基於 API 解析及運用深度學習的工業自動化及控制系統惡意軟體偵測機制

Malware Detection Mechanisms for Industrial Automation and Control Systems Based on API Analysis and Deep Learning

林詠章、馬毅凱
Iuon-Chang Lin, Yi-Kai Ma

隨著智慧製造應用的快速發展，智慧化的同時也帶來了一些潛在的資安風險，像是工業自動化及控制設備受到像勒索病毒等惡意軟體的威脅。因此，工控場域的端點防禦機制成為確保工控場域能否可靠運作的重要關鍵。本文將針對智慧製造場域之端點防護提出惡意軟體的偵測機制，透過沙盒環境萃取軟體之系統 API 呼叫序列並找出序列前後的潛在關係，進而利用深度學習來建立惡意軟體偵測模型，此機制可避免惡意程式透過變種來躲過相關偵測，有效降低智慧場域內端點設備的資安威脅。

With the rapid development of smart manufacturing applications, intelligence also brings some potential security risks, such as industrial automation and control equipment being threatened by malicious software such as ransomware. Therefore, the endpoint defense mechanism of the industrial control field has become an important key to ensure the reliable operation of the industrial control field. This paper will propose a malware detection mechanism for endpoint protection in the smart manufacturing field. We extract the system API call sequence of the software through the sandbox environment and find out the potential relationship before and after the sequence. Then use deep learning to build a malware detection model. This mechanism can prevent malicious programs from evading detection through variants, and effectively reduce the threat to the security of endpoint devices in the smart manufacturing field.

一、前言

資訊技術的發展導致了惡意軟體的變種速度飛快，讓現今無數的工控場域暴露在惡意軟體攻擊的風險之中。以前在工控場域的資安防護觀念中，普遍認為只要不將設備連到外網就

可以有效的避免惡意軟體攻擊，但越來越多工控資安事件也證明了即使場域設備沒有開放對外網連線，也有可能遭受供應鏈、機台更新、設備維護、社交工程等方式侵入內部網路，造成資安衝擊。Salahdine 等人[1] 便統計了超過 10 種關於社交工程攻擊的項目與過程，並且充分的說明各個攻擊類型的原理，其中特別強調攻擊手法變化多樣與快速，要能夠完全阻斷惡意攻擊是很困難的。而 Brewer[2] 針對勒索軟體攻擊的準備階段、檢驗階段、遏止階段、剷除階段及復原階段提出了以下防禦重點：

1. 準備階段：在開發過程中應該要盡量減少系統漏洞或透過輔助系統補強。
2. 檢驗階段：應針對場域端點或是網路流量進行異常檢測。
3. 遏止階段：如果攻擊已經發生，應該盡量縮小傷害範圍，避免持續擴散。
4. 剷除階段：將已經被攻擊的設備重置，剷除所有惡意軟體。
5. 復原階段：將備份資料重新復原到新系統中。

要有效降低勒索病毒所帶來損失，以災難預防的概念來看，若能在前兩個階段能有效偵測出惡意程式並有所部屬，就能避免惡意程式的攻擊，本文便是期望能在準備階段及檢驗階段的資安防禦機制做研究，提出一個場域端點惡意軟體的偵測方法。

在現有針對惡意軟體偵測的方式中，主要可以依照軟體分析的方式分為靜態分析與動態分析兩種類別，前者是惡意軟體尚未執行時就透過一些手段去萃取與剖析軟體，主要是查看存在於磁碟上的特定檔案內容，而不是觸發時出現的特定檔案或路徑，透過剖析數據原碼、逆向工程來分析軟體的行為。而動態分析則是假設惡意軟體執行時才觀測軟體本身的行為，根據軟體執行時的操作來對檔案進行分級，而不單單只依靠特徵碼來識別威脅軟體。一般來說動態分析更能具備惡意軟體的可視性，如 Matilda Rhode 等人[3] 便透過觀測系統資源使用量來建立一個透過 RNN 在惡意軟體執行前期的預測模型，並且證明其有不錯的偵測效果。Tina Rezaei 等人[4] 則透過 PE Header 檔案格式序列來判斷在 windows 作業系統中的軟體是否為惡意軟體，並且能夠在感染前期階段達到有效的預測，降低惡意軟體攻擊帶來的損失。動態分析的手法多樣，而其中最具有代表性的就是使用系統的 API (application programming interface) 呼叫序列作為動態分析的關鍵依據，所以本文章將基於考慮時間序列來針對惡意軟體系統 API 呼叫之動態行為建立偵測模型。

二、基於 API 呼叫序列之惡意軟體偵測機制

1. 動態分析之惡意軟體偵測架構

我們提出一個設備端點的惡意軟體偵測架構，透過沙盒環境來萃取惡意軟體的 API 序列做為資料來源，接著利用深度學習演算法來建立白名單機制與考慮 API 序列前後關係的惡意軟體偵測模型，整體流程如圖 1 所示。

當一個陌生軟體要被載入端點前，會先將該軟體執行在獨立的沙盒環境中，並且在該環境中分析該軟體的 API 序列，第一步先對比黑名單，如果該 API 序列存在於黑名單中，則直接將其軟體封鎖。如不存在於黑名單內，則再近一步比對白名單，存在於白名單中，則代表該軟體為善意軟體，是允許在目標端點部屬的，若也不存在於白名單，代表該軟體為陌生軟體，將透過基於 API 前後序列的惡意軟體偵測模型做判斷，若經模型判斷為惡意軟體則將該序列加入黑名單，若模型判斷為善意，則加入白名單，以供未來相同軟體之黑白名單驗證機制。

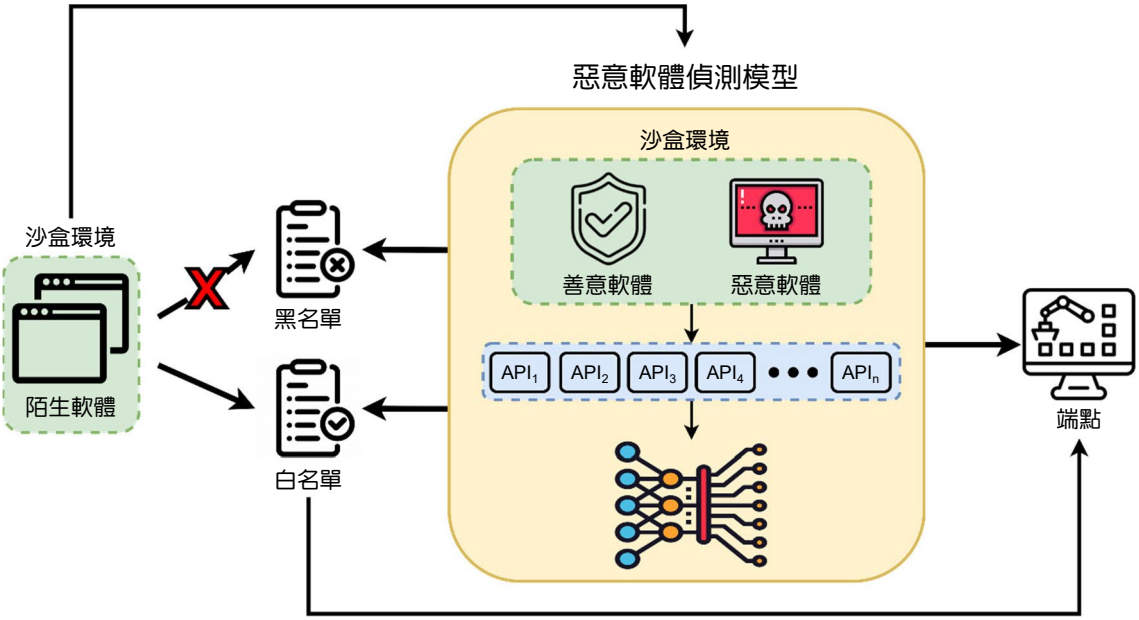

圖 1. 惡意軟體行為異常 API 呼叫偵測架構圖。

本文的偵測模型為二元分類偵測模型，該二元即為是「惡意軟體」或「善意軟體」，最後將嘗試使用不同長度的 API 序列作為訓練數據，來比較軟體執行時各個時間的準確度，以驗證模型的早期偵測能力。

2. 惡意軟體 API 序列監測與數據萃取

系統 API 是作業系統中的核心應用程式介面，API 會將有關於系統中複雜的操作封裝在動態函式庫中 (如 kernel32.dll、user32.dll 等)，主要是提供存取硬體和系統資源的功能。除了用於作業系統本身之外，同時也提供給程式開發者作為複雜操作的應用，降低開發難度，幾乎所有的軟體都會與作業系統 API 進行密切的互動，而 API 是依照其實際功能給予特定規則的命名，如圖 2 所示：

對於惡意軟體的系統 API 呼叫序列的數據萃取通常使用隔離環境來執行惡意軟體並監測，如圖 3 所示。我們使用 Cuckoo 來萃取惡意軟體的 API 序列，Cuckoo 是一個開源的沙盒環境，它提供了一個安全的虛擬隔離空間來執行軟體，Sainadh 等人[5] 在這個環境中可以監測軟體的行為，並且可以產生完整的軟體分析報表，我們透過此環境收集善意與惡意軟體的 API 序列。

3. 善意軟體 API 序列萃取

與惡意軟體不同，針對正常善意軟體的 API 數據萃取不需使用隔離環境來單獨執行，所以使用 Process Monitor 來監視系統 API 呼叫的操作過程，Process Monitor 是一個 Windows 監測工具，可以對系統中的任何檔案、註冊表操作進行監控，使用此工具可以完整的萃取善意軟體的 API 序列數據做為後續模型訓練與白名單建置的依據。

4. API 序列間的潛在關係意義

傳統的的機器學習模式中，訓練樣本的特徵供與資料前處理的好壞往往可以影響最後的

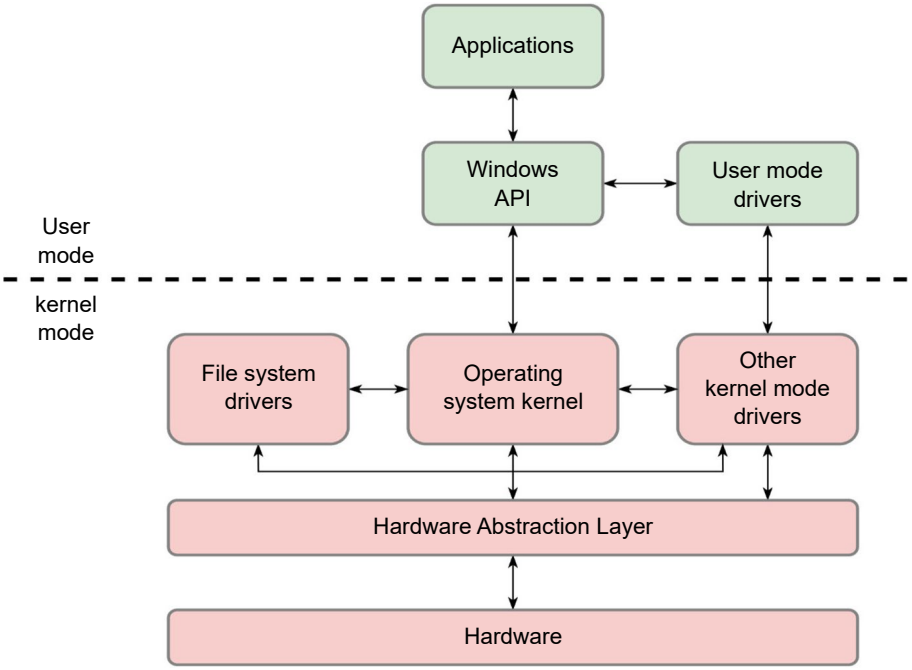

圖 2. Windows 作業系統 API 呼叫機制。

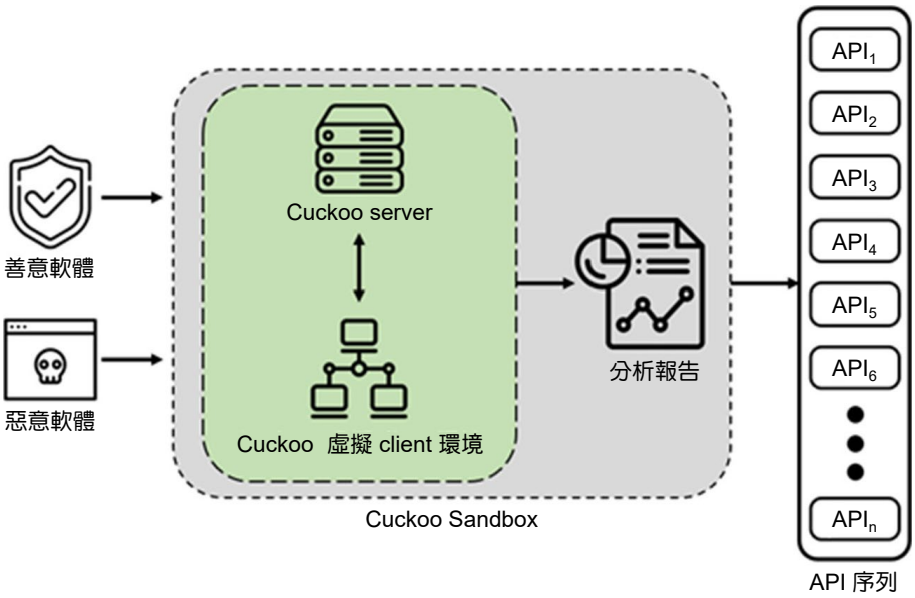

圖 3. 透過 Cuckoo Sandbox 收集軟體 API 序列。

訓練結果，其中 Label Encoding 與 One Hot Encoding 是常見的前處理手法，目的是為了將訓練數據中的類別或是文字資料轉換成二元向量，讓模型能夠更好的被訓練。但這種方式存在某些缺點，像是大幅度的增加數據特徵的維度，造成過多的無益數據，進而影響梯度下降的效果，或是增加額外的訓練時間，且使用這種方式轉換數據無法充分表達數據樣本間的關聯性。

而在深度學習中，Embedding 的技術被廣泛使用，Embedding 是一種將離散變數轉換為連續向量表示的一種方式，在深度學習的網路中，Embedding 相對於前述兩個方法更具優勢。因為它不僅可以減少離散變數的空間維度數量，同時還可以表達樣本之間的關聯，以本研究的系統 API 呼叫序列樣本來說，API 總共有 307 個，也就是說如果以 One Hot Encoding 做為樣本向量轉換，則表示每一個系統 API 呼叫都必須以變數總數 N-1 個 0 和單一個 1 所組成的 vector 來表示。但這樣的作法有兩個明顯的缺點，如對於具有非常多類型的變數，轉換後的維度過大且過於稀疏，另一缺點是轉換映射之間完全獨立，並不能夠表示不同類別之間的關係。而若以 Embedding 來做轉換，總數為 307 個的系統 API 呼叫，對於每一個 API，可以僅用一個包含 32 個數字的向量即可表示。除此之外，因為 Embedding 是可以學習的，所以在訓練過程中，相近的 API 的表示在 Embedding 空間內是彼此會越來越接近的，也就是映射後兩組向量的歐幾里得距離很小，這種作法除了可以表達樣本空間中數據間的潛在關係之外，也能夠將樣本以低維度的向量方式表示，達到降維效果。

此外，API 序列是有潛在的前後關係的，例如通常勒索軟體會先透過加密 API 加密檔案後再將其密鑰透過網路傳送出去，這一過程中可以觀測到這是具有前後關聯的，也就是前者會影響後者，所以基於這個現象，我們利用時間序列來建立偵測模型。

5. 惡意軟體白名單機制

在惡意軟體檢測方面，早期的多數方法是使用黑名單機制作為驗證的依據，例如，現有的防毒軟體主要是透過建立病毒碼資料庫，當一個陌生軟體進入端點時，會比對病毒資料庫中是否有該軟體的病毒碼，若存在才會將此軟體進行處理[6]。這種防禦手段存在致命的缺點，因病毒變種速度極快，病毒資料庫無法即時更新，且舊有的病毒也可以透過使用混淆技術來改變病毒碼，輕易攻破黑名單的防禦。此外，如果惡意軟體不夠流行，防毒軟體公司也很容易忽略其嚴重性，沒有加入病毒資料庫而造成破口。

近年來業界對於工控場域的資安議題有更高的關注，資安已經成為國安不可或缺的一部份，隨著人工智慧的發展與 Zero Trust 安全模型概念興起，強調「從不信任，總是驗證」已經是端點安全防護的重要趨勢。端點異常檢測中經常改成使用白名單概念來過濾惡意軟體，白名單機制可以完美解決黑名單核心的缺點，與黑名單相反，白名單只允許名單上記錄的軟體為善意軟體。Ginter 等人[7] 使用市面上四家防毒軟體公司的白名單機制進行測試，結果顯示可以有效的提高惡意軟體的攔截能力。

現有的軟體白名單機制大多使用檔案名稱或是來源網址作為驗證軟體是否安全的依據，這種作法過於簡易，容易偽造，安全性相對較低。所以我們提出使用 API 序列作為白名單驗證的依據，將善意軟體的 API 序列雜湊後加入白名單，如圖 4，使用這種方式比單純使用檔名白名單或是來源白名單更加安全，因為驗證的是更接近軟體本身的行為。再者，因是加入整段 API 序列作為驗證，所以相比單純只考慮有無使用該 API 更具安全性，因其順序性也變成驗證的一環。

三、實驗結果

根據上述惡意軟體行為異常 API 呼叫偵測架構進行模型訓練，透過比較傳統機器學習與深度學習神經網路進行多種監督式演算法訓練實驗，針對傳統機器學習分類演算法，本

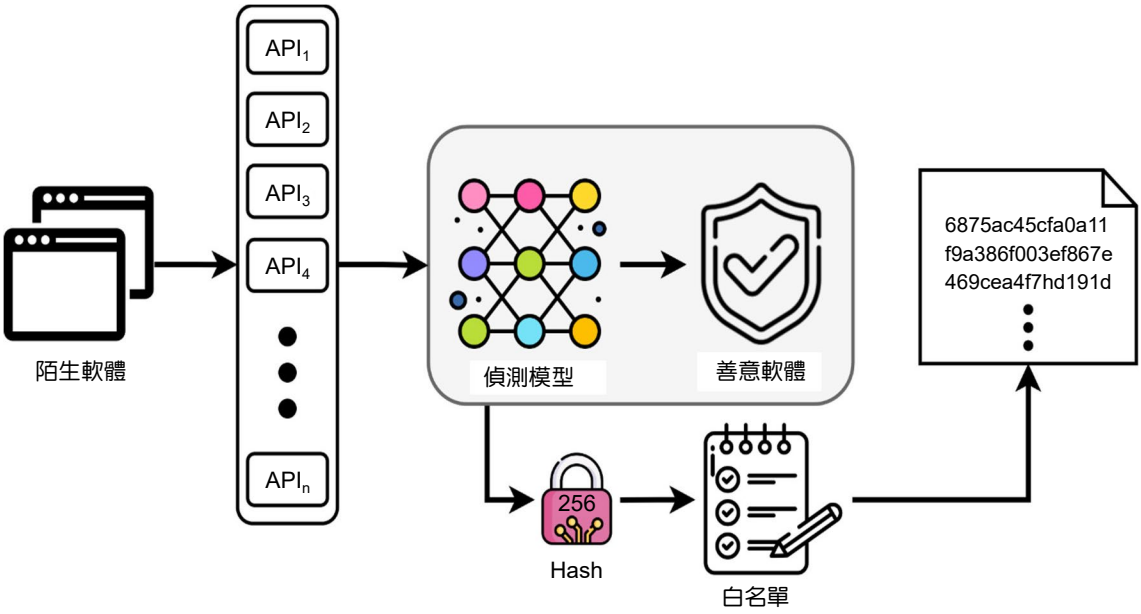

圖 4. 基於 API 序列之白名單機制。

研究使用支持向量機 (support vector machine, SVM)、單純貝氏分類器 (naive Bayes classifier, NB) 監督式機器學習方法進行預測模型的訓練，並將透過混淆矩陣作為評估與比較的標準。

其中，Accuracy、Precision、Recall 及 F1-Score 是機器學習中常用的分類模型性能指標，它們可以通過混淆矩陣計算得到。定義如下：

1. 準確率 (Accuracy)：指分類器正確分類的樣本數佔總樣本數的比例。準確率高表示模型整體分類效果好。
2. 精確率 (Precision)：指分類器預測為正類的樣本中，真正為正類的樣本數佔預測為正類的樣本數的比例。精確率高表示模型對於正類樣本的識別能力較強。
3. 召回率 (Recall)：指分類器預測為正類的樣本中，真正為正類的樣本數佔實際為正類的樣本數的比例。召回率高表示模型對於所有真正為正類的樣本都能識別出來。
4. F1-Score：是精確率和召回率的調和平均數。F1-Score 綜合考慮了精確率和召回率的性能，適合評估分類器的綜合性能。

而為了針對 API 序列的時間關係，我們使用長短期記憶 (long short-term memory, LSTM) 作為深度學習神經網路偵測模型的訓練，LSTM 是一種時間循環神經網路 (recurrent neural network, RNN)，其有效的解決預測時間序列中長期記憶消失的問題，LSTM 可以充分考慮 API 序列間的潛在關係，表 1 呈現了惡意軟體偵測模型的訓練結果。

表 1. 傳統機器學習分類方法與遞歸神經網路針對軟體異常之偵測結果。

	Accuracy	Precision	Recall	F1-score
SVM	0.90	0.92	0.90	0.90
NB	0.86	0.89	0.88	0.88
LSTM	0.94	0.93	0.93	0.93

根據實驗結果，傳統的機器學習分類方法雖有一定的準確度，但對於時間序列的樣本偵測效果還是有限，準確度介於 86%－90% 之間。而對於二元分類問題較為重視的召回率指標則最高是 90%。相對於傳統的分類方法而言，基於深度學習的神經網路則取得更亮眼的分數，準確率高達 94%，而召回率也達到了 93%，這是因為 API 序列具有時間順序性，且有每個 API 之間有潛在關係，遞歸神經網路更能夠找出訓練樣本中的時間特性，獲得較好的偵測效果。

四、結論

隨著科技的快速發展、工業 4.0 的到來，工業控制場域內的聯網設備數量急遽攀升，存在潛在的資安風險。包含勒索軟體等各種惡意軟體變種速度極快且數量日益攀升，對於工控場域威脅極大，若遭受病毒感染，產能嚴重損害，將造成嚴重虧損。

資訊安全問題已經是各個領域不可忽視的一部份，根據不同的資安威脅衍生出了各式各樣的防禦手段，而在智慧製造場域中，端點設備的防禦是置關重要的，所以本研究聚焦在惡意軟體的主動偵測，提出了一個惡意軟體行為 API 呼叫偵測架構。透過貫徹零信任概念與神經網路偵測模型達到高效的偵測，使用系統 API 序列建置白名單機制與考慮前後關係的惡意軟體偵測模型，並且取得優於傳統機器學習模型的結果。基於實驗結果，時間序列的神經網路可以取得 94% 的準確率與 93% 的召回率，達到智慧場域端點設備惡意軟體偵測之效果。

參考文獻

1. Fatima Salahdine and Naima Kaabouch, *Future Internet*, **11** (4), 89 (2019).
2. Ross Brewer, *Network Security*, **2016** (9), 5 (2016).
3. Matilda Rhode, Pete Burnap, Kevin Jones, *Computers & Security*, **77**, 578 (2018).
4. Tina Rezaei, Farnoush Manavi, Ali Hamzeh, *Journal of Information Security and Applications*, **60** (2), 102876 (2021).
5. Sainadh Jamalpur, Yamini Sai Navya, Perla Raja, Gampala Tagore, G. Rao, "Dynamic Malware Analysis Using Cuckoo Sandbox," *Proceedings of 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, September 27 (2018).
6. Daniel J. Sanok, "An Analysis of How Antivirus Methodologies Are Utilized in Protecting Computers from Malicious Code", *Proceedings of the 2nd Annual Conference on Information Security Curriculum Development*, September 23 (2005).
7. Andrew Ginter, Chuck Rohs, "An Analysis of Whitelisting Security Solutions and Their Applicability in Control Systems", *Proceedings of the S4: SCADA Security Scientific Symposium*,7-1 (2010).

作者簡介

馬毅凱先生現為國中興大學資訊管理所碩士生。

Yi-Kai Ma is currently a M.S. student in the Department of Management Information System at National Chung Hsing University.

林詠章教授為國立中正大學資訊工程研究所博士，現為國立中興大學資訊管理系教授兼系主任。

Iuon-Chang Lin received his Ph.D. in Computer Science and Information Engineering from National Chung Cheng University. He is currently a Professor and Head in the Department of Management Information System at National Chung Hsing University.