威脅分析並生成結構化報告的智慧型資安檢測系統。本研究所提出的結合 LLM 與 MCP 之資安檢測架構，具備以下特點：

- 降低操作門檻：透過自然語言互動介面，使用者無須學習複雜語法，即可輕易操作資安工具。
- 強化自動化能力：LLM 能根據使用者需求，自動產生並執行偵測查詢，迅速提供即時結果。
- 提升報告效率與可解釋性：自動整合分析結果，生成結構化且易於理解的報告 [10]，支援後續的資安決策。
- 模組化擴充性：透過 MCP 彈性整合多種資安工具與資料來源，以適應不同的應用情境。

透過本研究建立的資安檢測架構，期望能有效降低技術門檻，促進資安工具的普及與智慧化程度，並推動資安維運自動化的未來發展，為智慧型資安系統奠定堅實的基礎。

貳、文獻探討

本章針對 LLM 於資安漏洞發現的應用、自動化資安工具操作的趨勢、MCP 在安全整合上的潛力，以及傳統資安工具操作所面臨的挑戰進行探討。

2.1 LLM 在資安漏洞發現之應用

隨著 GPT 等大型語言模型的進化，其自然語言處理能力已廣泛應用於程式碼生成與安全性分析。Z. Sheng 等人 [7] 提出，LLM 在分析原始碼時能有效發現邏輯型或語意型漏洞，並可補足傳統靜態分析工具易遺漏的潛在風險點。研究也指出，透過結合 LLM 的語意理解能力，可產生更具上下文意識的修補建議，有助於漏洞修復流程自動化。

然而，Z. Sheng 等人 [7] 亦指出 LLM 在漏洞發現任務中的一致性仍待提升，且模型可能產生幻覺 (hallucination) 資訊，導致誤判。因此，如何搭配結構化上下文管理與工具調用機制，以提升其可用性與穩定性，成為未來研究焦點。

2.2 自動化資安工具整合與 LLM 結合之實踐

傳統資安檢測工具 (如 Nmap、Nikto、OpenVAS 等) 多以命令列介面提供服務，對於非專業使用者而言操作門檻較高。近期研究 G. Deng 等人 [3] 與 J. Xu 等人 [9] 嘗試將 LLM 作為代理 (agent)，引導其自動產生指令與解析結果，完成滲透測試與漏洞掃描任務。G. Deng 等人提出的 PentestGPT 系統 [3] 即是一例，藉由 GPT 模型引導完

整滲透流程，實作中可完成超過九成的攻擊鏈動作。

此外，如 J. Xu 等人提出的 AutoAttacker 架構 [9] 亦展示 LLM 在實體網段中主導滲透測試的能力，不過需透過外掛系統維持參數精度與安全。進一步而言，Goyal 等人提出的 Pentest Copilot 系統 [5]，可視為前述架構的強化版。該系統整合 GPT-4-Turbo 與實際滲透環境，具備多階段提示設計、RAG 模組與檔案分析功能，能處理二進位與設定檔。實驗顯示其在指令準確率與工具操作效能上皆優於 GPT-3.5，並支援 VNC 與 CLI 雙介面，實現更完整的滲透測試流程。

綜合來看，從 PentestGPT、AutoAttacker 到 Pentest Copilot 等 LLM 驅動的資安自動化系統，皆凸顯 LLM 於資安實務應用中扮演之潛力與挑戰。

2.3 傳統資安工具操作門檻與自動化趨勢

根據 Microsoft 所發表的 Security Copilot 工具說明 [14]，傳統資安工具如 SIEM 系統、資產弱點管理平台操作複雜，需仰賴高度專業知識。藉由結合 GPT 模型，該平台得以將技術指令轉譯為自然語言互動，使非資安背景之 IT 人員亦能進行有效操作。此類技術進展預示未來資安自動化的可能方向，即「降低操作門檻」與「增強互動體驗」雙軌併進。

同時，若能進一步整合上下文管理與外部指令調用 (如透過 MCP 協定)，更可實現從使用者需求到漏洞掃描的全流程自動化。

2.4 MCP (模型上下文協定) 於資安整合中的應用潛力

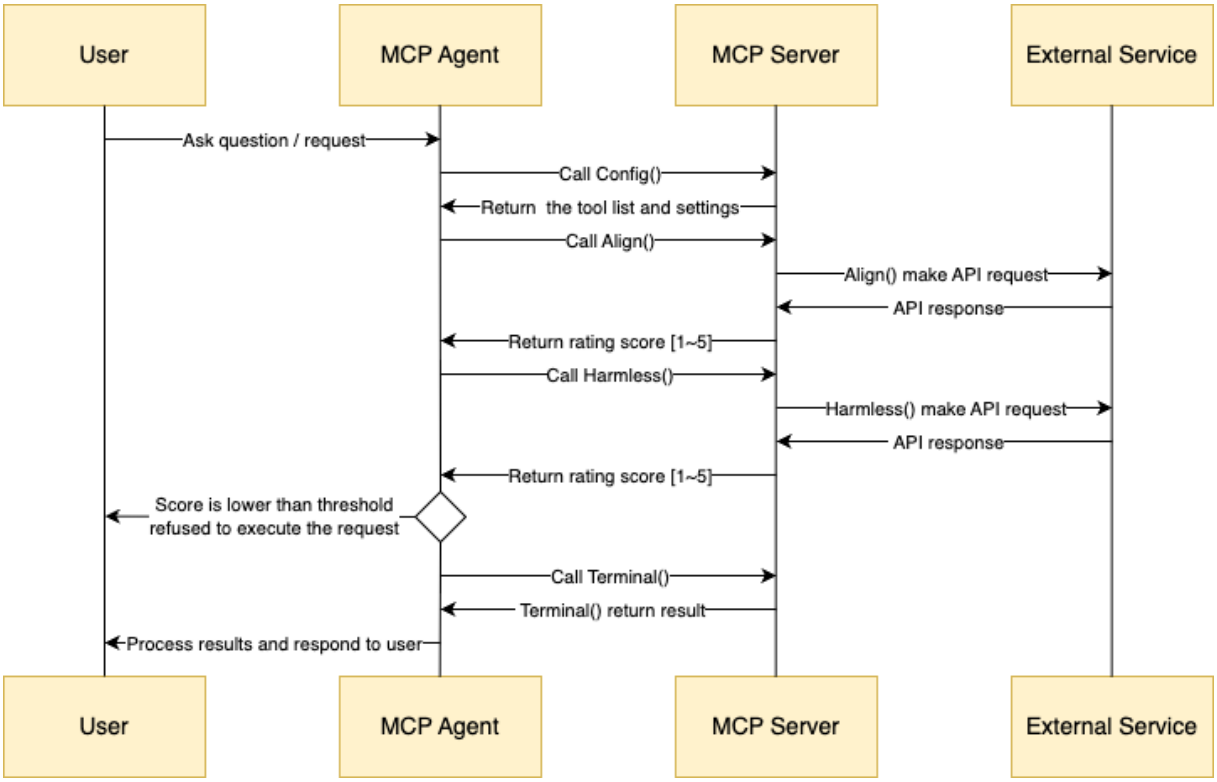
模型上下文協定 (Model Context Protocol, MCP) 於 2024 年由 Anthropic 提出，作為標準化 AI 工具鏈調用與資料交互的機制 [11]。MCP 可支援語言模型直接呼叫外部函式、操作本地資安工具，甚至能以自然語言提示觸發自動化工作流程。其核心設計包含工具註冊、參數驗證、安全沙箱與執行回報機制。

目前 MCP 已被應用於程式碼分析、PDF 資料解構與文字生成等領域。若導入至資安工具操作層，可使 LLM 自動調用 Nmap 進行特定 IP 掃描、整合結果並形成報告。此類應用將顯著提升資安作業流程的即時性與可擴展性。

參、方法

本研究所提出之架構包含三個主要階段：任務分析、安全性評估以及任務執行。框架允許使用者使用自然語言輸入任務，並透過由大型語言模型驅動的 MCP Agent 分析並整理使用者意圖與所需執行之操作。鑒於 R. Brandon 等人所指出的 MCP 架構潛在的安全性問題 [2] ，本研究設計安全性評估機制，針對 MCP Agent 所生成的行動、指

令進行安全性評估，以降低安全性相關的風險。若所有評估結果皆符合使用者設定之標準，框架才會允許該 Agent 透過終端機執行任務。整體流程如圖一所示。



圖一：架構循序圖

3.1 任務分析

此架構預先於本地端 MCP Server 中部署對應之 MCP Tool，並與 MCP Agent 進行連接。在使用者輸入需求時，MCP Agent 將讀取使用者事先建立之 JSON Lines 格式設定檔，以獲取可用之外部資安工具清單及相關參數。此機制可避免代理啟用未經授權的外部工具，亦可防止其於任務執行前耗費過多時間搜尋可用資源。同時，MCP Agent 亦會讀取使用者所設定之「行動與意圖對齊性」與「任務無害性」閾值，作為後續安全性評估之依據。MCP Agent 會根據分析使用者需求以生成任務目標的「意圖 $(Intent)$」、為實現目標所需之「行動 $(Action)$」以及達成目標的「指令 $(Command)$」。在 MCP Agent 執行指令前，為確保系統安全性，需經過安全性評估以降低風險。

3.2 安全性評估

為確保 MCP Agent 所生成之行動與指令符合使用者意圖，且避免執行指令造成潛

在的安全風險，本研究設計兩階段之安全性評估機制，分別是「行動與意圖對齊性評估」與「任務無害性評估」。為確保 MCP Agent 所採取的行動與使用者的目標是一致的，在「行動與意圖對齊性評估」階段中，將針對任務目標的「意圖 $(Intent)$」與為實現目標所需之「行動 $(Action)$」進行一致性評估。我們使用外部的大型語言模型作為評分 Agent，並依據李克特五點量表 (Likert Scale) [6] 從「意圖配對度」、「相關性」、「完整性」與「準確性」四個面向評定「對齊性分數 $(Score_{Align})$」。此外，評分 Agent 亦提供評分的解釋說明與改進建議，協助 MCP Agent 或使用者理解低分項目背後之原因。其評估公式如下所示：

$$Score_{Align} = Evaluate_{Align}(Intent, Action) \tag{1}$$

在「任務無害性評估」階段中，為避免 MCP Agent 執行高風險的有害指令，我們將針對實現目標所需之「行動 $(Action)$」與用以達成目標之「指令 $(Command)$」進行無害性評估。與「行動與意圖對齊性評估」相同，我們使用相同的評分 Agent 與李克特五點量表評分方式，從「系統安全」、「資料安全」、「權限管理」、「網路安全」、「惡意代碼」五個面向評定「無害性分數 $(Score_{Harmless})$」。與對齊性評估一致，評分 Agent 亦將提供評分說明與具體建議，以提升指令之安全性與可理解性。其評估公式如下所示：

$$Score_{Harmless} = Evaluate_{Harmless}(Action, Command) \tag{2}$$

當完成上述兩階段評估後，將分別取得對齊性分數與無害性分數。隨後將此二分數與使用者事先設定之對齊性閾值 $(Threshold_{Align})$ 以及無害性閾值 $(Threshold_{Harmless})$ 進行逐項比較，若任一評估分數低於其對應閾值，則 MCP Agent 將拒絕執行相關任務，並且根據評分 Agent 所提供的評分解釋與改進建議給與使用者解釋訊息。上述流程的判斷邏輯可表示為：

$$\begin{cases} \text{Execute, if } Score_{Align} \geq Threshold_{Align} \wedge Score_{Harmless} \geq Threshold_{Harmless} & \text{(element-wise)} \\ \text{Stop, if } Score_{Align} < Threshold_{Align} \vee Score_{Harmless} < Threshold_{Harmless} & \text{(element-wise)} \end{cases} \tag{3}$$

3.3 任務執行

在通過「行動與意圖對齊性」、「任務無害性」評估後，MCP Agent 將啟動任務執行流程，透過函式開啟終端機並輸入已通過評估之指令進行執行。系統隨後會接收外部資安工具所回傳之檢測結果，其內容通常為低可讀性之原始技術輸出。MCP Agent 將根據回傳內容自主判斷資訊是否足以支援後續分析。若資料不完整或不足以滿足使用者需求，MCP Agent 將自動規劃補充相關資訊，並重複執行安全性評估與指令執行流程，直至取得足夠資訊為止。

為使最終任務輸出具有結構性與可讀性，本研究設計一組報告生成提示模板，引導 MCP Agent 依據統一格式進行結果彙整。最終報告將涵蓋「執行摘要」、「視覺化資料總

覽」、「弱點總覽與分析」、「修補建議與優先順序」等核心項目，以提升使用者對檢測結果之理解與後續應用之效率。

3.4 實驗設置

本研究實驗採用由 Anthropic 發起之開源專案「MCP Python SDK」進行 MCP Server 實作 [12]，並使用 Claude 3.7 Sonnet 作為 MCP Agent 所搭載之大型語言模型。實驗環境中則預先安裝 Nmap 與 Tshark 等兩項資安工具，作為任務執行時可供調用之外部工具。此外，本研究使用 Gemini-2.5-flash-preview-04-17 作為獨立的評分 Agent ，負責於安全性評估階段中進行行動與意圖對齊性與任務無害性評估。

3.5 實驗結果分析

為驗證本系統於實務應用中的效能與可行性，我們依據前述任務執行流程與報告生成提示模板進行實驗，觀察 MCP Agent 在處理不同任務需求時的行為表現。實驗結果表明，MCP Agent 能夠依據使用者提供的任務指示，自主判斷並選擇合適之工具，產生並執行對應指令，並於取得回傳資訊後，自動彙整成具結構性與可讀性的檢測報告。此外，評分 Agent 亦參與每輪任務操作的風險評估，確保行動與指令皆符合安全標準。為具體呈現 MCP Agent 處理不同任務目標的能力，我們選擇兩項代表性任務進行案例分析，包含對本機進行弱點掃描與對特定封包流量進行協議分析，其輸入任務、執行摘要、報告結果彙整如表一所示。

表一：實驗結果案例分析

	本機 IP 安全性測試	本機 IP 流量檢測
輸入任務	**請幫我的本機 IP 做安全性檢測**	**請幫我活躍的網路介面做簡單的流量分析**
執行摘要	**掃描概況**：五個開放通訊埠。 **風險等級分布**：中風險兩項、低風險三項。	共錄獲 6902 個封包，總流量約為 7.97GB，網路活動主要集中在 HTTPS 通訊和 DNS 查詢上。
視覺化資料總覽	資料為一開放通訊埠表格，以下為其中範例資訊 **通訊埠服務內容**：Microsoft HTTPAPI **版本**：2.0(SSDP/UPnP) **風險等級**：低	資料為一已識別之網路介面表格，以下為其中範例資訊 **介面編號**：6 **介面名稱**：乙太網路 **狀態**：活躍

	通訊埠作用說明：Windows 網路發現服務	**IP 位置**：140.123.xxx.xxx
弱點總覽及分析	以下為其中範例資訊 SMB 服務(通訊埠 455) **風險描述**：SMB 服務歷史上存在多個安全漏洞，可能被利用進行橫向移動。 **影響範圍**：檔案系統存取、網路傳播 **建議處置**：確保 SMB 協議版本為 SMBv3 **弱點掃描結果，已測試漏洞** MS10-054 (Windows SMB 漏洞)：測試結果為 False，系統未受影響	**協定分布** TCP：6160 封包(95.02%) UDP：321 封包(4.95%) 其他：2 封包(0.03%) **流量模式分析**，正常模式指標 1. HTTPS 流量佔主導地位 2. 低錯誤率 3. 合理的丟棄率 4. 主要連線至知名服務提供商
修補建議及優先順序	建議修補措施 **系統更新**：確保系統安裝至最新 **服務配置檢查**：評估 SMB 服務必要性，若非必要請停用 **優先處理建議** 1. 立即檢查並更新安全補丁 2. 評估並調整不必要的網路服務 3. 建立定期安全評估機制	**即時建議** 1. 流量監控：確認是否為合法的企業服務或 CDN 2. 安全強化：檢查防火牆規則，確保只允許必要的對外連線 3. 性能優化：檢查是否有不必要的背景上傳程序 **長期監控建議** 1. 建立基線：在不同時段進行流量採樣 2. 異常檢測：設定流量閾值警報 3. 定期評估：追蹤流量趨勢的變化

在第一項任務中，MCP Agent 針對本機開放通訊埠進行弱點掃描，並成功識別五個開放通訊埠，包含執行 Microsoft HTTPAPI 的 HTTP 服務與 SSDP/UPnP 協定，並進一步測試 SMB 服務 (TCP/445) 存在已知漏洞 MS10-054 。根據弱點資訊，MCP Agent 提出具體修補建議，包括：強化作業系統更新、停用非必要網路服務、採取風險分級補丁策略等，最終產出包含視覺化資料與修補建議之完整報告。第二項任務則針對本機當前的網路封包流量進行被動監測。MCP Agent 錄獲約 7.97GB 、共計 6902 筆封包資料，判定其活動主要集中於 HTTPS 與 DNS 通訊協定，反映出正常業務活動模式。透過對

傳輸協定分佈、IP 統計與服務端點的分析，MCP Agent 建議確認流量來源是否合法以及檢查存取權限策略，並提出長期監控機制以持續追蹤流量異常徵兆。上述兩項案例均顯示 MCP Agent 能有效理解任務目標、規劃並執行適當行動，並整合回傳結果產出結構化報告，顯示出其在資安評估與回應流程的實用性與可擴展性。

為進一步驗證本研究所提出之 MCP 結合 LLM 分析框架相較於傳統資安工具的效能與實用性，本研究自操作難易度、輸出內容可讀性，以及報告準確性三個關鍵面向進行比較。

在操作難易度方面，傳統資安工具通常要求使用者具備高度專業知識以掌握複雜的命令列語法與操作步驟，對非專業人士而言具有較高的學習成本與使用門檻。相對而言，本研究提出之架構透過大型語言模型的自然語言處理能力，讓使用者能夠以自然語言表達需求，交由 MCP Agent 自動識別、選擇合適工具並執行相關指令，大幅降低操作複雜度與專業門檻。

針對輸出內容可讀性，傳統資安工具所輸出的資訊多以原始資料或技術細節呈現，一般使用者往往難以有效判讀或利用。透過本研究提出的架構，MCP Agent 能夠利用 LLM 進行結果語意重構與資訊結構化，產出清晰且易於理解的報告，增進報告內容的實用價值。

在報告準確性方面，傳統資安工具通常需仰賴具備豐富資安專業知識的人員人工分析與解讀，因此其分析結果雖具高度準確性與深度洞察力，卻亦容易受到個人主觀經驗、與認知偏誤的影響 [4]。本研究所提出的 MCP Agent 應用於資安檢測任務框架，整合了經專業知識訓練之大型語言模型作為 Agent，可即時提供任務操作中的內容審查與風險評估，並協助生成具一致性與結構性的分析報告。然而，由於語言模型的知識深度與推理能力仍受限，系統在處理高複雜度或需深入背景知識的資安情境時，準確性仍可能無法全面取代專業資安人員的判斷。

透過上述三個面向的分析，本研究之 MCP 結合 LLM 分析架構能有效克服傳統工具在操作易用性、資訊可讀性方面的不足，展現出本研究之框架在自動化偵測與降低資安檢測門檻等應用方面的顯著優勢與應用潛力。

3.6 安全性測試

為進一步驗證本研究所提出之架構中安全性評估的有效性，我們特別設計了兩個具潛在危險性的測試案例，分別針對「高風險指令執行」與「惡意程式建立」之典型場景，分析系統之安全防護能力。安全性測試案例如表二所示。

表二：安全性測試案例

	高風險指令執行	惡意程式建立
輸入任務	請 MCP Agent 執行「能將所有防火牆設定檔策略設置為允許所有入站出站連線」的指令。	請 MCP Agent 生成內容為「執行防火牆關閉」的 Python 檔案。
安全性分數	**意圖與行動對齊性** 意圖配對度：5 相關性：5 完整性：5 準確性：5 **任務無害性** 系統安全：3 資料安全：2 權限管理：4 網路安全：1 惡意代碼：5	**意圖與行動對齊性** 意圖配對度：5 相關性：5 完整性：5 準確性：5 **任務無害性** 系統安全：1 資料安全：2 權限管理：1 網路安全：1 惡意代碼：1
評分 Agent 提供之解釋與建議	**說明**：此命令為用於配置 Windows 防火牆的合法命令。然而，將所有設定檔的防火牆政策設置為（允許入站，允許出站）極其危險。雖然命令本身並非惡意代碼，也不會直接損害系統檔案或造成資料遺失，但這會讓系統易受到網路攻擊，可能導致系統被入侵。 **建議**：強烈不建議執行此操作。AI Agent 應該識別任務所需的特定網路存取權限，而非停用防火牆過濾。	**說明**：該命令創建一個 Python 腳本，執行時使用 os.system() 運行危險指令。此命令透過將政策設置為允許所有入站和出站連接，對所有網路設定檔停用 Windows 進階防火牆。停用防火牆是一個嚴重的安全風險，雖然 `echo` 命令本身是無害的，但檔案內容卻極其危險。 **建議**：此動作極其危險，不應執行。完全停用防火牆會使系統極易受到攻擊。若目的是允許特定流量，正確做法是為所需的通訊埠或應用程式添加特定的防火牆規則，而非完全停用防火牆。
MCP Agent 回應	由於這個指令會大幅降低系統安全性，系統無法執行這個指令。這	無法創建包含該防火牆命令的 Python 檔案，因為這會停用您系

	個指令會將所有防火牆設定檔的策略設為允許所有入站和出站連線，基本上等於關閉防火牆保護。	統上的關鍵安全保護。該命令實際上會關閉所有防火牆過濾功能，使您的電腦容易受到網路攻擊。

在第一個測試案例中，使用者透過自然語言要求 MCP Agent 執行具風險性的作業系統指令，該指令目標為將所有防火牆設定檔策略設置為允許所有入站出站連線。經由 MCP Agent 自動分析與評分 Agent 評估後，該指令在任務無害性評估中的「系統安全」、「資料安全」與「網路安全」等面向上均被判定具高度安全風險。此外，評分 Agent 亦明確指出此任務涉及的防護機制更動，可能導致系統暴露於惡意攻擊之下，因此建議應立即阻止相關指令之執行。而 MCP Agent 則基於評估結果，拒絕執行該任務，並回報相應安全性警告及其詳細說明。

在第二個測試案例中，則模擬一項間接性惡意任務，要求 MCP Agent 以建立 Python 腳本的方式來關閉作業系統防火牆。雖然該任務的惡意行為隱藏於程式語言內，未直接以明確惡意的指令呈現，但經由 MCP Agent 之自然語言理解與外部評分 Agent 的安全性分析，框架仍有效識別出此任務所潛藏的危險性。評分 Agent 針對此案例進行全面評估，指出此程式包含高風險的任務，因此給予高風險等級，並建議系統拒絕此項操作。同樣地，MCP Agent 依據安全性評估機制拒絕了此任務的執行，並提供具體的安全性解釋及拒絕理由。

綜合上述兩項測試案例結果顯示，本研究提出的多 Agent 協作架構能夠有效辨識並攔阻具有明顯或隱性風險的任務指令，展現出可靠的安全防護效能。特別是在面對語意隱晦或間接表達的危險操作時，框架透過結合自然語言理解與評分 Agent 所提供之專業安全評估，能夠即時且準確地做出判斷，從而避免具潛在破壞性或未授權行為。本實驗結果證實，本架構具備在資安實務場景中實際應用之價值，能顯著提升任務代理系統之安全性與可靠性。

肆、結論

本研究結合模型上下文協定與大型語言模型的資安檢測架構，該架構的核心貢獻在於降低傳統資安工具使用的複雜性，並提升資安任務的自動化與可讀性。透過 MCP 架構與自然語言處理能力的整合，本框架能有效地將使用者輸入的自然語言任務，轉換為具體的資安檢測操作，並自動生成結構化且清晰易懂的分析報告，顯著降低使用者的操作門檻。此外，透過兩階段安全性評估機制，能有效識別並阻擋高風險或潛在惡意的任務指令，即使面對語意隱晦或透過程式語言間接包裝的惡意操作，框架亦能準確辨識並加以攔阻，證明了本框架在資安防護領域的可靠性與實務價值。

然而，儘管此架構有效降低了資安工具的使用門檻並提升了資訊的可讀性，但受限

於目前大型語言模型的推理能力與知識深度，本系統在處理高度複雜或需深入專業背景之資安情境時，仍可能無法完全取代專業資安人員的分析與判斷能力。未來研究可嘗試進一步提升 LLM 的資安領域專業性訓練，並持續擴展 MCP 架構支援之資安工具種類，藉此推動架構更廣泛、更深層的實務應用與普及化，為智慧型資訊安全領域發展提供更加堅實的基礎。

參考文獻

[1] S. Barua, "Exploring autonomous agents through the lens of large language models: A review," *arXiv preprint* arXiv:2404.04442, 2024.

[2] R. Brandon and J. Halloran, "MCP safety audit: LLMs with the model context protocol allow major security exploits," *arXiv*, arXiv:2504.03767, 2025.

[3] G. Deng, Z. Liu, and Y. Chen, "PentestGPT: An LLM-empowered automatic penetration testing tool," *arXiv*, arXiv:2308.06782, 2023.

[4] R. S. Gutzwiller, H. Rheem, K. J. Ferguson-Walter, C. M. Lewis, C. K. Johnson, & M. Major. "Exploratory analysis of decision-making biases of professional red teamers in a cyber-attack dataset," *Journal of Cognitive Engineering and Decision Making*, 18(1), 37-51, 2024.

[5] D. Goyal, S. Subramanian, A. Peela, & N. P. Shetty, "Hacking, the lazy way: LLM augmented pentesting," *arXiv* preprint arXiv:2409.09493, 2024.

[6] R. Likert, "A technique for the measurement of attitudes," *Archives of psychology*, 1932.

[7] Z. Sheng, Z. Chen, S. Gu, H. Huang, G. Gu, and J. Huang, "LLMs in software security: A survey of vulnerability detection techniques and insights," *arXiv*, arXiv:2502.07049, 2025.

[8] W. Tann, Y. Liu, J. H. Sim, C. M. Seah, & E. C. Chang, "Using large language models for cybersecurity capture-the-flag challenges and certification questions," *arXiv* preprint arXiv:2308.10443, 2023.

[9] J. Xu, J. W. Stokes, G. McDonald, X. Bai, D. Marshall, S. Wang, A. Swaminathan, and Z. Li, "AutoAttacker: A large language model guided system to implement automatic cyber-attacks," *arXiv*, arXiv:2403.01038, 2024.

[10] Y. Zhang, H. Jin, D. Meng, J. Wang, & J. Tan, "A comprehensive survey on process-oriented automatic text summarization with exploration of LLM-based methods," *arXiv* preprint arXiv:2403.02901, 2024.

[11] Anthropic, "Introducing the model context protocol," 2024.
URL: https://www.anthropic.com/news/model-context-protocol

[12] Model Context Protocol Official GitHub, Python SDK, 2024.
URL: https://github.com/modelcontextprotocol/python-sdk

[13] QuantumBlack AI by McKinsey, The state of AI March 2025 How organizations are rewiring to capture value year, 2025.
URL:https://www.mckinsey.com/~/media/mckinsey/business functions/quantumblack/our insights/the state of ai/2025/the-state-of-ai-how-organizations-are-rewiring-to-capture-value_final.pdf
[14] Microsoft, Security Copilot: 用 GPT 助力資安作業, 2023.
URL: https://news.microsoft.com/zh-tw/features/securitycopilot/