基於強化學習的工業控制系統無檔案攻擊防禦機制

蔡士煒　郭振忠

南臺科技大學資訊工程系

{mb3g0101, spotkuo}@stust.edu.tw

論文摘要

工業控制系統（Industrial Control Systems, ICS）隨著 IT/OT 整合趨勢，正面臨越來越嚴峻的資安威脅。近年來，無檔案攻擊（Fileless Attack）因爲執行過程完全在記憶體中進行，不會在磁碟上留下實體檔案，能有效規避傳統特徵碼式防毒軟體，成爲針對 ICS 環境的新型高危威脅手法。

本研究設計並實作一套基於強化學習（Reinforcement Learning, RL）的工控網路資安防禦系統，用來偵測透過 HID 裝置注入、PowerShell 無檔案執行，進而以 Modbus TCP 操控 PLC 設備的多段式攻擊鏈。系統採用 Windows Event Log 作爲資料來源，透過 Logstash 即時收集命令列執行記錄，並以 TF-IDF 詞彙、字元 n-gram 及 42 項手工特徵共 1042 維向量表示每筆指令。防禦模型採用近端策略優化演算法（Proximal Policy Optimization, PPO），以二元分類方式判定指令是否爲惡意行爲，並透過四次系統性 Reward Shaping 迭代，在抑制誤報的同時有效提升對進階威脅的偵測能力。

實驗以 2,451 筆訓練樣本與 614 筆測試樣本進行驗證，系統最終達到 97.39% 正確率、98.29% 精確率、95.04% 召回率及 96.64% F1 分數，誤報率僅 1.08%。結果驗證了 PPO 在命令列惡意行爲偵測任務上的可行性，並展示強化學習應用在 ICS 網路防禦場景的潛力。

關鍵詞：工業控制系統、強化學習、無檔案攻擊、PowerShell 偵測、入侵偵測

前言

隨著工業 4.0（Industry 4.0）時代的到來，資訊技術（IT）與操作技術（OT）的深度融合已成爲現代工業發展的核心趨勢。工業控制系統（Industrial Control Systems, ICS）廣泛部署在電力、水資源、石油化工與製造業等關鍵基礎設施，負責監控與調度實體製程設備，包含可程式化邏輯控制器（PLC）、監控與資料擷取系統（SCADA）及人機介面（HMI）等核心組件。

然而，IT/OT 整合趨勢在提升生產效率的同時，也大幅擴展了 ICS 的攻擊面。TXOne Networks 發布的 2024 年度 OT/ICS 資安報告顯示，94% 的受訪組織在過去十二個月內遭遇 OT 資安事件，其中 68% 的攻擊來自 IT 環境的直接滲透，30% 爲 IT 環境的連帶影響，僅 2% 屬於純 OT 導向攻擊 [1]。傳統 ICS 環境假設設備運行在封閉的氣隙網路（Air-gapped Network）中，在設計上普遍缺乏通訊加密、身份驗證與存取控制等安全機制，讓這些設備在面對現代網路威脅時特別脆弱 [2]。

在眾多 ICS 攻擊手法中，**無檔案攻擊**（Fileless Attack）近年來受到資安界的高度關注。這類攻擊利用 Windows 系統內建工具（如 PowerShell、WMI、MSHTA）及合法系統二進位檔案（Living off the Land Binaries, LOLBins）在記憶體中執行惡意邏輯，全程不會在磁碟上留下可供特徵比對的執行檔，因此能有效規避傳統防毒軟體和端點偵測回應（EDR）工具 [3]。搭配 HID（Human Interface Device）裝置的硬體鍵盤模擬注入，攻擊者可以在短時間內完成從實體入侵、無檔案執行到 PLC 暫存器竄改的完整攻擊鏈，對工廠現場的物理安全構成直接威脅。

現有 ICS 資安防禦方案大多聚焦在網路流量分析（如 Modbus 封包過濾）或傳統機器學習特徵比對，對於利用合法系統工具執行、而且不會產生網路異常流量的無檔案攻擊，缺乏有效的應對機制。強化學習（Reinforcement Learning, RL）透過代理人和環境互動來動態調整策略，能在威脅模式持續演變的場景中保持偵測靈活性，適合應用在這類挑戰上。

本研究的主要貢獻包含以下三點：

- **攻擊鏈仿真**：設計並實作涵蓋 HID 裝置注入、PowerShell 無檔案執行和 Modbus TCP 操控 PLC 的完整三段式 ICS 攻擊鏈仿真環境，提供真實威脅情境下的防禦驗證基礎。
- PPO **偵測模型**：提出以 1042 維特徵向量結合近端策略優化（PPO）演算法的命令列惡意行爲偵測框架，並透過四次系統性 Reward Shaping 迭代，將 F1 分數從 80.92% 提升到 96.64%，誤報率降到 1.08%。
- **系統整合**：實現 Windows Event Log → Logstash → Flask API → PPO 模型的完整即時偵測管道，驗證系統在真實 ICS 場域中的可部署性。

文獻探討

工業控制系統安全威脅

ICS 環境中最廣泛使用的 Modbus TCP 通訊協定因爲設計年代久遠（1979 年制定），不具備身分驗證、加密傳輸和封包完整性保護機制，容易遭受中間人攻擊（MITM）、暫存器竄改和阻斷服務（DoS）等威脅 [4]。NIST SP 800-82 將 ICS 的普渡（Purdue）架構分成五個層次，指出 IT/OT 整合後各層邊界防護失效是主要風險 [2]。Raval 等人的系統性調查研究整理了 2021 到 2023 年間對關鍵基礎設施的主要攻擊類型，包含供應鏈攻擊、韌體漏洞利用和社交工程，並指出缺乏有效防禦機制的現場設備是最常見的入侵切入點 [5]。

Aslam 等人針對 ICS 通訊網路的架構漏洞進行深度分析，指出在普渡模型各層之間缺乏零信

任（Zero Trust）機制的環境下，攻擊者一旦突破企業網路邊界，就可以在 OT 環境中橫向移動 [6]。Yang 等人則針對控制頻率不匹配的漏洞進行研究，說明攻擊者可以利用 PLC 回應延遲來進行隱匿式暫存器竄改而不觸發告警 [7]。

無檔案攻擊技術

無檔案攻擊（Fileless Attack）最早可以追溯到 2001 年的 Code Red 蠕蟲，近年來因爲防毒軟體的進步而被攻擊者廣泛採用 [3]。MITRE ATT&CK 框架將這類技術歸類在多個戰術底下，包含 T1059（命令和腳本直譯器）、T1218（系統二進位代理執行）和 T1055（程序注入），並記錄超過二十種常見 LOLBins 濫用手法 [8]。

PowerShell 因爲具備強大的系統管理能力和高度的靈活性，成爲無檔案攻擊的主要載體之一。攻擊者常利用下列技術規避偵測：（1）用 `-ExecutionPolicy Bypass` 略過執行原則；（2）用 `-WindowStyle Hidden` 隱藏執行視窗；（3）透過 `Invoke-Expression`（IEX）在記憶體中載入並執行遠端腳本；（4）利用編碼混淆指令內容。搭配 HID 裝置（如 USB Rubber Ducky、Hak5 O.MG Cable）的硬體鍵盤模擬，攻擊者可以在幾秒內完成整個入侵流程，不需要任何使用者互動 [8]。

現有的無檔案攻擊偵測方法主要分成兩類：（1）基於規則的方法，透過比對已知惡意指令模式來偵測，精確率高但難以應對 Zero Day 變種；（2）基於機器學習的方法，利用統計特徵和行爲模型來分類，能處理沒見過的攻擊變體，但需要大量已標記的訓練資料。

強化學習在入侵偵測的應用

強化學習（RL）是一種讓代理人（Agent）透過和環境持續互動、最大化累積獎勵的學習方式，能夠在動態且不確定的環境中調整決策策略，近年來被應用在網路安全的多個場景 [9]。

近端策略優化（Proximal Policy Optimization, PPO）由 Schulman 等人提出，透過引入剪切（Clipping）機制來限制策略更新幅度，在保持訓練穩定性的同時提升樣本效率，是目前最廣泛採用的深度強化學習演算法之一 [10]。Stable-Baselines3 提供 PPO 的工業級實作，具有良好的可重現性和工程整合性 [11]。

和監督式學習方法相比，RL 代理人有以下優勢：（1）能透過 Reward Shaping 靈活調整誤報率和漏報率的平衡，不需要重新標記訓練資料；（2）具有一定程度的線上自適應能力，可以在持續和環境互動中更新策略；（3）能處理資料不平衡問題，透過差異化獎懲機制來強調稀有類別的重要性。這些特性讓 RL 特別適合應用在無檔案攻擊偵測這類威脅模式持續演變、而且標記成本高的資安場景。

研究方法

系統架構總覽

本研究提出的防禦系統採用三層架構設計，如圖 1 所示，由被入侵端、工控網路和防禦分析端組成。攻擊者透過 HID 裝置入侵企業電腦後，用無檔案方式執行 PowerShell 指令並向 PLC 發送 Modbus TCP 封包；防禦端則持續監控 Windows Event Log，經過 Logstash 管道傳送到分析主機，由 PPO 模型即時判定並記錄惡意行爲。當 PPO 模型判定爲惡意指令時，系統會同步通知防火牆發出封鎖指令，切斷攻擊連線以阻止後續的 PLC 操控。

整體資料流如下：Windows Event Log 由 Logstash（port 5044）用 HTTP Output 外掛即時轉發到 Flask REST API（port 5000），Flask API 收到命令列字串後進行特徵提取，產生 1042 維向量後傳入已載入的 PPO 模型，模型回傳 Benign 或 Malicious 判定結果，並記錄在系統日誌裡供後續分析。

$$\text{Log} \xrightarrow{\text{Logstash}} \text{API} \xrightarrow{f} \vec{x} \in \mathbb{R}^{1042} \xrightarrow{\pi_{\text{PPO}}} a \in \{0, 1\} \qquad (1)$$

攻擊情境模擬

本研究模擬一套完整的三段式 ICS 攻擊鏈，用來驗證防禦系統在眞實威脅情境下的有效性。

第一段：HID 裝置注入。攻擊者把預先載入 Lua 腳本的 HID 裝置插入企業電腦 USB 埠，裝置模擬鍵盤輸入，按 Win+R 開啓執行視窗，並自動注入以下 PowerShell 指令：

```
powershell -NoP -EP Bypass -NoExit -Command
"$p=(68..90|ForEach-Object{'{0}:\m.ps1'
-f([char]$_)}|Where-Object{Test-Path $_}
|Select-Object -First 1);if($p){& $p}"
```

第二段：PowerShell **無檔案執行。**上述指令用 `-EP Bypass`（略過執行原則）和 `-NoP`（略過設定檔）參數啓動 PowerShell，掃描 D 到 Z 磁碟槽找到攻擊腳本後在記憶體中執行，不寫入任何磁碟檔案，來規避傳統特徵碼掃描。

第三段：Modbus TCP **攻擊** PLC**。**攻擊腳本直接對 PLC（目標 IP：實驗場域位址，Port 502）建立 TCP 連線，依序發送三個 Modbus TCP 封包：（1）FC05 寫入線圈 3003（設爲 ON）；（2）FC06 寫入保持暫存器 1999（值設爲 50）；（3）FC05 寫入線圈 3001（設爲 ON），直接改變工業設備的控制狀態，造成實體上的影響。

上述攻擊鏈對應 MITRE ATT&CK 框架中的 T1200（硬體新增）、T1059.001（PowerShell）和 T0836（修改參數）等戰術 [8]。

特徵工程

系統對每筆 Windows 命令列字串進行前處理後提取共 1042 **維**特徵向量，如表 1 所示。

Table 1. 特徵工程組成

特徵類型	維度	說明
TF-IDF（詞彙）	500	word n-gram (1–3)
TF-IDF（字元）	500	char_wb (3–5)
手工特徵	42	見表 2
合計	1042	

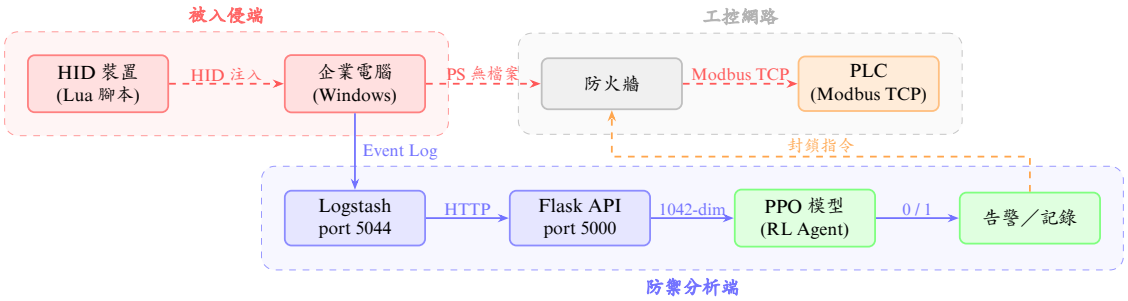


Figure 1. 基於強化學習的工控網路資安防禦系統架構

Table 2. 42 項手工特徵類別

類別	特徵範例
基礎統計	指令長度、特殊字元比例、大寫比例、管道數量
編碼與混淆	Base64 比例、反引號、字串拼接
PS 關鍵字	`IEX`、`-Bypass`、`-Hidden`
LOLBins	`certutil`、`mshta` 等 9 種
網路活動	HTTP/S URL、IP 位址
進階攻擊	AMSI/ ETW Bypass、反射式注入、WMI、COM Object
持久化與竊取	排程任務、Mimikatz

Table 3. PPO 超參數設定

超參數	數值
Policy	MlpPolicy
Learning Rate	3×10^{-4}
n_steps	2048
batch_size	64
n_epochs	10
γ（折扣因子）	0.99
gae_lambda	0.95
clip_range	0.2
ent_coef	0.01
訓練步數	200,000

前處理階段會統一展開 PowerShell 參數縮寫（如 `-NoP` → `-NoProfile`、`-EP` → `-ExecutionPolicy`）並移除完整執行路徑（如 `C:\Windows\...\powershell.exe` → `powershell.exe`），縮小訓練資料和真實執行環境之間的格式落差（Train/Serve Skew）[12]。

PPO 強化學習模型

環境定義。本研究把無檔案攻擊偵測問題建模爲強化學習二元分類任務。依照標準強化學習環境介面，環境定義如下：

- **觀測空間**（Observation Space）：$\mathcal{S} = [0,1]^{1042}$，也就是每筆指令的正規化特徵向量。
- **動作空間**（Action Space）：$\mathcal{A} = \{0,1\}$，0 代表正常（Benign），1 代表惡意（Malicious）。
- **獎勵函數**（Reward Function）：根據預測結果和真實標籤的對應關係給予差異化獎懲，詳見第四節表 5。

超參數設定。PPO 採用 MlpPolicy（多層感知機策略），主要超參數如表 3 所示，實作基於 Stable-Baselines3 [11]。

代理人依序走過訓練集中的每筆命令列字串，根據動作和標籤的對應結果接收獎勵信號，並在每輪訓練回合結束後重置索引來進行下一輪訓練。訓練完成後，特徵器（TF-IDF Vectorizer）用 joblib 序列化，PPO 模型用 zip 格式儲存，由 Flask API 在服務啓動時一次性載入到記憶體，後續推論全程不需要磁碟 I/O。

實驗結果

實驗環境

模型訓練電腦搭載 AMD Ryzen™ 7 8845HS 處理器和 NVIDIA® GeForce RTX™ 4060 Laptop（8GB），作業系統爲 Windows 11。資料集共 3,065 筆 Windows 命令列字串，按照預先分割的方式切分成訓練集（2,451 筆）和測試集（614 筆），如表 4 所示。惡意樣本佔比約 39.5%，屬於輕微不平衡的資料集，透過 Reward Shaping 來調整。

Table 4. 資料集組成

分割	惡意	正常	合計
訓練集	968	1,483	2,451
測試集	242	372	614
合計	1,210	1,855	3,065

Reward Shaping 迭代

本研究透過四次系統性的 Reward Shaping 迭代，逐步優化 PPO 代理人的偵測策略。各次迭代的獎懲設計如表 5 所示，效能演進如表 6 所示。

Table 5. Reward 設計演進（第 4 次中，進階威脅另有加成）

項目	第 1 次	第 2 次	第 3 次	第 4 次
TP	+10	+10	+10	+10†
TN	+1	+2	+3	+3
FP	−2	−5	−10	−10‡
FN	−20	−20	−20	−20§

† 進階威脅 +20；‡ 白名單指令減爲 −3；§ 進階威脅額外 −15~−50

Table 6. 各次迭代效能比較

指標	第 1 次	第 2 次	第 3 次	第 4 次
Accuracy	83.33%	84.25%	97.44%	97.39%
Precision	72.28%	70.95%	99.49%	98.29%
Recall	91.90%	100.00%	93.81%	95.04%
F1-score	80.92%	83.00%	96.57%	96.64%
FP 數	74	86	1	4
FN 數	17	0	13	12

第一次迭代因爲誤報懲罰偏低（FP=−2），模型傾向激進判定，產生 74 筆誤報。第二次把 FP 懲罰提高到 −5 後達到 100% Recall，但誤報反而增加到 86 筆，顯示獎懲幅度還不足以抑制過度敏感的行爲。第三次將 FP 懲罰大幅調升到 −10，誤報從 86 驟降到 1 筆（降低 98.8%），但仍有 13 筆進階攻擊漏報，漏報樣本都是 WMIC 呼叫、反射式 DLL 注入和 LOLBins 繞過等高級手法。第四次在維持相同基礎獎懲數值的前提下，大幅更新資料集並加入高級威脅額外懲罰機制（正確偵測進階威脅額外 +20、白名單指令 FP 減輕至 −3、進階威脅漏報額外 −15~−50），讓 Recall 回升到 95.04%，同時把誤報穩定控制在 4 筆，F1 分數達到最佳的 96.64%。

最終效能評估

最終版本（第 4 次迭代）在 614 筆測試集上的混淆矩陣和效能指標如表 7 所示。

Table 7. 最終混淆矩陣與效能指標

	預測：正常	預測：惡意
實際：正常	368（TN）	4（FP）
實際：惡意	12（FN）	230（TP）

指標	數值	指標	數值
Accuracy	97.39%	F1-score	96.64%
Precision	98.29%	FPR	1.08%
Recall	95.04%	FNR	4.96%

誤報分析。剩餘 4 筆 FP 都是含有系統管理語法的合法 PowerShell 指令（如 `Get-EventLog`、`Export-Csv`、`Get-CimInstance` 等），它們的 TF-IDF 特徵因爲包含檔案路徑操作和系統查詢參數，和攻擊指令有部分重疊。

漏報分析。12 筆 FN 集中在高度進階的攻擊技術，包含：（1）遠端腳本下載執行，用 `Invoke-WebRequest` 搭配 `IEX` 在記憶體中載入；（2）C2 資料外傳，透過 `Invoke-RestMethod` 上傳；（3）反射式 PE 注入；（4）憑證竊取後經 REST API 外傳。這類樣本在訓練集中數量有限，模型沒有學到足夠的辨識特徵，是後續擴充資料集的主要方向。

結論與討論

本研究提出一套基於近端策略優化（PPO）強化學習的工控網路無檔案攻擊即時偵測系統，整合 Windows Event Log 日誌收集、1042 維多層次特徵工程和系統性 Reward Shaping 迭代優化。在模擬 HID 裝置注入、PowerShell 無檔案執行到 Modbus TCP 操控 PLC 的完整三段式攻擊鏈環境中，系統最終達到 97.39% 正確率和 96.64% F1 分數，誤報率只有 1.08%，驗證了強化學習應用在 ICS 防禦場景的可行性和實用性。

研究貢獻。本研究的主要貢獻可以歸納爲三點。第一，建立可以模擬眞實 ICS 三段式無檔案攻擊鏈的仿眞環境，提供涵蓋 HID 注入、PowerShell 無檔案執行和工業協定操控的完整攻擊鏈資料集，填補現有公開資料集在 ICS 無檔案攻擊場景上的不足。第二，提出用 Reward Shaping 動態調整偵測策略的 PPO 訓練框架，透過四次系統性迭代驗證了這個方法在平衡誤報率和漏報率上的有效性，提供可複現的調優方法論。第三，實現 Logstash → Flask API → PPO 模型的即時偵測管道，驗證系統在眞實 Windows 環境中的可部署性，爲集中式 IT/OT 日誌分析結合 RL 防禦的研究方向提供實證基礎。

現有限制。剩餘 12 筆漏報（FN）集中在反射式 DLL 注入、C2 資料外傳和遠端腳本下載執行等高度進階的攻擊手法，主要原因是這類樣本在訓練集中數量不足，模型沒辦法學到足夠的辨識特徵。另外，4 筆誤報（FP）都是含有系統管理語法的合法 PowerShell 指令，它們的 TF-IDF 向量和攻擊指令有部分重疊，顯示純文字特徵在語意層次的區分能力還有限制。

未來展望。後續研究可以從以下方向延伸。首先，引入大型語言模型（Large Language Model, LLM）取代現有的 TF-IDF 特徵工程，利用預訓練語言模型對命令列字串進行語意編碼，提升模型對混淆指令和 Zero Day 攻擊變種的理解能力，有望解決目前 12 筆進階威脅漏報的問題。其次，結合 LLM 的推理能力建構攻擊鏈語意分析模組，讓系統能理解多步指令之間的上下文關係，辨識攻擊鏈的意圖模式，而不是只針對單一指令進行判定。最後，探索以 LLM 驅動的自主代理（AI Agent）進行自動化滲透測試與紅隊演練，主動產生新型攻擊樣本來擴充訓練資料集，形成「以 AI 攻擊訓練 AI 防禦」的對抗式學習框架，持續提升 ICS 環境的整

體資安韌性。

誌謝

本研究感謝兆勤科技在研究資源上的支持與協助。實驗工控場域和設備支援由國立成功大學關鍵基礎設施資安人才培育計畫（CIIS）提供，在此感謝。

參考文獻

[1] TXOne Networks, "OT/ICS cybersecurity report 2024," TXOne Networks, Annual Cybersecurity Report, 2024.

[2] K. Stouffer, S. Lightman, V. Pillitteri, M. Abrams, and A. Hahn, "Guide to operational technology (OT) security," National Institute of Standards and Technology, Special Publication NIST SP 800-82 Rev. 3, 2023. DOI: 10.6028/NIST.SP.800-82r3

[3] A. Bohara, T. Bhattacharya, and P. Narang, "A survey on fileless malware: Attack and detection," IEEE Communications Magazine, vol. 59, no. 3, pp. 68–73, 2021.

[4] A. Varol and M. Iskefiyeli, "Intrusion detection in industrial control systems using machine learning: Modbus TCP traffic analysis," Engineering Applications of Artificial Intelligence, 2025. DOI: 10.1016/j.engappai.2025.110652

[5] M. Raval, P. Bhatt, D. Mehta, and D. Patel, "A survey on security of industrial control systems: Threats, attacks, detection and countermeasures," International Journal of Critical Infrastructure Protection, 2024. DOI: 10.1016/j.ijcip.2023.100672

[6] M. M. Aslam, A. Tufail, R. A. A. H. M. Apong, L. C. De Silva, and M. T. Raza, "Scrutinizing security in industrial control systems: An architectural vulnerabilities and communication network perspective," IEEE Access, 2024. DOI: 10.1109/ACCESS.2024.3394848

[7] Z. Yang, L. He, P. Cheng, and J. Chen, "Mismatched control and monitoring frequencies: Vulnerability, attack, and mitigation," IEEE Transactions on Dependable and Secure Computing, vol. 22, no. 1, pp. 16–33, 2025. DOI: 10.1109/TDSC.2024.3384146

[8] MITRE Corporation, MITRE ATT&CK: Adversarial tactics, techniques, and common knowledge, https://attack.mitre.org, 2024.

[9] G. Brockman et al., "OpenAI gym," in arXiv preprint arXiv:1606.01540, 2016.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[11] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," Journal of Machine Learning Research, vol. 22, no. 268, pp. 1–8, 2021.

[12] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.