Regular Paper

Communications of the CCISA
Vol. 30 No. 1 Feb. 2024

惡意攻擊主動防禦框架

謝毅民[1]、曾國鈞[2*]
國立宜蘭大學資訊工程學系
[1]erichsieh124@gmail.com、[2]kctseng@niu.edu.tw

摘要

近年來隨著網路的發展，5G 網路、雲端服務，科技的發展下，企業網路型態發生的變化，為了使用者的方便，公司資源必須讓使用者能夠存取，使得公司企業的網路邊界變得更加的薄弱，導致駭客可以利用身分識別的漏洞缺失，從而進行網路中的攻擊，當攻擊行為多樣化難以預防，資訊安全防護已然成為當務之急。

為有效防範網路入侵事件，本研究提出一種自動化防火牆規則生成方法，在企業現有環境中利用既有資源，以不增加企業預算又能加強保或企業抵禦網路攻擊為目的，該方法主要透過分析事件日誌 Windows Event Log，與 Palo alto 次世代防火牆 Log 日誌，在偵測到重要安全事件威脅時，能夠即時找出遭受攻擊的來源 IP，並自動在防火牆中新增規則，封鎖該 IP 的網路存取。

此框架系統架構將持續監控事件日誌，一旦任意的偵測演算法偵測到入侵跡象，例如試圖存取系統檔案或暴力破解登入等，將立即萃取事件相關資訊，透過分析事發前後的事件紀錄，尋找造成該事件的來源，並擷取相關網路資訊，例如目的地 IP、連接埠號等。最終，系統將自動編輯防火牆規則，封鎖具威脅性的 IP，以達到主動防禦[1]的效果。相較於傳統需要人工分析事件日誌才能新增防火牆規則的方式，本研究的自動化方法能大幅降低因人工作業延遲產生的風險，提高防護反應的效率，並節省企業營運成本。未來，這個框架系統可導入任何日誌系統，並採用各種偵測演算法，以不斷改良演算法、提升事件偵測和規則生成的正確性，建構更完善的網路防護機制。

關鍵詞：網路安全、事件日誌分析、入侵偵測、主動防禦

* 通訊作者 (Corresponding author.)

Regular Paper

Communications of the CCISA
Vol. 30 No. 1 Feb. 2024

Active defense framework for malicious attacks

Eric Hsieh[1], Kuo-Chun Tseng[2*]
Department of Computer Science and Information Engineering, National Ilan University
[1]EricHsieh124@gmail.com, [2]kctseng@niu.edu.tw

Abstract

In recent years, with the advancement of technology, including the development of 5G networks and cloud services, enterprise network structure has transformed. To facilitate user convenience, corporate resources must be made accessible, leading to a thinner boundary in the company's network infrastructure.This increased accessibility, however, exposes vulnerabilities in identity authentication, providing opportunities for hackers to exploit and carry out cyberattacks. Information security defense has become an urgent priority with the proliferation of diverse attack strategies.

This study proposes an automated firewall rule generation method to effectively prevent network intrusion incidents. This method leverages existing resources within the enterprise environment without increasing the corporate budget, aiming to strengthen defenses against network attacks. It primarily involves analyzing event logs such as Windows Event Log and Palo Alto Next-Generation Firewall Log. Upon detecting critical security threats, this method promptly identifies the source IP under attack and automatically adds rules in the firewall to block network access from that IP.

This framework system architecture continuously monitors event logs. Once any detection algorithm detects signs of intrusion, such as attempted access to system files or brute-force login attempts, it immediately extracts relevant event information. Analyzing the event records before and after the incident identifies the event's origin and captures pertinent network information, such as destination IP addresses and port numbers. Ultimately, the system automatically configures firewall rules to block threatening IPs, ensuring proactive defense. Compared to traditional approaches requiring manual analysis of event logs to add firewall rules, this automated method significantly reduces the risk caused by human operational delays, enhances response efficiency, and saves on operational costs for enterprises. Moving forward, this framework system can be integrated with various log systems and utilize diverse detection algorithms to continually improve algorithms and enhance event detection and rule generation accuracy, thereby constructing a more robust network protection mechanism.

Keywords: Network Security, Event Log Analysis, Intrusion Detection, Proactive Defense

壹、前言

隨著資訊科技的快速發展,網路病毒和駭客入侵手法層出不窮,傳統的防毒軟體已經難以有效防範所有威脅，因應安全資訊和事件管理 (SIEM) [2]，許多企業紛紛決定導入入侵偵測系統,以強化其資訊安全防護能力,這些偵測與預警系統成為企業的必要工具，能夠及早發現潛在的安全風險並快速做出應對，企業在遵循政府相關法規的同時，導入部署這些系統，不僅能有效防範資安風險，也有助於提高整體資訊安全管理的效能，然而一般企業要導入的入侵偵測系統 (Intrusion Detection System，IDS) [3]，如 Darktrace[4]、ForeScout[5]用於監視網路或系統的活動，提升網路行為可視性，透過即時監控和分析來識別並應對潛在的惡意活動、攻擊或異常行為，這些平台能夠自動將威脅根據其優先級進行排序，優先處理最具威脅性和緊迫性的事件，從而提高對安全風險的應對效率，效果固然很好，但對於小型企業而言，這些系統的價格相對較高，操作和分析也需專業資訊人才，若無專人進行管理與控管，將難以發揮實際效果，對於中小企業而言，用在資安防禦上的預算相對較少，負擔較為沉重。

根據 Asia Spam-message Research Center (ASRC) 2023[6]年第三季電子郵件安全觀察，釣魚郵件仍是最主要的攻擊，一般釣魚郵件的常見特徵包括顯示的連結與實際前往的連結不一致，攻擊者也能對郵件內惡意連結中的域名做其他變更，例如將小寫字母切換為大寫字母，或插入不可見字元等，嘗試透過社交工程的手法，誘使企業成員安裝遠端控制軟體和後門軟體，但在郵件中完全未提及此類軟體的資訊，巧妙地將遙控軟體及後門工具放置於外部連結，避開郵件掃描的檢測，可以將郵件順利地送達到使用者的郵件信箱，又因使用資安意識較為薄弱時，可能導致攻擊者後門程式進入企業內部網路架構中進行資料的竊取或者進行側錄特權帳號的提取，進而爆發更加嚴重的資安事件，在病毒與網路攻擊行為越來越多樣化下，要如何加強防禦入侵的攻擊行為上和降低企業在資訊安全防護的門檻上，如何取得一個平衡點，一直都是企業中資安人員努力的方向，

鑒於現今網路環境，郵件社交攻擊[7]仍佔大多數攻擊手法，且發生事件爆發時皆已在企業內部爆發開來，因此本研究中我們著重於系統日誌的分析與 Palo alto (PA) 防火牆對於企業內部橫向流量的日誌警示，使用市面上較為容易安裝的 Elastic Stack[8]開源軟體平台，結合安全事件日誌，來實現網路入侵的自動防禦，達到主動防禦的目的，相較於手動分析日誌再人為的處理事件應變中，此研究框架方法利用 Elastic Stack 收集日誌，由此防禦平台從中擷取日誌分析[9]後，進行主動防禦動作，將可以大幅降低反應時間，提升入侵防禦的效率，以減少安全事件的發生概率，未來將能夠整合高效的偵測演算法，透過分析提供的所有日誌 (log) 資料，進行評估判斷是否需要採取阻擋措施，只要偵測演算法的精準度提高，我們將能夠有效地利用這系統框架來阻擋相應的行為。

貳、文獻探討

企業網路防禦中，系統日誌 (LOG) 扮演了關鍵角色，日誌紀錄了企業網路環境中所發生的事件，提供了關於使用者和系統之間行為的詳盡記錄。透過日誌分析，我們能夠更進一步地識別是否存在不正常的活動或者異常的登入登出[10]行為，然而在企業網路環境中，Windows 事件檢視器的功能受限制，它僅能監控和檢視本機的事件日誌，難以擴展到整個網路環境，這種局限性導致在事件分析、跨系統日誌分析方面存在著限制，特別是在處理大量事件日誌並進行分析時，其效率和能力明顯不足，難以滿足大型環境的需求，為了彌補這些限制，導入跨平台的 Elastic Stack 軟體系統，這個系統不僅適用於 Windows，還能用於整個網路環境的事件日誌收集和分析，另外透過 Microsoft Learn Windows EventLog 附錄中對事件的說明，我們可以識別出與使用者行為中異常情況相關的事件識別碼，這將有助於作為防禦平台架構施行的依據。

2.1 Threat Hunting Using Elastic Stack: An Evaluation[11]

在 2021 年，Subramanian 等人提出關於探討了使用 Elastic Stack 軟體工具進行威脅防護的效果，Elastic Stack 是一套開源的日誌分析和搜索引擎工具，包含 Elasticsearch、Logstash 和 Kibana 三個組件，Elasticsearch 是一個開源的分佈式搜尋和分析引擎，它能夠存儲、搜索和分析大量資料，Elasticsearch 通常用於支援各種日誌分析和全文檢索的案例。Logstash 是一個開源的服務器端資料處理管道，它能夠同時從多個來源擷取資料，對資料進行處理、轉換，然後將資料送往一個或多個目的地，在 Elastic Stack 中，Logstash 通常用於從各種來源中收集日誌，並將這些日誌統一格式後發送到 Elasticsearch 中。Kibana 是 Elasticsearch 的開源視覺化平台，它可以用於對 Elasticsearch 索引中的資料進行搜尋、視覺化與分析，在 Elastic Stack 中，Kibana 使得人們可以方便地檢索、分析和視覺化存儲在 Elasticsearch 中的大量日誌資料。

在此論文研究中使用 Elastic Stack 收集事件日誌，以達到分析日誌的行為，作者通過實際安裝佈置 Elastic Stack 並進行多種攻擊測試，比較評估了它在日誌監控和威脅偵測方面的能力。相關研究方面，作者提到 Elastic Stack 已經被用於 IT 監控和企業架構設計，通過 Elastic Stack 為中小企業建立了安全日誌分析系統，闡述了如何通過 Elastic Stack 的日誌記錄和監控來維護網絡安全，建立了基於 Elastic Stack 的 Kibana 使用訊息可視化平台。

作者的主要貢獻是全面地評估和比較 Elastic Stack 在威脅防護方面的效果，實驗部分，作者通過 Hydra、XHydra[12]等工具模擬了暴力破解、字典攻擊、DDoS 攻擊、社會工程學等多種攻擊，並使用 Auditbeat、Winlogbeat 等代理收集相關日誌，最後在 Kibana 中進行分析和威脅偵測，結果顯示，利用 Elastic Stack 可以有效地識別這些攻擊的跡象，此外，作者還根據 9 個評估指標與 HP ArcSight、IBM QRadar、LogPoint 和 Splunk 等 4

種類似工具進行了比較，結果顯示 Elastic Stack 在大規模環境下具有更好的擴展性和成本效益。

Elastic Stack 作為一個集成的日誌管理和分析工具，提供了開源、高效能、功能豐富和可擴展的解決方案，市面上有很多付費軟體在某些方面可能有些特定的優勢，但對於一些小型組織來說，Elastic Stack 是一個可以考慮用來取代付費軟體的優秀選擇。如圖一所示：

圖一：Elastic Stack 平台流程圖

2.2 Endpoint log analysis and anomaly detection[13]

於 2017 年 Chan-Chi Yeh 提出日誌事件的記錄分析對企業和組織的資訊安全至關重要，因為透過這些事件可以獲取關鍵資訊，有助於發現系統問題、政策違規、內部威脅等潛在風險，這些分析有助於及早偵測攻擊，從而降低安全風險，這項研究旨在分析 Windows 作業系統主機端點的日誌事件，以探測使用者行為中可能存在的異常情況。研究的目的在於提出針對 Windows 日誌格式設計的專用異常檢測方法。

研究中，作者參考了多項相關工作，包括 OSSEC[14]入侵偵測系統、從日誌消息中擷取有用資訊的方法，以及對 Windows 日誌事件的可靠性分析。這些先研究為此論文提供了堅實的基礎和啟發，為了達成此目的，作者提出了一種針對 Windows 日誌事件格式的專用方法，首先篩選了重要的日誌事件類型，並為每種事件建立了模式，以擷取描述字段中的關鍵資訊，最後，發展了一種異常檢測演算法，創建了代表使用者行為的特徵向量，以便在時間序列中檢測異常行為。

作者為了評估方法的效果，使用個人電腦的日誌事件數據進行了實驗。通過在數據中植入三種不同的異常行為，評估了演算法的檢測能力，結果顯示，該演算法能夠有效檢測這些異常行為，且僅有 2%的誤報率，此論文主要提出了針對 Windows 日誌格式的專用異常檢測演算法，在此顯現出來 Windows 日誌分析對於防禦入侵的重要性。

2.3 Zero Trust Architecture[15]

2020 年 8 月，美國國家標準與技術研究院 (NIST) 發布了 SP 800-207 文件，對於零信任架構 (Zero Trust Architecture，ZTA) 進行了說明和應用指南，ZTA 是一種資訊安全架構，其核心理念是在任何情況下都不要信任內部或外部的用戶、系統或服務，並要求進行驗證和授權所有的訪問和操作，SP 800-207 文件提供了詳盡的指南，意在幫助企業組織人員理解、設計和實施零信任架構，強調了企業實施 ZTA 後的持續監控和評估的重要性。

ZTA 顛覆了傳統的安全模型，強調在任何情況下都不信任內部或外部的用戶、系統或服務。這樣的轉變使得不再依賴單一的安全邊界，而是將安全性置於每個訪問和操作的核心，文件強調了驗證和授權所有訪問和操作的重要性。這意味著即使是內部用戶，也需要進行驗證，並僅在必要時獲得授權才能訪問敏感資源，ZTA 推崇微分授權和最小權限原則，意味著每個用戶或系統只能獲得訪問所需資源的權限，以減少攻擊表面。

文件強調了連續性驗證和持續監控的重要性，ZTA 強調，安全性不應僅是一次性的，而應持續地監控、驗證和評估，以應對不斷變化的威脅，SP 800-207 對於 ZTA 提供了深入的指導，幫助企業和組織理解、設計和實施這種安全架構，它強調了不信任的基本概念，並提出了實踐 ZTA 所需的驗證、授權、最小權限和持續監控等關鍵原則，以提高系統的安全性和抵禦入侵的能力。

參、方法

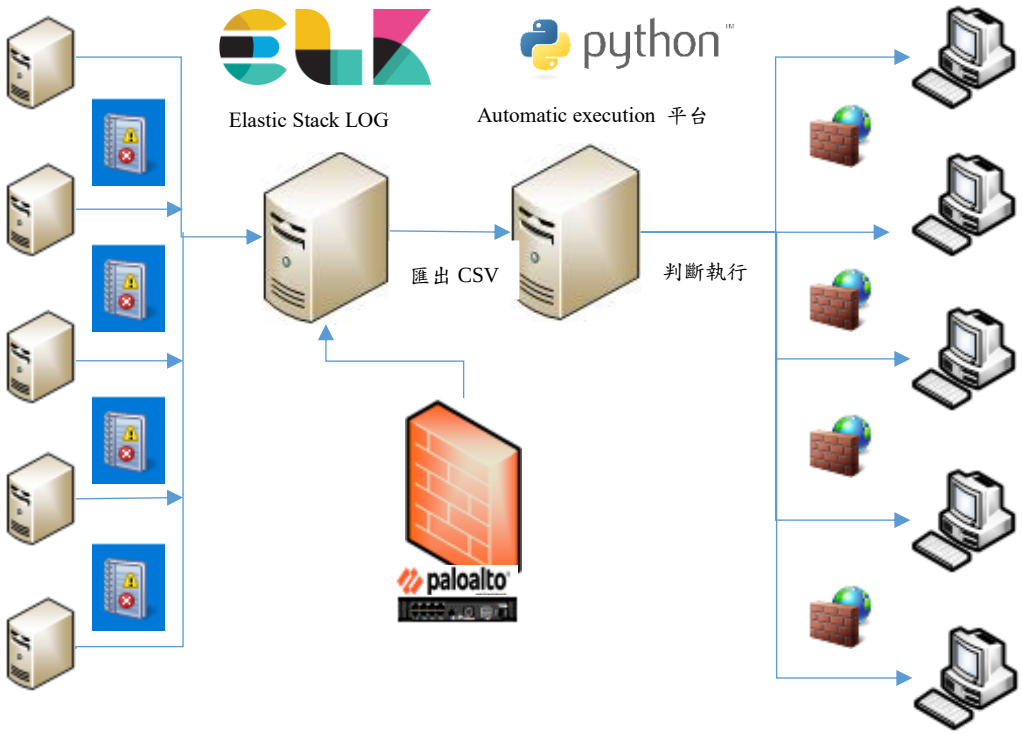
3.1 系統框架架構

研究框架將重點運用 Elastic Stack 蒐集資料與 Logstash Csv output plugin 插件[16]，從 Windows 日誌或其他系統的記錄中提取資料，進行進一步的分析，利用設定防禦平台，能夠定制特定關鍵字觸發條件 (使用 Python 語法)，並且在特定事件發生時，即時發送指令至可能發生問題的電腦，例如關閉其防火牆，以實現自動化的即時防禦阻擋災害發生，在過去的日誌分析研究中，大多僅限於日誌的顯示和相關警告通知，缺乏在事件發生時即時防禦的能力，因此，本研究框架的重點在於利用相關工具和軟體來實現即時阻斷事件發生的行動，提高安全監控的效能，透過這項研究框架，希望展示一種更進階、更有效的方法，使得系統能夠不僅僅是分析日誌，更能在事件發生時採取及時的防禦措施，這樣的能力對於應對日益複雜的安全威脅和快速回應安全事件具有重要意義。

本研究框架選擇了 Linux Ubuntu 作為主要系統平台，這是因為 Linux 具有開源性和穩定性，適合長時間運行，提供更為彈性的系統，穩定性也比 Windows 系統穩定，在過去的日誌管理中，許多研究可能面臨了一些問題，缺乏一個集成且高效的系統來收集和管理不同系統的日誌數據，因此收集日誌系統方面我們選擇了 Elastic Stack 平台，作為我們的日誌管理平台，其中 Elasticsearch、Logstash 和 Kibana 都是開源軟體，提供了

系統靈活性和可擴展性，允許根據需求自定義環境和擴展系統，透過使用 Logstash 進行日誌資料收集、處理和轉發工具，可以從多個來源收集資料系統進行預定收集各系統的日誌，此研究系統中數據的主要來源包括 Windows Server 系統中部屬 Active Director (AD) [17]角色系統的 EventLog 和 Palo Alto 防火牆[18]流量及威脅的 LOG 數據，利用 Elastic Stack 平台 Logstash Csv output plugin 插件匯出已定義的數據轉換成 CSV 檔，CSV 是一種常見的表格數據格式，易於讀取和使用，這使得匯出的數據能夠被廣泛支持的軟體和工具使用，在此研究中 Logstash To CSV 處理和轉換的數據，進行結構化和整理後，能夠迅速地被處理。

此系統框架擁有高度的可擴展性，確保使用最有效的演算法來處理日誌數據，定義是能夠將任何的系統日誌資料，套入到系統框架中的 Python Script 作為工具來處理日誌數據，Python 擁有豐富的模組和強大的功能，能夠讀取、分析和篩選大量的日誌數據，對於檢測和辨識疑似攻擊事件，Python 展現出極佳的能力，透過 Python Script，我們能夠精確地篩選出疑似遭受攻擊事件的目標，系統框架可設定自動排程，定期執行分析操作，將有威脅問題的電腦透過 Script 的自動執行，透過主動防禦自動添加防火牆規則，及時阻斷網路災害擴散的可能性，使得系統框架能夠在不斷演進的環境中保持高效、靈活且具有更新能力。如圖二所示：



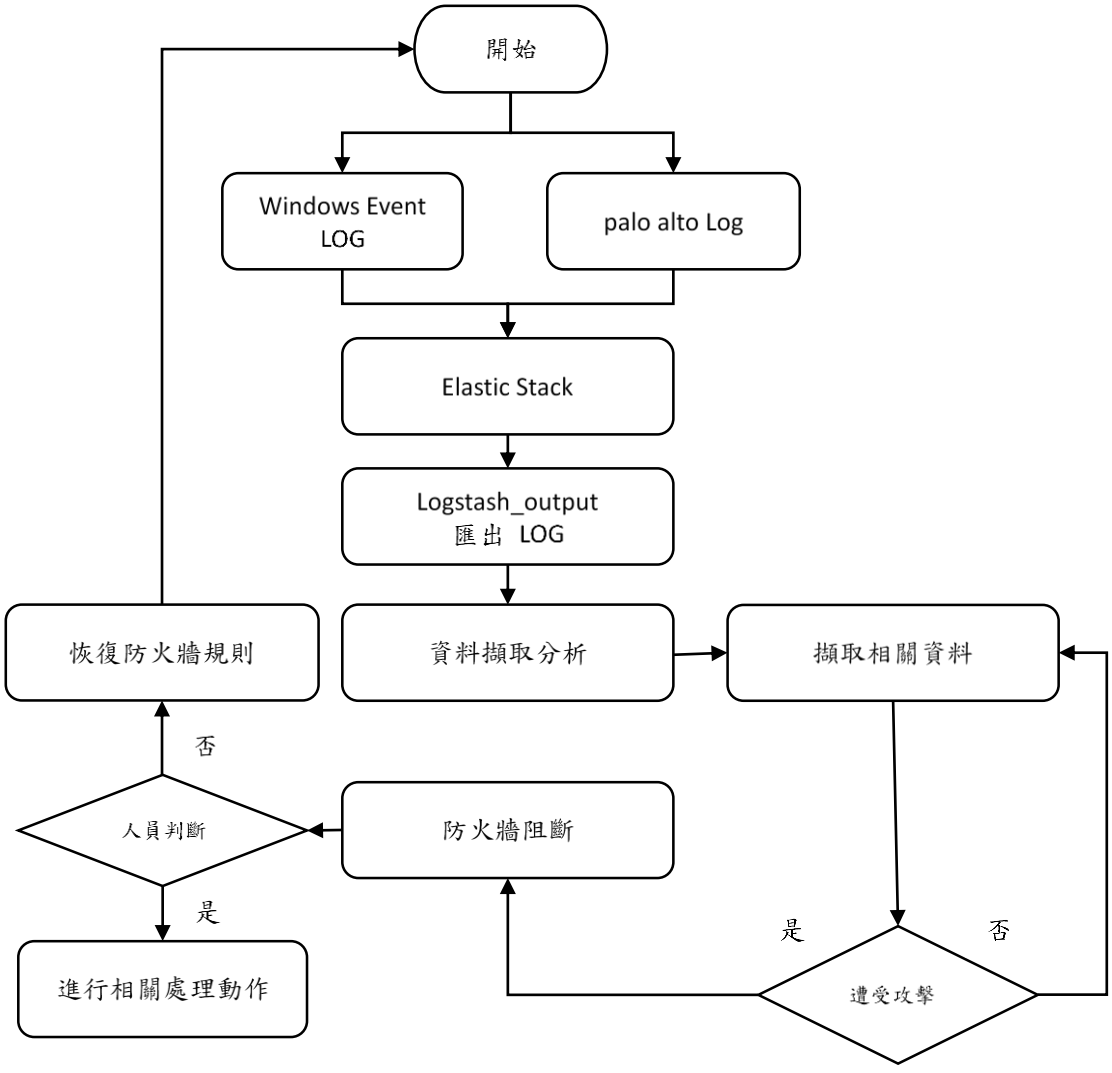
圖二：系統架構圖

3.2 系統流程

首先，針對 Windows Active Directory 系統的 EventLog，我們事先定義了一系列常見入侵行為標的，如暴力破解和非授權存取檔案等，並指定相對應的事件日誌編號，透過 Elastic Stack，將持續監控和分析這些事件日誌，經由分析後透過 Logstash Csv output plugin 持續不間斷的匯出 CSV 檔。

其次，蒐集來自 Palo Alto 防火牆的數據，該防火牆位於企業內部橫向移動[19]流量的關鍵位置，經由機器學習 Machine Learning 大量的數據分析後，篩選出可能存在攻擊跡象或潛在危險的目標，再由 Elastic Stack 持續分析防火牆產生的事件日誌。

一旦經由 python script 分析擷取後偵測到疑似攻擊跡象，我們將啟動自動回應機制，從事件紀錄中提取疑似遭受攻擊的 IP，並立即在疑似遭受攻擊的電腦防火牆中新增相應 TCP、UDP[20]、ICMP[21]的阻擋規則，以遏制潛在的安全風險。如圖三所示：



圖三：系統流程圖

3.3 日誌收集來源

本文研究中所取得的數據來源共有兩種來源，一為 Windows Active Directory 系統的 EventLog，二為 Palo Alto 次世代防火牆的數據，對於網路防禦皆具有重要性，因為它們提供了不同但關鍵的資訊，有助於全面監控和保護網路安全。

3.3.1 Windows Active Directory 系統的 EventLog[22]

Active Directory 日誌記錄使用者登錄、登出、許可權變更等活動，這些資訊對於確定網路中誰正在做什麼以及他們具有什麼許可權至關重要，異常的用戶活動可能表明帳戶被盜用或異常行為。

EventLog 記錄安全事件，如安全群組原則的更改、帳戶鎖定、安全性漏洞嘗試等。這些事件的監控有助於即時識別潛在的安全威脅，並及時採取措施。

通過 EventLog 可以追蹤許可權的分配和更改，有助於確保僅有授權的用戶能夠訪問特定資源，減少內部威脅，在此研究中我們專注於帳戶安全事件。如表一所示：

表一：Windows Event ID 表

Event ID 4625	登錄失敗事件。記錄登錄失敗嘗試,可以發現暴力破解行為。
Event ID 4648	提升許可權嘗試事件。記錄提升許可權的嘗試,可發現 Privilege Escalation 攻擊。
Event ID 4720	建立新賬戶事件。記錄新賬戶的建立,可監控賬戶的異常創建。
Event ID 4722	修改賬戶事件。記錄對賬戶進行修改,可監控對敏感賬戶的更改。
Event ID 4738	變更賬戶密碼事件。記錄賬戶密碼更改,可監控對重要賬戶密碼的可疑更改。

3.3.2 Palo Alto 次世代防火牆

防火牆記錄網路流量的詳細資訊，包括來源、目的地、埠和協議，這有助於識別潛在的攻擊、異常流量或惡意行為，另外 Palo Alto 防火牆能夠識別和阻止威脅，如惡意軟體、零時差攻擊等，監控防火牆的威脅日誌有助於發現和應對新型攻擊，並可以監控和控制網路中使用的應用程式，同時進行內容過濾，有助於控制不同應用程式的使用。

3.4 Elastic Stack 日誌收集設定

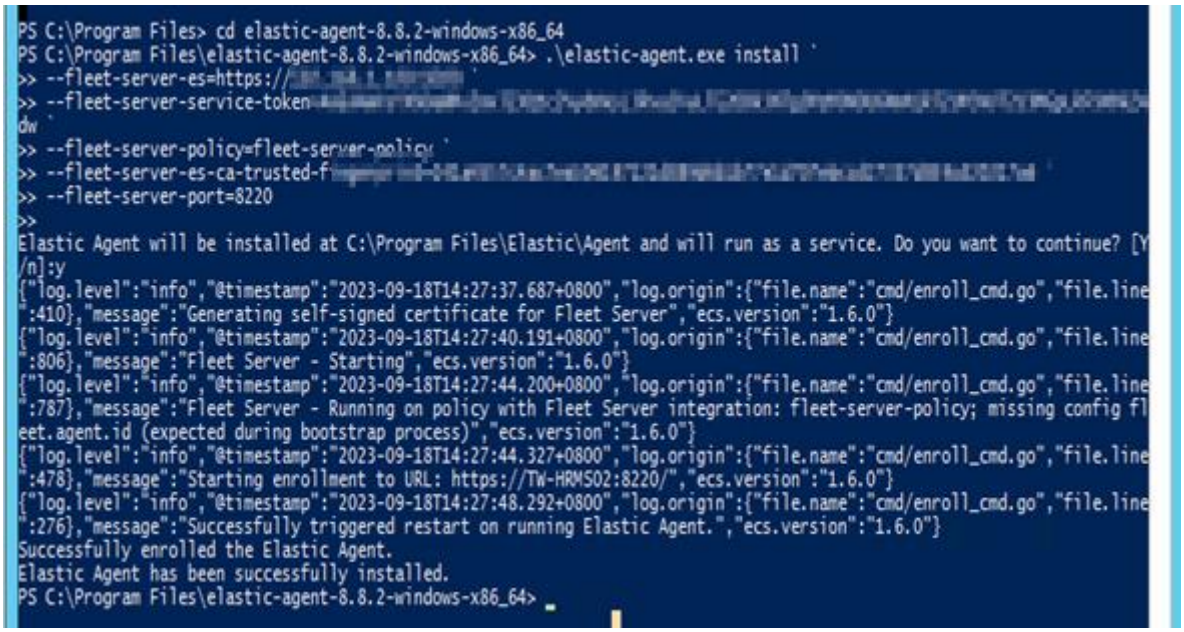
在進行收集日誌前須在 Elastic Stack 主機上安裝設定 Fleet Server 組件，Fleet Server 是 Elastic Stack 中的一個模組，用於集中管理和監控多個 Elastic Agents，而無需單獨設

定每個代理程式，提供了集中式的安全性管理功能，可幫助管理 Elastic Agents 的安全策略和配置,能夠定義和強制執行統一的安全標準和配置,從而提高整個系統的安全性，在管理上如需變更設定，可以輕鬆地對代理程式進行配置更改、應用更新或升級，而無需逐個訪問和操作每個代理程式。

3.4.1 Windows 日誌收集設定

Elastic Agent 安裝：Elastic Agent 是 Elastic Stack 中的一個輕量代理程式，以搜集系統和應用程式的日誌、指標、事件等數據，並將這些數據發送到 Elasticsearch 或 Logstash 進行儲存和分析,透過 elastic-agent.exe install 命令在 Windows 系統上安裝 Elastic Agent，使用 --fleet-server-es 參數指定 Fleet Server 的 Elasticsearch IP 位址，通過 --fleet-server-service-token 參數指定用於驗證和授權的服務 token，使用 --fleet-server-policy 參數指定 Fleet Server 上的策略名稱，--fleet-server-es-ca-trusted-fingerprint 參數則指定 Fleet Server 的 CA 憑證指紋，用於安全性驗證，安裝的 Elastic Agent 會通過上述提的參數與 Fleet Server 建立連接。Fleet Server 管理員可以通過控制面板配置策略和代理程式設定。Elastic Agent 會定期從 Fleet Server 拉取最新的配置和策略，並根據配置進行更新和操作。

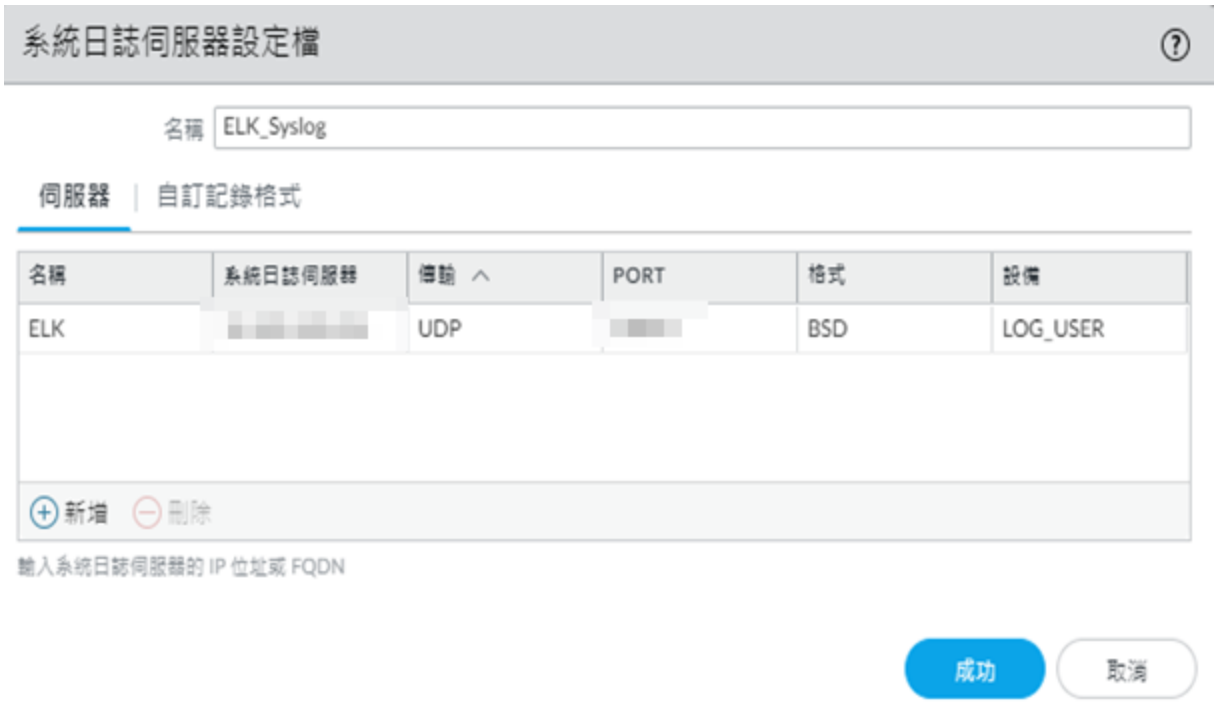
這些設定參數的主要作用是建立一個安全的通道，以確保 Elastic Agent 與 Fleet Server 之間的通信是受保護和經過驗證的，通過使用加密連接和驗證憑證，能夠確保資料在傳輸過程中的安全性和完整性，同時防止未經授權的訪問。如圖四所示：



圖四：Windows Server HOST Add agent

3.4.2 Palo Alto 防火牆日誌收集設定

預先配置 Palo Alto 防火牆，登入 Palo Alto 防火牆管理界面，於 Device Syslog 中建立適當的日誌配置，指示防火牆將日誌數據發送到 Elastic Stack。如圖五所示：



圖五：Paloalto firewall 設置

3.4.3 設置 Palo Alto.conf 檔

設置 conf 檔主要分為三個區塊，分別為 input、fillter、output，這個配置文件是用來接收來自 Palo Alto 防火牆的日誌數據，將其經過篩選、解析後再輸出 Elasticsearch，以下是對配置文件中各個部分的詳細解釋：Input 部分配置 Logstash 接收日誌數據的方式，使用 UDP 協議監聽在本地端口 15044，需與 Palo Alto 防火牆管理界面中 Device Syslog 相同設定。Filter 部分定義了對收到的日誌數據進行處理的規則。Output 部分定義了日誌處理完後的輸出方式。elasticsearch: 設置了將處理好的日誌數據輸出 Elasticsearch。hosts: Elasticsearch 的地址。index: 日誌在 Elasticsearch 中的索引名稱，格式為"paloalto-%{+YYYY.MM.dd}"，功能是接收 Palo Alto 防火牆發送的日誌數據，將其解析後送入 Elasticsearch 進行儲存和索引化。如圖六和七所示：

```
drwxr-s--- 2 root logstash 4096  六  20 13:49 ./
drwxr-s--- 4 root logstash 4096  四  17  2023 ../
-rw-r----- 1 root logstash  852  四  13  2023 [redacted].conf
root@hqelk01:/etc/logstash/conf.d# vim [redacted].conf
```

圖六：Elastic Stack 建立 logstash conf 檔

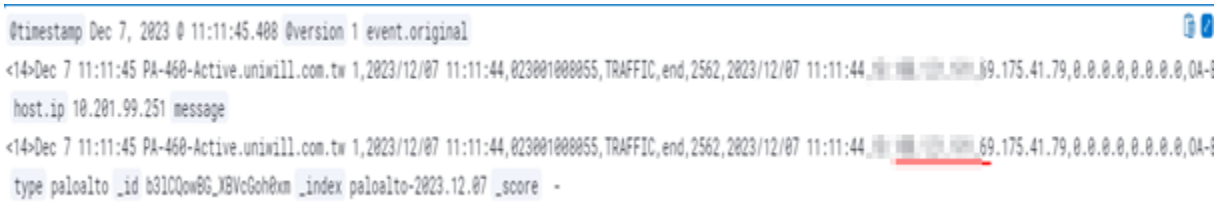
```
input {
  udp {
  port => [redacted]
  type => paloalto
#  ssl => true
#  ssl_certificate_authorities => [redacted]
#  ssl_certificate => [redacted]
#  ssl_key => [redacted]
#  ssl_verify_mode => "force_peer"
  }
}

filter {
  if [type] == "syslog" {
    mutate {
      replace => { "type" => "paloalto" }
    }
    grok {
      match => { "message" => "%{SYSLOG5424PRI}%{GREEDYDATA:message}" }
    }
    date {
      match => [ "timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
    }
  }
}


output {
  elasticsearch {
    hosts => ["[redacted]"]
    index => "paloalto-%{+YYYY.MM.dd}"
    ssl_certificate_verification => false
#    ssl => true
#    cacert => [redacted]
    user => [redacted]
    password => [redacted]
  }
}
```

圖七：Palo Alto logstash conf 設置程式碼

如設定正確，Elastic Stack 則會正確收到由 Palo Alto NetWorkst 傳送之日誌訊息，以利後續截取分析資訊，是否帶有高危險之資訊日誌。如圖八所示：



圖八：Palo Alto Elastic Stack 日誌

3.5 資料擷取分析

資料可以來自不同來源資料，然後對這些資料進行分析和處理的過程，這包括從數據庫、網站、文件、API (應用程序接口) 等不同來源中提取資料，然後使用特定的工具或技術進行分析，以獲得洞察力和策略，這樣的分析可以幫助理解趨勢、發現模式、做出預測，或者制定基於資料的決策，對此研究論文中也將利用此方法進行分析與應用。

3.5.1 Logstash CSV output plugin

Logstash Csv output plugin 可以將從不同來源 (例如日誌文件、數據庫、API 請求等) 收集到的數據轉換成 CSV 格式，有助於統一數據格式，使其易於處理和分析。

CSV 格式的數據非常適合生成報告和統計訊息，使用 CSV 輸出插件，Logstash 可以將數據轉換成 CSV 文件，將數據以 CSV 格式存儲在文件中，CSV 是一種常用的數據交換格式，因此使用 CSV 輸出插件可以將數據轉換成可與其他應用程序共享的格式，在此研究中使用 Logstash Csv output plugin 插件將有用的資料會出轉換為 CSV 檔。

資料匯出 CSV 檔，Logstash 則需相對應的配置，主要分為三個主要部分，input、filter、output：

Input：使用 elasticsearch plugin 作為輸入源，它指定連接到 Elasticsearch 位於 192.168.X.X:port 的特定 index logs-system* 索引，通過 query 欄位指定了一個 Elasticsearch 查詢，該查詢在 Elasticsearch 中選擇了在過去 15 分鐘內發生的所有事件，在最符合主機狀況下，時間可依實際狀況進行調整，指定用戶名和密碼進行驗證，ssl_enabled 設置為 true，這表示啟用了 SSL 連接。如圖九所示：

```
input {
  elasticsearch {
    hosts => [redacted]
    index => "logs-system*"
    query => '{"query": {"bool": {"must": [{"match_all": {}}, {"range":
    user => [redacted]
    password => [redacted]
    ssl_enabled => true
    ssl_verification_mode => "none"
  }
}
```

圖九：input 配置文件程式碼

Filter：date filter 被用來重新格式化事件的 @timestamp 字段，將其轉換為符合 "ISO8601" 格式的日期時間字符串，timezone 設置為 "Asia/Taipei"，將事件的時間戳轉換為台北時區的時間，ruby filter 用於將 @timestamp 字段的時間戳值增加 8 小時 (以秒為單位) ，這是因 CSV 輸出插件在導出數據時沒有自動進行時區轉換所致，為了將時間修正與時區相關的差異，進行了戳值增加 8 小時，以達到匯出資料與系統一致。如圖十所示：

```
filter {
  date {
    match => ["@timestamp", "ISO8601"]
    target => "@timestamp"
    timezone => "Asia/Taipei"
  }
  ruby {
    code => 'event.set("@timestamp", LogStash::Timestamp.new(event.get(
  }
}
```

圖十：Filter 配置文件程式碼

Output：使用 csv output 插件將經過處理的事件輸出到 CSV 文件。fields 配置了 CSV 文件中的字段，這些字段包括了 @timestamp、[event][code]、[host][hostname] 和 [event][created]，path 指定了輸出的 CSV 文件存放路徑，根據日期會生成對應格式的文件名。這段程式用於從 Elasticsearch 中查詢特定時間範圍內的日誌數據，重新格式化和加工這些數據，並將其以 CSV 格式存儲到指定路徑中。如圖十一所示：

```
output {
  csv {
    fields => ["@timestamp","[event][code]","[host][hostname]","[event]
    path => "/mnt/nfs66/output-%{+YYYY-MM-dd}.csv"
  }
}
```

圖十一：Output 配置文件程式碼

配置完成後設置排程自動執行，依據設置路徑及定義檔名如 output-%{+YYYY-MM-dd}.csv 依照年月日，進行檔案編碼。如圖十二所示：

```
root@elk-Virtual-Machine:/usr/share/logstash# ./bin/logstash --path.settings /etc/logstash -f /etc/logstash/conf.d/csv.conf --path.data=/var/lib/logstash1
Using bundled JDK: /usr/share/logstash/jdk
Sending Logstash logs to /var/log/logstash which is now configured via log4j2.properties
[2023-11-15T17:47:16,933][INFO ][logstash.runner          ] Log4j configuration path used is: /etc/logstash/log4j2.properties
[2023-11-15T17:47:16,934][INFO ][logstash.runner          ] Starting Logstash {"logstash.version"=>"8.8.2", "jruby.version"=>"jruby 9.3.10.0 (2.6.8) 2023-02-01 107b2e6697 OpenJDK 64-Bit Server VM 17.0.7+7 on 17.0.7+7 +indy +jit [x86_64-linux
]"}
[2023-11-15T17:47:16,935][INFO ][logstash.runner          ] JVM bootstrap flags: [-Xms1g, -Xmx1g, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djruby.compile.invokedynamic=true, -XX:+HeapDumpOnOutOfMemoryError, -Djava.security.egd=file:
/dev/urandom, -Dlog4j2.isThreadContextMapInheritable=true, -Djruby.regexp.interruptible=true, -Djdk.io.File.enableADS=true, --add-exports=jdk.compiler/com.sun.tools.javac.api=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.file=A
LL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.parser=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.tree=ALL-UNNAMED, --add-exports=jdk.compiler/com.sun.tools.javac.util=ALL-UNNAMED, --add-opens=java.base/java.secur
ity=ALL-UNNAMED, --add-opens=java.base/java.io=ALL-UNNAMED, --add-opens=java.base/java.nio.channels=ALL-UNNAMED, --add-opens=java.base/sun.nio.ch=ALL-UNNAMED, --add-opens=java.management/sun.management=ALL-UNNAMED]
[2023-11-15T17:47:17,011][WARN ][logstash.config.source.multilocal] Ignoring the 'pipelines.yml' file because modules or command line options are specified
[2023-11-15T17:47:17,210][INFO ][logstash.agent           ] Successfully started Logstash API endpoint {:port=>9600, :ssl_enabled=>false}
[2023-11-15T17:47:17,353][INFO ][org.reflections.Reflections] Reflections took 45 ms to scan 1 urls, producing 132 keys and 464 values
[2023-11-15T17:47:17,511][INFO ][logstash.javapipeline    ] Pipeline `main` is configured with `pipeline.ecs_compatibility: v8` setting. All plugins in this pipeline will default to `ecs_compatibility => v8` unless explicitly configured othe
rwise.
[2023-11-15T17:47:17,526][INFO ][logstash.javapipeline    ][main] Starting pipeline {:pipeline_id=>"main", "pipeline.workers"=>8, "pipeline.batch.size"=>125, "pipeline.batch.delay"=>50, "pipeline.max_inflight"=>1000, "pipeline.sources"=>["/e
tc/logstash/conf.d/csv.conf"], :thread=>"#<Thread:0x5ac57c9c@/usr/share/logstash/logstash-core/lib/logstash/java_pipeline.rb:134 run>"}
[2023-11-15T17:47:17,819][INFO ][logstash.javapipeline    ][main] Pipeline Java execution initialization time {"seconds"=>0.29}
[2023-11-15T17:47:17,830][WARN ][logstash.inputs.elasticsearch][main] You have enabled encryption but DISABLED certificate verification, to make sure your data is secure set `ssl_verification_mode => full`
[2023-11-15T17:47:17,947][INFO ][logstash.inputs.elasticsearch][main] ECS compatibility is enabled but `target` option was not specified. This may cause fields to be set at the top-level of the event where they are likely to clash with the E
lastic Common Schema. It is recommended to set the `target` option to avoid potential schema conflicts (if your data is ECS compliant or non-conflicting, feel free to ignore this message)
[2023-11-15T17:47:17,947][INFO ][logstash.javapipeline    ][main] Pipeline started {"pipeline.id"=>"main"}
[2023-11-15T17:47:17,954][INFO ][logstash.agent           ] Pipelines running {:count=>1, :running_pipelines=>[:main], :non_running_pipelines=>[]}
[2023-11-15T17:47:18,287][INFO ][logstash.outputs.csv     ][main][042da1bd9cbd21fb60f601c4744437dae02c1648531282ee0125d4db0da1b6907] Opening file {:path=>"/mnt/nfs66/output-2023-11-17.csv"}
[2023-11-15T17:47:18,292][INFO ][logstash.outputs.csv     ][main][042da1bd9cbd21fb60f601c4744437dae02c1648531282ee0125d4db0da1b6907] Opening file {:path=>"/mnt/nfs66/output-2023-11-15.csv"}
[2023-11-15T17:47:18,295][INFO ][logstash.outputs.csv     ][main][042da1bd9cbd21fb60f601c4744437dae02c1648531282ee0125d4db0da1b6907] Opening file {:path=>"/mnt/nfs66/output-2023-11-16.csv"}
[2023-11-15T17:47:21,529][INFO ][logstash.javapipeline    ][main] Pipeline terminated {"pipeline.id"=>"main"}
[2023-11-15T17:47:21,960][INFO ][logstash.pipelinesregistry] Removed pipeline from registry successfully {:pipeline_id=>:main}
[2023-11-15T17:47:21,962][INFO ][logstash.runner          ] Logstash shut down.
root@elk-Virtual-Machine:/usr/share/logstash#
```

圖十二：資料匯出儲存

3.6 擷取相關資料

執行腳本軟體[23]是用來執行和管理腳本文件的工具，它們有各自的功能和特點，這些工具提供了一個平台來執行和管理腳本文件，幫助用戶自動化任務、進行系統管理和處理各種操作，使用哪個工具取決於所需的功能、熟悉程度以及作業系統等因素。

3.6.1 讀取判斷數據

在本研究中我們選擇使用 Python 編寫的腳本，Python 是一個跨平台的腳本語言，能夠在多種不同的操作系統上運行，這使得在遠端控制不同系統時更加方便，本研究中其主要目的是從指定的 CSV 文件中讀取數據，然後根據特定條件對系統的防火牆進行配置和管理，這個腳本使用了 Pandas 來處理 CSV 文件數據，Pandas 是 Python 中用於數據處理和分析的強大庫之一，它提供了許多功能，可以方便地讀取、處理和操作各種數據格式，並使用了 subprocess[24] 模組來調用命令行工具，subprocess 模組是 Python 標準庫提供的一個用於執行外部命令或程序的工具，允許從 Python 程序中創建新的進程，連接到其輸入輸出流，並獲取進程的返回輸出，這個模組非常有用，特別是需要在 Python 中與系統命令行工具或其他程序進行交互時。例如 psexec[25] 和 netsh[26]，這些工具在 Windows 上用於執行遠程命令和配置防火牆規則，以達到主動防禦的效果。如表二所示：

表二：執行腳本工具比較表

腳本工具軟體	優點	缺點
PowerShell	1.PowerShell 是一個功能強大的腳本語言，具有許多內建的命令和功能，能夠處理許多複雜的遠端控制任務。 2.在 Windows 系統上與其它 Microsoft 產品整合良好。	1.對於初學者來說，PowerShell 的語法可能會比 Batch 文件複雜一些。 2.有些操作需要適當的權限，可能需要提升權限才能執行。
Batch 文件	1.Batch 文件的語法相對簡單，容易上手。 2.對於舊版的 Windows 系統具有很好的相容性。	1.Batch 文件的功能相對受限，尤其在處理複雜或現代化的遠端控制操作時，可能會遇到限制。
Python	1.Python 是一個跨平台的腳本語言，能夠在多種不同的操作系統上運行，這使得在遠端控制不同系統時更加方便。 2.Python 擁有龐大的開源庫和社群支持，可以輕鬆擴展功能。	1.如果目標系統上沒有安裝 Python，就需要事先安裝 Python 環境才能執行。 2.相對於 Batch 文件和 PowerShell，Python 可能會在效能方面稍微慢一些。

3.6.2 命令、權限、控制防火牆工具

遠端管理工具的 command 指的是可以用於在遠端系統上執行命令、配置設置或管

理目標系統的命令列工具，這些 command 可以是命令提示字元 (Command Prompt) 或 PowerShell 中的命令、腳本，或者是特定工具 PsExec、netsh advfirewall、RSAT 所支援的命令。

在此研究中我們將使用 PsExec 和 netsh advfirewall 是兩個的工具，各自擁有不同的功能，它們通常用於不同的用途，在這裡列出它們一起使用的理由：

遠端執行命令：PsExec 可以在遠端系統上運行命令，而 netsh advfirewall 則是用於配置 Windows 防火牆的命令，一起使用時，可以通過 PsExec 在遠端系統上執行 netsh advfirewall 命令，以配置或修改目標系統的防火牆規則。

遠端管理：PsExec 允許用戶以系統帳戶權限遠端登入 Windows 系統並執行命令，這使得管理員可以在不需要直接登入目標系統的情況下，對其進行配置和管理。當結合 netsh advfirewall 使用時，可以通過 PsExec 在遠端系統上配置防火牆規則。

腳本化操作：將 PsExec 和 netsh advfirewall 結合可以通過腳本自動化遠端系統的防火牆管理，用戶可以創建包含遠端命令的腳本，使用 PsExec 遠端執行這些腳本以配置目標系統的防火牆。如表三所示：

表三：遠端管理工具表

Command Tool	優點	缺點
PowerShell	1.提供豐富的 cmdlet 來管理防火牆規則，可以執行更多高級操作。 2.可以更精準地設置和修改防火牆規則。	1.需要 PowerShell 知識和相應的 cmdlet 使用技能。 2.操作複雜度可能較高，需要理解和學習 PowerShell 語法和概念。
PsExec	1.可以透過命令行方式在遠端系統執行任意命令或腳本，包括管理防火牆。 2.適合於經常需要遠端管理的環境。	1.操作較為基本，不提供防火牆特定的命令集。 2.需要使用者有適當的遠端訪問權限。 3.需要額外的指令以修改防火牆設置
netsh advfirewall	1.提供基本的命令集用於防火牆管理。 2.相對於 PowerShell 較易於使用和理解。	1.不提供與 PowerShell 或其他腳本相同的靈活性和進階功能。 2.操作較為基本，無法處理較複雜的設置。
RSAT	1.提供圖形化介面，較易於用戶操作。 2.包含各種用於管 Windows 的工具，而不僅僅是防火牆。	1.使用 GUI 介面可能較為耗時，特別是對於需要重複操作的任務。 2.操作步驟可能較多，不如命令行工具那麼快捷。

肆、資料應用轉換與系統展示

防禦系統框架在執行日誌分析與資料探勘的過程，主要的動作要協助網路主動防禦的形成，要達成主動防禦的架構，我們在此研究系統中進行了程式的編寫，程式主要分為幾個部分，分別為讀取 CSV 檔案、篩選數據定義數據、檢查並操作生成防火牆規則。

4.1 讀取 CSV 檔案

在此預先由 Elastic Stack 匯出 LOG 所定義好的路徑裡取得 CSV 檔，FILE_PATH = "E:\\csv\\elk_data_xxxxcsv"，導入必要的模塊和設定變數，並設置了一些變數來存放文件路徑、IP 規則、允許的 IP、使用者名稱和密碼。如圖十三所示：

```
import pandas as pd
import subprocess

FILE_PATH = [redacted].csv"
IP_RULE = [redacted]
ALLOWED_IP = [redacted]
USERNAME = [redacted]
PASSWORD = [redacted]
```

圖十三：變數定義

讀取 CSV 檔案的函數 ，這個函數通過 pandas 的 read csv 函數來讀取指定路徑下的 CSV 檔案，如果文件不存在，會顯示出 FileNotFoundError，並顯示相應的錯誤訊息。其他可能的錯誤也會被捕獲並顯示相應的錯誤訊息。如圖十四所示：

```
def read_csv(file_path):
    try:
        data = pd.read_csv(file_path, encoding='ISO-8859-1')
        return data
    except FileNotFoundError:
        print(f"檔案 {file_path} 未找到")
        return None
    except Exception as e:
        print(f"讀取檔案時發生錯誤：{str(e)}")
        return None
```

圖十四：讀取 CSV 檔案的函數

4.2 篩選數據定義數據

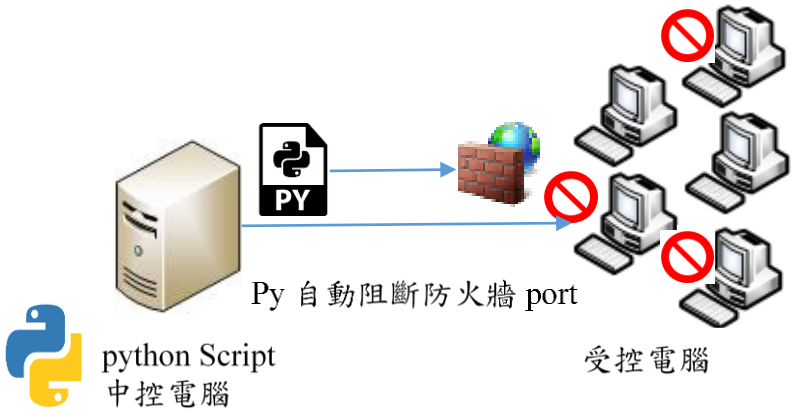
篩選數據的函數，這個函數對給定的數據進行篩選。使用 pandas 的 DataFrame 的 str.startswith() 函數尋找符合指定 IP 規則的資料，同時也要求 'GPT-attack' 列的值為 'yes'。如果指定的列名不存在，會顯示出 KeyError，並顯示相應的錯誤訊息。如圖十五所示：

```python
# 篩選 IP 地址規則和 GPT-attack 為 yes 的行
def filter_data(data, ip_rule):
    try:
        filtered_data = data[data['Source Address (src)'].str.startswith(ip_rule) & (data['GPT-attack'] == 'yes')]
        return filtered_data
    except KeyError:
        print("數據列名稱不匹配")
        return None
```

圖十五：篩選數據程式碼

4.3 自動執行 Python 編碼程式

此系統框架函數使用 PsExec 工具執行多個 netsh 命令，以關閉特定 IP 的 UDP、ICMP 和 TCP 流量，同時添加防火牆規則允許特定允許 IP 通過指定的特定端口連接，通過使用排程管理員定期執行 Python 程式，實現不間斷的檔案搜尋和比對，可以即時阻斷並關閉防火牆的 TCP (1-65535)、UDP (1-65535)、ICMP 流量，以效防止潛在災害的擴大。如圖十六、十七、十八、十九 所示:



圖十六：受控電腦架構

```python
def execute_psexec(filtered_ips, username, password):
    for ip in filtered_ips:
        try:
            # 關閉 UDP 流量
            udp_command = (
                fr"E:\SysinternalsSuite\psexec.exe "
                fr"\\{ip} -i -u {username} -p {password} -h "
                fr"netsh advfirewall firewall add rule name=disable-UDP-inbound dir=in action=block protocol=UDP localport=1-65535"
            )
            subprocess.run(udp_command, shell=True, check=True)
            print(f"已成功關閉 {ip} 的 UDP 流量")

            # 關閉 ICMP 流量
            icmp_command = (
                fr"E:\SysinternalsSuite\psexec.exe "
                fr"\\{ip} -i -u {username} -p {password} -h "
                fr"netsh advfirewall firewall add rule name=disable-ICMP-inbound dir=in action=block protocol=ICMPv4"
            )
            subprocess.run(icmp_command, shell=True, check=True)
            print(f"已成功關閉 {ip} 的 ICMP 流量")

            # 關閉 TCP 流量
            tcp_command = (
                fr"E:\SysinternalsSuite\psexec.exe "
                fr"\\{ip} -i -u {username} -p {password} -h "
                fr"netsh advfirewall firewall add rule name=disable-TCP-inbound dir=in action=block protocol=TCP localport=1-65535"
            )
            subprocess.run(tcp_command, shell=True, check=True)
            print(f"已成功關閉 {ip} 的 TCP 流量")
```

圖十七：Python 編碼

```
E:\csv[redacted] py

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


確定。

netsh exited on [redacted] with error code 0.
已成功添加允許 [redacted] 通過 [redacted] 端口連接的規則

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


確定。

netsh exited on [redacted] with error code 0.
已成功關閉 [redacted] 的UDP流量

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com


確定。

netsh exited on [redacted] with error code 0.
已成功關閉 [redacted] 的ICMP流量

PsExec v2.43 - Execute processes remotely
Copyright (C) 2001-2023 Mark Russinovich
Sysinternals - www.sysinternals.com
```

圖十八：阻斷關閉防火牆 TCP、UDP、ICMP

圖十九：Windows 防火牆規則生成

伍、結論

此研究系統框使用 Elastic Stack 作為日誌收集系統，不僅提升了對數據收集和分析的效率，更強化了對入侵行為的即時感知能力，這種系統的優越性在於能夠快速而主動地應對潛在的安全威脅，結合事件監控和自動化回應的策略，明顯提高了整體系統的安全性，有力地確保企業資訊資產的完整性和機密性。

透過系統框架方法的應用，我們成功建立了一個自動化流程的體系，隨著演算法的不斷進步和優化，系統框架也能夠更準確地分析資料，持續監控事件日誌以識別潛在的入侵跡象，使我們能夠即時辨識潛在的威脅，同時，通過針對安全事件的自動防火牆規則生成，我們能夠強化防禦措施，應對不斷變化的網絡安全威脅，這些應用的策略不僅增強了我們檢測和應對安全事件的迅速性，還降低了因人工干預延遲而帶來的風險。

這些措施在保護敏感數據、維護系統完整性以及減輕潛在的網絡威脅對組織資產的影響方面至關重要，事件監控和自動化回應的方法共同構建了一個強大的安全防禦框架，不僅適用於當前系統結構，還可擴展應用於其他 Syslog 系統介面，這樣的系統架構既能降低企業在資訊安全方面的資本支出，節省商用入侵偵測系統 (IDS) 的開銷，同時全面保護企業資產，進一步增強整體安全基礎架構，鑒於網路環境的不斷變化和威脅的持續提升，我們將持續優化入侵偵測的方法，並仔細檢討事件日誌欄位的解析，以提高防護系統的偵測正確性與完整性，這樣的努力將有助於強化企業抵禦入侵的能力，確保安全防禦系統在快速變化的威脅環境中保持高效運作。

參考文獻

[1] J.-H. Cho et al., "Toward proactive, adaptive defense: A survey on moving target

defense," in IEEE Communications Surveys & Tutorials, vol. 22, no. 1, pp. 709-745, 2020.

[2] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, "Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures," Sensors, vol. 21, no. 14, p. 4759, 2021.

[3] H.-J. Liao et al., "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16-24, 2013.

[4] A.P. Veiga, "Applications of artificial intelligence to network security," arXiv:1803.09992, 2018.

[5] G. Dupont et al., "Similarity-Based Clustering For IoT Device Classification," in 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), pp. 1-7, 2021.

[6] Asia Spam-message Research Center (ASRC), "ASRC," Available at: https://www.asrc-global.com/insights.html?nid=1106.

[7] F. Salahdine and N. Kaabouch, "Social engineering attacks: A survey," Future Internet, vol. 11, no. 4, p. 89, 2019.

[8] Elasticsearch, "Elastic Stack: Elasticsearch, Kibana, Beats & Logstash," Available at: https://www.elastic.co/elastic-stack/.

[9] S. He et al., "Experience report: System log analysis for anomaly detection," in 2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE), 2016, IEEE.

[10] J. Kim et al., "A Model for Illegal File Access Tracking Using Windows Logs and Elastic Stack," Journal of Information Processing Systems, vol. 17, no. 4, 2021.

[11] K. Subramanian and W. Meng, "Threat Hunting Using Elastic Stack: An Evaluation," in 2021 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), pp. 1-6, 2021.

[12] Kali, "Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution," Available at: https://www.kali.org/.

[13] C.-C. Yeh, "Endpoint log analysis and anomaly detection," NTU Theses and Dissertations Repository, 2017.

[14] OSSEC, "OSSEC - World's Most Widely Used Host Intrusion Detection System - HIDS," Available at: http://ossec.github.io/index.html.

[15] V. Stafford, "Zero trust architecture," NIST Special Publication, 800: p. 207, 2020.

[16] Elasticsearch, "Csv output plugin | Logstash Reference [8.11] | Elastic," Available at: https://www.elastic.co/guide/en/logstash/current/plugins-outputs-csv.html.

[17] M. Learn, "Active Directory Overview of Domain Services," Available at: https://learn.microsoft.com/zh-tw/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview.

[18] Palo Alto Networks, "Leader in Cybersecurity Protection & Software for the Modern Enterprises - Palo Alto Networks," Available at: https://www.paloaltonetworks.tw/resources/datasheets/firewall-feature-overview-datasheet.

[19] M. Meijerink, "Anomaly-based detection of lateral movement in a microsoft windows environment," University of Twente, 2019.

[20] M. Polese et al., "A survey on recent advances in transport layer protocols," in IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3584-3608, 2019.

[21] DARPA, "Internet Control Message Protocol - Wikipedia," Available at: https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol.

[22] M. Learn, "Appendix L - Events to monitor | Microsoft LearnWindows EventLog," Available at: https://learn.microsoft.com/zh-tw/windows-server/identity/ad-ds/plan/appendix-l--events-to-monitor.

[23] P. Stöckle, B. Grobauer, and A. Pretschner, "Automated implementation of windows-related security-configuration guides," in Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering, 2020.

[24] Python Software Foundation, "Subprocess management," Available at: https://docs.python.org/zh-tw/3/library/subprocess.html.

[25] M. O'Leary and M. O'Leary, "Firewalls," in Cyber Operations: Building, Defending, and Attacking Modern Computer Networks, pp. 521-563, 2015.

[26] V.O. Kayhan, M. Agrawal, and S. Shivendu, "Cyber threat detection: Unsupervised hunting of anomalous commands (UHAC)," Decision Support Systems, vol. 168, p. 113928, 2023.