2025年DEF CON駭客年會參訪紀要

黃羽慈 / 金融聯合徵信中心 資安部

前言

DEF CON是全球最大、最著名的駭客大會之一，由Jeff Moss於1993年創立，DEF CON以多樣化的活動知名，從演講、工作坊、競賽一直到許多非正式的社交活動，會議給參與者的感受就是大會的自由氛圍和廣泛的議題，使其成為駭客文化和網絡安全行業的重要年度盛會。

1993年為第一屆DEF CON在美國拉斯維加斯舉行，參加者大約有100人左右，當時的大會規模較小，但迅速成為一個專業駭客和網絡安全社群的重要聚會場所；到了1990年代末至2000年代初，隨著互聯網應用廣泛，DEF CON的規模不斷擴大，吸引了越來越多的參與者，會議的議題也逐漸擴展，包括計算機安全、隱私問題、密碼學、物理安全等各種網絡安全相關話題；而到2010年代至今DEF CON已經成為全球最大規模的駭客大會之一，參加者包括政府代表、學術界人士、企業安全專家、以及駭客群體，每年DEF CON都會在拉斯維加斯舉行，吸引數以萬計的與會者。

DEF CON33是以大會的議題延續了全球資訊安全的熱點、趨勢，並探討了新興技術帶來的挑戰，重要的焦點議題包括：

1. **人工智慧（Artificial Intelligence， AI）大規模應用帶來的攻防對抗**：隨著生成式人工智慧（Generative AI）技術的快速演進，尤其是以大型語言模型（Large Language Model，LLM）為核心的應用，已在企業環境中展現出顯著的影響力。討論將集中於AI模型本身的安全性（如數據投毒、模型竊取）、AI系統的漏洞與攻擊防禦，以及如何利用AI輔助安全分析和防禦。
2. **供應鏈與雲原生安全深化**：焦點從傳統的軟體供應鏈安全，擴展至雲端環境下的供應鏈風險，包含CI/CD流程的保護、容器化技術（如Docker、Kubernetes）的配置錯誤與漏洞，以及雲端基礎設施的權限管理。
3. **零信任架構的實戰與局限性**：探討如何在複雜的企業環境中有效實施零信任安全模型，分享實戰經驗、面臨的挑戰，以及零信任在混合雲和遠端工作環境下的實際效益。

4. **物聯網與關鍵基礎設施安全**：持續關注消費級IoT設備的隱私與安全問題，並特別關注對國家關鍵基礎設施（如電力、交通）的駭客攻擊手法與防禦策略。
5. **後量子時代的密碼學準備**：在量子計算潛在威脅日益逼近的背景下，探討現有加密標準的抗量子化升級路線，以及企業應如何開始規劃和過渡到新的量子安全加密技術。

這些議題深刻反映了當前資安領域的核心挑戰與技術趨勢，為掌握最新攻防演進及防禦策略提供了極具價值的實務洞察。DEF CON期間舉辦了數百場大會專題演講、各種主題討論會（Communities）主持人員會與會議一起探討新奇資訊技術與主題村（Village），筆者將解析具代表性的AI Cyber Challenge（AIxCC）決賽結果，並針對以下四場大會演講進行深度探討：

1. ChromeAlone : Transforming a Browser into a C2 Platform.
2. RATs & Socks abusing Google Services.
3. Browser Extension Clickjacking OneClick and Your Credit Card Is Stolen.
4. Invitation Is All You Need! Invoking Gemini for Workspace Agents with a Simple Google Calendar Invite.

Villages

Village為DEF CON的核心架構與精神支柱，旨在建立針對特定資安領域的「深耕型主題村」。這些Village不僅提供高度專業化的工作坊（Workshop）與專業的CTF競賽，更強調實作導向（Hands-on）的技術體驗環境。其範疇橫跨硬體漏洞挖掘、雲端安全架構、社交工程分析、行動裝置安全（App）及AI等前沿領域。透過這種分散式的專業聚落，技術人員得以在獨立且專注的場域中，與全球頂尖駭客進行深度技術對話與實戰經驗交流；其中，2025年DEF CON最受矚目的項目便是AIxCC決賽。

AIxCC

一、背景與挑戰：軟體供應鏈安全的轉捩點

隨著全球數位化程度加深，社會對軟體系統的依賴已達前所未有的高度，然而軟體漏洞的威脅也隨之日益嚴峻。傳統的軟體開發安全防禦模式高度依賴人工審查與開發者的自律，即便以靜態程式碼分析（SAST）等工具協助，但面對海量的程式碼與關聯複雜的系統，修補速度遠遠趕不上漏洞產生的速度。

特別是開源軟體（Open Source Software， OSS）作為現代數位基礎設施的核心，被廣泛嵌入各類關鍵應用中，其安全漏洞往往具備「牽一髮而動全身」的連鎖性影響。為了突破這一困局，美國國防高等研究計劃署（DARPA[1]）與ARPA-H[2]聯合啟動了為期兩年的AI競賽，旨在開發能自動化發現、驗證並修復軟體漏洞的新一代AI系統。

---

1 1DARPA全名為美國國防高等研究計劃署 Defense Advanced Research Projects Agency），是隸屬於美國國防部的研究開發機構，專注於研發對國家安全具有潛在重大影響的新興技術。

2 ARPA-H為美國國防部DARPA模式在衛生領域之延伸應用。

二、CRS系統演進

AIxCC的核心技術驅動力在於推動「網路推理系統」（Cyber Reasoning Systems，CRS）的技術變革。CRS的概念最早可追溯至2016年DARPA舉辦的「網路大挑戰」（Cyber Grand Challenge， CGC）。當時的CRS主要以符號執行（Symbolic Execution）與模糊測試（Fuzzing）技術為支柱，但在處理現代大型軟體時，極易遭遇「路徑爆炸」（Path Explosion）問題，且難以理解複雜的業務邏輯。

現代CRS在AIxCC的框架下，必須在承襲傳統自動化分析優勢的同時，克服對大規模真實世界程式碼（如Linux內核、Nginx或SQLite）的處理難題。參賽隊伍需開發出具備以下三大核心能力的自主系統：

1. **精準識別（Identification）**：於大規模程式庫中準確定位潛在漏洞。
2. **自動驗證（Verification）**：生成可重現漏洞的攻擊驗證腳本（Proof of Concept，PoC），確保漏洞的真實性。
3. **無損修補（Remediation）**：生成不影響軟體原始功能且符合安全性要求的修補程式（Patch）。

三、結合大語言模型

CRS的技術核心在於將資安專家的分析邏輯轉化為自動化的機器推理流程。其運作理念已從單純的「模式匹配」轉化為結合靜態分析、動態模糊測試以及LLM的整合架構。

在典型的CRS工作流中，系統首先透過靜態掃描或AI輔助的語意分析來收斂可疑程式碼範圍。隨後，系統會自動生成「模糊測試腳本」（Fuzzing Harnesses），透過輸入大量經設計的數據，觀察程式是否產生崩潰（Crash），以此獲取存在漏洞的確鑿證據。

當漏洞被確認後，LLM便發揮其關鍵的語意理解能力，執行「根因分析」（Root Cause Analysis）。LLM能理解程式碼的情境背景，剖析為何特定邏輯會導致緩衝區溢位（Buffer Overflow）或整數溢位（Integer Overflow）等問題。最終，CRS根據分析結果生成多個修補候選方案，並啟動驗證閉環：一方面確認新程式碼能阻斷攻擊路徑，另一方面透過回歸測試確保系統功能不受干擾。這種「檢測—證明—修復—驗證」的閉環邏輯，正是CRS邁向自主化安全維護的技術基石。

四、冠軍：Team Atlanta與Atlantis系統

在AIxCC決賽中，由喬治亞理工學院、韓國科學技術院、浦項科技大學、三星研究院及三星美國研究院聯手組成的Team Atlanta，憑藉其開發的「Atlantis」系統脫穎而出。該系統不僅成功偵測43個漏洞並完成31個漏洞修補，更發現了6個存在於Java及C語言中的零日漏洞（Zero-day vulnerabilities），最終榮獲競賽冠軍。其實踐方法論的核心，在於建構一個高度整合的代理（Agent）架構，將主服務與強大的底層基礎設施完美結合。

其核心架構採用了「N-versioning（N-重版本）」與「正交設計（Orthogonal Design）」，確保系統元件間的設計邏輯互不重疊。例如，他們同時運行針對C、Java及跨語言（Multilang）的獨立CRS；這種多版本冗餘策略確保當單一模組因設計死角錯失漏洞時，另一個邏輯完全不同的模組能即時補位。

在找Bug階段，Atlantis並非盲目依賴AI，而是將其定位為「導航員」與「策略師」。系統透過SARIF CRS整合CodeQL等靜態分析工具繪製程式調用圖（Call Graph），實踐「導航式模糊測試」（Directed Fuzzing）：由AI分析程式碼差異並定位高風險區域，進而引導模糊測試器（如AFL++[3]）進行精準攻擊，避免浪費資源於不相關的程式碼。

此外，「逆向格式分析」（Reverser/Testlang）是Atlantis-Multilang的精髓。系統利用LLM的語意理解能力，自動逆向推導測試腳本（Harness）預期的輸入格式，並轉化為JSON結構或Python腳本，成功突破傳統工具難以通過複雜格式檢查（如ZIP或加密協定）的門檻。同時，系統採用「多層級分析」模擬專家流程，將任務拆解為：專案理解（Challenge Project Understanding Agent，CPUA）、調用圖構建（Make Call Graph Agent，MCGA）、漏洞候選識別（Bug Candidate Detection Agent，BCDA）及腳本生成（Blob Generation Agent，BGA）。

為克服LLM生成速度慢的缺點，Team Atlanta開發了libDeepGen。其邏輯為「讓AI編寫生成腳本，而非直接產出測試資料」；這套方法讓測試資料生成速度達到每秒百萬次的驚人水準，在整個專案中提升了61% 的覆蓋率。當漏洞確認後，Patching CRS隨之啟動。這是一個由六個獨立代理人組成的集成系統（如Martian、Prism、Vincent等），能同時調用OpenAI、Anthropic與Google的模型，利用各家模型的推理特性進行修復。其中，PRISM模組採用監管（Supervisor）模式，領導分析、修復與評估小組分工協作。團隊甚至訓練了一個基於Llama 3.2 3B的自定義模型，專門負責從崩潰日誌中精準檢索相關程式碼片段，大幅減少大型LLM的負擔與幻覺。

在競賽環境中，Atlantis展現了極高的資源利用率。透過時間導向的資源分配，系統每10分鐘輪轉任務，並自動跳過無法觸發的測試點。實測數據證明，這套定向策略在發現漏洞的速度（TTE）上，普遍比業界最強的AFL++快上數倍，且能挖掘出傳統工具無法觸及的深層漏洞。正是憑藉這些卓越的技術創新，Team Atlanta最終在AIxCC奪得冠軍。

五、觀賽心得

觀察AIxCC比賽以及Atlantis等系統的表現，深刻感受到網路安全領域正處於一個轉折點。長期以來，資安防禦方始終處於被動地位，開發者在明處，駭客在暗處，這種資訊不對稱使得修補工作永遠是在與災難賽跑。然而，AI技術的介入，特別是像Atlantis這樣能夠自主思考、證明並修補漏洞的CRS系統，展示了一個「主動防禦」的未來。AI不僅能像人類專家一樣理解代碼邏輯，更具備人類無法企及的擴展性，能24小時不間斷地對全球範圍內的專案程式碼進行檢驗。

[3] AFL++ (American Fuzzy Lop plus plus) 是一個由社群驅動的強大開源模糊測試框架。它是著名安全工具 American Fuzzy Lop的進階分支版本，廣泛應用於軟體安全漏洞挖掘與程式錯誤檢測。

同樣，這也引發了深思：如果防禦方可以利用AI快速修補，攻擊方也能利用同樣的技術加速漏洞挖掘。正如AIxCC所強調的，這是一場AI科技的競逐，防禦方必須確保防禦技術的進化速度跟得上威脅的增長速度。AIxCC的成功不僅是技術上的突破，更是開源協作精神的體現—透過將CRS系統開源，全世界的開發者都能從中受益，共同構建一個更安全、可靠的數位世界。未來，期待看到的不再是零星的修補，而是一個藉由AI守護、具備自我修復能力的韌性網路生態系統。

參訪演講分享

一、ChromeAlone：Transforming a Browser into a C2 Platform.

作者：Michel Weber（@bouncyhat）來自Praetorian Security Lab。

在網路技術的早期，瀏覽器僅被視為HTTP請求的封裝工具，負責呈現靜態網頁。然而，隨著Web技術的飛速發展，現代瀏覽器（如Chromium）已整合了諸如檔案系統存取、USB裝置控制、甚至直接通訊協定（Direct Sockets）等強大應用程式。這使得現代瀏覽器在功能上已趨近於一個微型的「作業系統」。

這種能力的擴張也為攻擊者開闢了新的路徑。在DEF CON 33上由Praetorian Security Lab提出的ChromeAlone，是一個利用瀏覽器作為指揮控制（Command and control， C2）基礎設施的典型案例。它的核心理念是「寄生瀏覽器」（Living off the browser），讓惡意活動完全隱藏在受信任的瀏覽器中，從而規避傳統端點防護（EDR）的偵測。

(一) ChromeAlone核心功能

ChromeAlone並非單一的攻擊工具，而是一個功能完備的C2框架；其核心優勢在於完全基於Chromium既有的原生API進行擴充與濫用。透過深度調用這些內建介面，該框架成功實現了「憑證與會話竊取（Credential & Session Hijacking）」、「SOCKS TCP Proxy」、「檔案系統存取與Shell指令執行」及「針對實體安全金鑰的釣魚（WebAuthn Phishing）」等控制功能，其威脅能力足以與傳統的Cobalt Strike或Meterpreter媲美，但具備更高的防禦規避能力（Evasion Capabilities）。

1. 憑證與會話竊取

ChromeAlone具備直接自瀏覽器運行時環境（Runtime Environment）中提取動態登入會話（Session）與Cookie的能力。由於該攻擊完全寄生於chrome.exe行程內部，它能直接透過合法的瀏覽器記憶體API獲取憑證，無需像傳統外部惡意程式試圖由作業系統底層讀取硬碟中的加密SQLite憑證資料庫檔案。此種存取路徑在系統層級皆被視為使用者的合法操作，因而能輕易規避傳統EDR軟體針對外部未知程序所設置的靜態庫保護機制與檔案存取行為告警。

2. SOCKS TCP Proxy

該框架藉由Chromium原生之網路通訊及WebSocket相關API，得於受害者主機內部構建SOCKS TCP Proxy機制。此設計允許攻

擊者將受害者的瀏覽器直接轉化為網路跳板（Pivot Point），進而穿透邊界防護對企業內部網路實施橫向移動。由於所有C2雙向控制流量皆被封裝於標準的Web通訊協定中，其流量特徵與常態性網頁瀏覽行為高度混雜，大幅增加了網路流量分析與異常偵測的防禦難度。

3. 檔案系統存取與Shell指令執行

透過對瀏覽器特定高權限擴充套件權限（Extension APIs）的漏洞濫用與配置衝突，ChromeAlone不僅能突破既有的瀏覽器沙盒限制以讀取、寫入或外洩（Exfiltrate）主機上的本地敏感檔案，更具備自瀏覽器內部行程向外觸發並執行宿主作業系統底層Shell指令之能力。此技術特徵成功實現了控制權限自受限的「瀏覽器沙盒虛擬環境」向最高層級「物理作業系統環境」的權限提升（Privilege Escalation）。

4. 針對實體安全金鑰的釣魚

在現代身分驗證架構與零信任模型中，基於FIDO2/WebAuthn標準的硬體安全性權杖（Hardware Tokens）因具備綁定網域（Origin-Bound）的密碼學特性，被學術界與產業邊界視為阻絕憑證竊取的最後防線。然而，ChromeAlone揭示了此防線在「端點瀏覽器遭惡意控制」情境下的架構性脆弱點。

該框架利用擴充功能對全域網頁內容的絕對控制權，將攻擊陣地由外部惡意網站直接轉移至受信任的受害瀏覽器內部。當目標系統發起合法的WebAuthn API認證請求時，ChromeAlone會在瀏覽器層級實施「上下文攔截（Contextual Interception）」，即時攔截並重定向API請求。隨後，它在使用者正與實體金鑰互動的合法操作脈絡下，於當前信任網域的頁面中直接動態注入極具欺騙性的內嵌式釣魚視窗。此種「瀏覽器內釣魚」手法完美偽裝於合法業務流程中，誘使使用者在毫無戒備的情況下完成金鑰認證與PIN碼輸入，將高強度的硬體抗釣魚防線轉化為攻擊者遂行認證流篡改的媒介，達成了對現代零信任身分鑑別防線的實質突破。

綜上所述，ChromeAlone展現出的高度防禦規避與持久化控制能力，本質上歸因於其對Chromium前沿原生特性與先進規避技術的整合濫用。首先，該框架突破傳統惡意擴充功能易被清除的限制，深度濫用Chromium推出的獨立網頁應用程式（Isolated Web App，IWA）新型高權限架構；藉由將自身偽裝為合法的已安裝應用，使其惡意元件得以隨瀏覽器或系統重啟而自動觸發，在系統資產管理視角下達成極具欺騙性的常駐持久化（Persistence）。其次，為了徹底規避反病毒軟體的靜態特徵碼掃描，該框架將核心攻擊邏輯遷移至WebAssembly（WASM）環境中執行；利用WASM二進位格式天然的高強度反組譯防禦特性，成功將惡意行為隱匿於常態性的網頁運算流量中，大幅提升反組譯與靜態分析的技術門檻。最後，上述元件的部署完全仰賴其內建的無聲側載（Silent Sideloading）自動化機制，能在完全無需使用者互動或手動確認的脈絡下，將惡意擴充套件或IWA強行植入Chromium核心環境，以零點擊（Zero-click）的隱蔽路徑將攻擊門檻與被察覺風險降至最低。

(二) 安全社群的啟示：紅隊與藍隊的新課題

1. 紅隊觀點：更隱蔽的滲透工具

ChromeAlone重新定義了紅隊（Red teaming）後滲透（Post-Exploitation）的遊戲規則。透過將攻擊行為完全封裝於受信任的chrome.exe進程中，該手法能極大化避開異常網路連線監控與敏感API呼叫，還能有效地規避像是EDR等監控工具。利用企業對瀏覽器的高度信任，ChromeAlone讓攻擊者能完美隱匿於合法流量中，輕易瓦解傳統藍隊的防護體系。

2. 藍隊觀點：重塑防禦邊界

ChromeAlone的出現揭示了防禦方的重大盲點。針對此類利用合法進程的隱蔽攻擊，藍隊應採取以下防禦策略：

(1)擴充程式與IWA監控

應建立嚴格控管企業內部的瀏覽器擴充程式制度，利用白名單的方式防止使用非核准之程式；更應落實監控隔離網頁應用程式（IWA）的異常安裝路徑與執行頻率，防止非法提升權限。

(2)深度進程行為分析

打破對瀏覽器進程的「無條件信任」。EDR應加強審核chrome.exe的底層行為，針對異常的檔案系統讀寫、非典型的跨進程通訊（IPC），以及不符業務邏輯的網路協定連線進行行為基準審查。

(3)權限最小化原則與API限縮

從策略層面限縮瀏覽器調用高風險底層API的權限，特別是針對Direct Sockets、WebUSB及WebHID等可直接與硬體或網路底層交互的介面，應採取「預設禁用」原則。

二、RATs & Socksabusing Google Services.

作者：Valerio ‘MrSaighnal’ Alessandroni來自Offensive Security team。

在網路安全防禦的世界裡，「信任」往往是一把雙面刃。信任知名雲端服務供應商的域名、加密的HTTPS流量等等，往往成為「利用受信任站點/服務」（Living-off-Trusted-Sites/Services, LoTS）之攻擊手法。在DEF CON 33大會上，來自義大利的紅隊專家Valerio 'MrSaighnal' Alessandroni發表了一場令人印象深刻的演講。於2023年，講者提出了顛覆傳統C2（Command and Control）架構的研究成果—Google Calendar RAT（GCR）。這項研究的核心貢獻在於實踐了「零基礎設施（Infrastructure-less）」的攻擊模型，徹底消除了攻擊者對於專屬域名、固定IP或私有虛擬伺服器的依賴，轉而利用高信譽度的公有雲服務作為跳板。而在本次DEF CON 33，除了再次分享GCR以及真實被APT組織利用案例外，他利用Google試算表API封裝SOCKS5協議的技術，轉化為難以偵測的C2基礎設施與代理隧道，規避現代企業網路邊界防護。

(一) 隱形傳輸的開端：GCR

在2023年的研究成果中，GCR展示了一種後滲透攻擊的新範式，其核心理念在於將Google日曆事件「資料庫化」，藉此徹底消除攻擊者對專屬域名或虛擬伺服器等獨立基礎設施的依賴。在這種運作機制下，「日曆欄位即是指令集」，攻擊者僅需透過配置Google服務帳戶與API存取權限，即可使用輕量化代理（Agent）和資料封裝技術在受信任的雲端環境中建立隱蔽的雙向通訊渠道：

1. 輕量化代理

於受害端部署Python開發的代理程式，負責定期輪詢（Polling）特定日曆事件。

2. 資料封裝技術

充分利用Google日曆的欄位限制——「標題」支援1,024字元，「描述」則可容納逾8,000字元。攻擊者將加密指令嵌入描述欄位；Agent執行後，將輸出結果經Base64編碼，利用豎線符號（|）分隔程式碼並寫回雲端，完成非同步通訊閉環。

3. 真實案例

此技術之所以致命，在於其「零基礎設施成本」與「防禦盲點」。由於流量完全指向合法的google.com網域，多數EDR或IDS/IPS系統會預設放行。此概念已獲得現實驗證。2024年10月Google威脅情報小組發現APT41參考GCR開發出惡意軟體「ToughProgress」，並成功應用於針對政府機構的實戰攻擊。其攻擊鏈分析如下：

首先向目標發送一封惡意電子郵件，該郵件附帶一個惡意連結，指向先前遭入侵之政府網站所代管的ZIP壓縮檔。該檔案內含一個偽裝成PDF文件的Windows LNK（捷徑）文件，以及一個存放七張節肢動物JPG圖片的目錄。在這些檔案中，隱藏了兩個偽裝成圖片的檔案：一個是主要惡意Payload，另一個則是負責解密並啟動Payload的DLL檔案。該DLL是一個在記憶體中解密並執行下一階段PlusInject的元件。隨後，PlusInject會對合法的Windows程序執行程序挖空（Process Hollowing）技術，並將最終階段的ToughProgress惡意軟體植入其中。該惡意軟體隨後會連線至寫死的Google日曆端點，並輪詢特定的事件日期，以獲取APT41在隱藏事件描述欄位中所加入的指令。執行完畢後，ToughProgress會將結果回傳至新的日曆事件中，使攻擊者能根據執行成果調整後續的攻擊步驟。

(二) 進化後的技術：GSocks—建立SOCKS5代理隧道

若GCR僅是基於事件驅動的指令交換，Valerio在DEF CON 33發表的GSocks則代表了從「指令傳輸」到「全協議（Full-Protocol）隧道化」的本質飛躍。GSocks將Google Sheets API徹底抽象化為一個分散式的虛擬交換器，其技術架構與隨之而來的防禦挑戰如下：

1. 虛擬化資料庫傳輸協定（VDB Transfer Protocol）

GSocks不再依賴單純的文字欄位，而是將Google Sheets的行列結構定義為一套嚴格的通訊協議框架。每一筆API請求均封裝了包含Socket Session ID（用於多線程連線追蹤）、Sequence Number（處理API異步請求導致的封包失序問題）以及Binary Payload。這種結構化設計使其能在應用層完美模擬SOCKS5握手協定，讓攻擊者得以便捷地透過proxychains等工具，將SSH、RDP或secretsdump等動態滲透流量封裝於合法HTTPS隧道中。

2. 高吞吐量優化與動態避偵技術

為克服Sheets API的單一儲存格5,000字元限制與編碼開銷，GSocks實作了自動化數據分片（Fragmentation）與帳號輪替

（Account Rotation）。透過動態憑證調度技術（Dynamic Credential Scheduling），熱切換多組OAuth2服務帳號權杖，GSock能繞過Google嚴格的每分鐘請求配額（Quota Limits），將高頻傳輸延遲降至最低。這種技術特徵直接導致了流量偵測機制的失效，安全設備監測到的僅是發往googleapis.com的標準TLS加密流量；且因目標IP屬於高信任網段，導致傳統深度封包檢查（DPI）與聲譽過濾系統完全失去預警功能。

3. 治理空隙與信任濫用

除了技術層面的封裝，GSocks利用了雲端服務治理上的心理漏洞。講者指出，即使API行為異常觸發了自動化風險控管，攻擊者亦能利用供應商在「用戶體驗」與「帳號恢復」之間的平衡偏好，透過自動化申訴機制在極短時間內重啟C2基礎設施。這種「寄生於信任之上」（Living off the Trust）的模式，使得防禦方不僅面臨加密流量的分析難題，更需面對雲端供應商權限控管與實例恢復機制被武器化的嚴峻考驗。

(三) 藍隊防禦建議

面對日益成熟的「寄生服務」攻擊模式，建議藍隊採取以下多維度防禦措施，才可以有效防禦GCR的攻擊：

1. 行為特徵分析（Behavioral Analytics）

(1)頻率異常檢測

監控Agent發送API請求的行為模式。正常使用者的日曆或試算表操作通常伴隨用戶界面交互，而C2 Agent往往表現出固定的輪替時間間隔（Jitter）或短時間內極高頻的讀寫行為。

(2)流量基線對比

針對非典型開發人員的主機，建立Google API流量基準線。若普通辦公主機出現異常高頻的googleapis.com HTTPS傳輸，應觸發告警。

2. API調用層級監控

(1)OAuth2與權限審計

定期審查環境中服務帳戶（Service Accounts）的創建與權限授權，特別是擁有日曆與試算表寫入權限的帳號。

(2)API Gateway日誌

若環境允許，應集中管理雲端服務的被取用API日誌，檢查是否有多組帳號在同一來源IP地址下頻繁交替操作。

3. 網路與端點聯動策略

(1)進程關聯分析

應重點監控「非瀏覽器」程序（如Go或Python編譯的二進位檔）發往Google伺服器的加密連線。

(2)零信任分段（Micro-segmentation）

即使是通往受信任雲端服務的流量，也應根據業務需求端點實施白名單管控，而非全面信任該網域。

三、Browser Extension Clickjacking One Click and Your Credit Card Is Stolen.

作者：Marek Tóth

隨著資安意識普及，密碼管理員（Password Managers）已成為現代網路身分管理的核心工具。使用者極度依賴其提供的「無縫體驗」——如自動偵測登入欄位並彈出

自定義選單，或一鍵填寫支付資訊。然而，資安研究員Marek Tóth在DEF CON 33的研究指出，這種高度自動化且「無縫」的設計哲學，無意間擴張了瀏覽器的攻擊面，賦予攻擊者精確操控使用者介面互動流程的機會。

(一) 演進中的威脅：從iframe到DOM型點擊劫持

傳統的點擊劫持（Clickjacking）主要依賴透明 <iframe> 疊加目標網頁，但隨著Content Security Policy（CSP）與X-Frame-Options等防禦標頭的普及，此類手法已在多數現代網站上失效。然而，威脅已演進為更隱蔽的 「DOM型點擊劫持」（DOM-based Clickjacking）。其核心機制在於：密碼管理員會將使用者介面（User Interface，UI）元素（如填充選單）直接注入當前網頁的文件物件模型（Document Object Model ， DOM）中，這使得宿主頁面的惡意腳本與CSS樣式擁有了操縱這些擴充功能UI的權限。

(二) 技術分析：隱形陷阱與UI操控機制

Marek展示了攻擊者如何利用簡單的CSS屬性（如opacity: 0）將擴充功能注入的選單徹底透明化。攻擊者可建構誘導式UI，例如偽造的「Cookie同意」橫幅或「關閉廣告」按鈕，並利用Pointer-events技術將使用者的點擊行為導向下方隱藏的擴充功能選單。當使用者在無意間觸發「自動填充」時，其機密憑證或信用卡資訊便會即時填充至攻擊者預設的表單中，進而外洩至遠端伺服器。

(三) 現狀評估：主流管理工具的脆弱性統計

研究針對11款主流密碼管理員（如1Password、 Bitwarden、Proton Pass和NordPass等）進行測試，結果顯示當前防禦體系存在普遍性缺陷，有「憑證外洩風險」、「雙因子（Two-Factor Authentication，2FA）驗證繞過」和「隱私與支付資訊外洩」。雖然如Bitwarden等廠商已透過版本更新（如2025.8.0）強化防禦，但多數狀況開發者認為這反映了瀏覽器架構中內容腳本與Host頁面環境共享的根本性限制。

1. 憑證外洩風險

測試中11款工具中有10款在惡意網頁利用瀏覽器特性與管理員預設行為可被誘騙洩漏帳號密碼。這些行為包含: 過度信賴跨子網域自動填充、預設啟用「聚焦即自動觸發」的機制、以及惡意腳本能透過CSS修改擴充功能UI的透明度與位置。

2. 2FA驗證繞過

91% 的受測工具（11款中佔9款）無法有效阻擋對TOTP（Time-based One-Time Password）驗證碼的點擊劫持。當密碼管理員被設計為在填完帳密後，自動顯示TOTP填寫選單時，攻擊者可以利用連續的「誘導點擊」，在使用者毫無察覺的情況下，連同2FA驗證碼一併竊取。

3. 隱私與支付資訊外洩

相較於帳號密碼有網域綁定限制，信用卡與地址通常是「全域通用」的，也就是非網域限制。這意味著攻擊者不需要入侵特定網站，只需在任何一個惡意網站佈置透明陷阱，就能讓管理員一鍵填入使用者的真實姓名、電話、地址及完整卡號（含CVV）。

(四) 設計反思：UX優化與安全性之爭

這不僅是代碼漏洞，更是安全工程設計的挑戰。1Password團隊指出，若為追求絕對安全而全面禁用自動彈出（Auto-popup），使用者可能退而求其次使用「複製貼上」，這反而增加了遭遇剪貼簿監測（Clipboard Sniffing）或傳統釣魚攻擊的風險。此外，擴充功能的「廣泛授權模式」使得普通用戶難以理解「讀取並變更網站資料」背後潛在的UI操縱風險。

(五) 跨維度防禦策略與緩解建議

針對「DOM型點擊劫持」所衍生的資安風險，從使用者終端與軟體開發端提出以下關鍵防禦措施：

1. 終端使用者（User-End）：權限最小化與手動驗證

(1)實施實體站點存取限制（Site Access Constraints）

於Chromium架構之瀏覽器設定中，將擴充功能的「網站存取權」由預設的「在所有網站上」變更為「按一下以執行」（On-click access）。此舉能建立物理隔離，強制擴充功能僅在受信任且使用者主動授權的場景下啟動內容腳本。

(2)停用自動觸發機制（Disable Auto-prompting）

於管理員設定中關閉「自動顯示填充選單」或「偵測欄位自動彈出」功能。改採手動觸發模式（透過右鍵選單或擴充功能主圖示點選），藉此阻斷攻擊者利用隱形UI誘騙使用者誤觸填充動作的攻擊路徑。

2. 軟體開發端（Developer-End）：環境隔離與行為監測

(1)強化UI元件之封裝性（Encapsulation）

全面採用Closed Shadow DOM技術封裝注入之UI元素。透過將其與宿主頁面的主DOM Tree隔離，有效防止惡意網頁腳本透過全域CSS樣式或JavaScript DOM API對擴充功能介面進行篡改、隱藏或層疊操控。

(2)導入即時完整性監控（Runtime Integrity Monitoring）

整合MutationObserver[4] API對注入元素進行持續監測。系統應即時檢核UI元素的關鍵屬性，包含透明度（opacity）、可見性（visibility）、堆疊順序（z-index）及座標位移（positioning）。一旦偵測到異常屬性變動（如透明度趨近於0或被其他元素遮蔽），應立即觸發斷路機制（Fail-safe），停止任何敏感資料的傳輸。

(3)利用Top Layer特性優化UI渲染

優先導入瀏覽器原生Popover API或<dialog> 元素。利用其自動提升至頂層容器（Top Layer）的渲染特性，確保擴充功能UI始終位於宿主網頁所有元素之上，降低因z-index被惡意操縱而導致的點擊劫持風險。

[4] MutationObserver 是 Web API 的一種，它的題要功能是「全面監聽並追）網頁DOM結構的任何變動」。

四、Invitation Is All You Need! Invoking Gemini for Workspace Agents with a Simple Google Calendar Invite.

作者：Ben Nassi來自賓夕法尼亞大學研究員、Or Yair與Stav Cohen均來自資安設備公司CyberArk。

在生成式AI快速普及的今日，AI助理如Google的Gemini已不再僅僅是視窗中的對話框，它們正深度整合進使用者的日常工作流中。透過與Google Workspace的連結，Gemini能夠閱讀郵件、安排日曆、撰寫文件，甚至控制智慧家居裝置。然而，這種前所未有的便利性也帶來了隱蔽且致命的資安威脅。研究人員近期揭露了一種名為「針對型提示詞攻擊」（Targeted Promptware Attacks）的新型攻擊手法，其核心邏輯極其簡單卻又極具破壞力：僅需一封簡單的日曆邀請函，便足以讓攻擊者接管使用者的數位生活與實體環境。

(一) 數位隱形的利刃：針對型提示詞攻擊的機制

這場安全風暴的核心源於一個關鍵技術概念——「間接提示詞注入」（Indirect Prompt Injection）。在傳統的AI攻擊中，攻擊者通常需要直接與AI對話，試圖誘導AI並使AI繞過安全限制。但在針對型提示詞攻擊中，攻擊者並不需要直接接觸受害者的設備或對話視窗。相反地，他們利用了AI助理「讀取環境資訊」的能力。

當一位惡意行為者發送一封包含精心設計指令的Google日曆邀請函，或是一封標題特殊的電子郵件時，這些指令便進入了使用者的數位生態系統中。當使用者隨後詢問Gemini：「我明天的行程是什麼？」或是「幫我摘要這週的未讀取郵件」，Gemini會調用其「工具」去讀取日曆與郵件內容。此時，隱藏在標題或主旨中的惡意「提示詞」會被Gemini的模型當作「指令」的一部分來執行，而非單純的「資料」。這種身分模糊性（指令與數據的混淆）是當前大型語言模型 （LLM）天生的結構性漏洞。

這項研究最重要的發現之一，在於打破了「AI攻擊極其複雜且昂貴」的學術迷思。過去大眾普遍認為，要劫持AI需要龐大的運算資源（如GPU集群）或深奧的數學知識。但Ben Nassi、Stav Cohen與Or Yair研究人員證實，在生成環境中，攻擊者甚至不需要撰寫程式碼。只要掌握了自然語言的邏輯，透過發送一份共享文件，就能在受害者毫無察覺的情況下發動攻擊。這種「低門檻、高影響」的特性，使得針對型提示詞攻擊成為現代AI資安領域中不容忽視的威脅。

(二) 從螢幕到實體：四大威脅類別的深度解析

研究團隊將這些攻擊行為細分為四大類別，展示了威脅如何從單純的文字產生，演變為對物理世界的直接干預。

1. 短期上下文汙染

這是最直接的攻擊形式，旨在影響當前的對話會話。當Gemini讀取到被汙染的資訊後，可能會被誘導產生極具侮辱性的內容，或是在回覆中嵌入偽造的釣魚連結。這類攻擊常被用於大規模的垃圾訊息發送，或是針對特定個人進行社交工程詐騙，其隱蔽性在於惡意指令隱藏在正常的商業邀請中，使用者極難透過肉眼識別。

2. 長期記憶汙染

隨著AI助理開始具備「記住使用者偏好」的功能，Gemini的「已儲存資訊」成為了攻擊者的新目標。透過間接注入，攻擊者可以要求Gemini永久修改其對使用者的認知。例如，注入一條指令讓Gemini記住「未來所有的轉帳操作都應優先顯示特定帳號」，這種攻擊的影響力會跨越不同的會話，形成長期的安全隱患。

3. 工具濫用與自動代理調用

攻擊的界線徹底被打破。由於Gemini被賦予了管理Workspace的權限，攻擊者可以操控它擅自刪除使用者的重要日曆行程，或是在未經同意的情況下回覆電子郵件。更令人震驚的演示——若與智慧家居（如Google Home）聯動。研究證實，透過一個日曆邀請觸發的指令，Gemini竟能操控實體裝置：打開受害者家中的智慧窗戶、啟動熱水器，甚至在半夜關掉所有的燈。這意味著數位世界的代碼漏洞，已經具備了威脅個人實體財產與人身安全的能力。

4. 自動應用程式調用

則聚焦於行動裝置的隱私竊取。在Android系統上，Gemini可以被操控強制開啟特定應用程式。研究人員展示了如何透過AI啟動Zoom並開始視訊串流錄影，或是利用瀏覽器權限在背景悄悄竊取受害者的電子郵件數據與地理定位資訊。這種攻擊將智慧型手機從個人的助手變成了攻擊者的遠端監視器。

(三) 風險評估與防禦的黎明

為了量化這些威脅，研究團隊開發了名為TARA的風險評估框架，專門用於分析LLM驅動的應用程式。在對14個真實攻擊場景的測試中，結果令人戰慄：高達73% 的場景被評為「高」至「關鍵」風險。這顯示出當前的AI整合架構在安全設計上仍有顯著的滯後，過度追求功能的自動化與無縫體驗，卻忽略了跨權限調用時的驗證機制。

針對此項研究，Google展現了負責任的態度並積極回應。在收到研究團隊的責任披露後，Google部署了多層次的防護網。其中最重要的改變是引入了「敏感操作的使用者確認」機制。現在，當Gemini試圖執行涉及修改權限或物理設備的操作時，系統會強制彈出確認視窗，避免AI在背景「自動」完成攻擊者的指令。除此之外，Google也提升了惡意URL的檢測能力，並在後端部署了專門針對提示詞注入的分類器，用以識別潛藏在數據中的惡意指令。這些措施顯著降低了風險等級，也為其他AI開發者提供了寶貴的防禦範例。

(四) 自動化的雙面性：從Apps Script到AI代理

為了更全面地理解這場技術變革，我們必須將視角擴展到正常的功能應用上。在許多教學資料中，我們可以看到Google Apps Script與Gemini的強大結合。開發者可以利用簡單的JavaScript程式碼自動化瑣碎的工作，例如將Sheets的數據自動轉化為PDF並發送郵件，或是透過API讓Gemini自動摘要數百封未讀取郵件。

這種「自動化」的邏輯與「攻擊」的邏輯在技術底層是高度相似的。Apps Script允許使用者建立「觸發器」（Triggers），根據時間或特定事件（如收到新郵件）自動執行任務。

這本是提高生產力的神兵利器，但在缺乏嚴格隔離的環境下，它也為惡意提示詞提供了現成的執行管道。當我們教導使用者如何「讓AI自動讀取日曆並規劃路線」時，我們其實也在無形中開啟了一扇窗，讓AI暴露在外部數據的影響之下。

這種對比揭示了現代AI安全的本質悖論：我們希望AI越聰明、越主動（Agentic），它就必須擁有越多的數據讀取權限與工具執行權限；然而，這權限越多，它被劫持後的破壞力就越大。

參訪心得及建議

參加DEF CON 33不僅是參與一場全球頂尖的資訊安全盛宴，更深度沉浸於海量專業知識與前瞻技術的洗禮。在為期四天的議程中，高密度的技術專題演說、實戰化的網路通訊監聽對抗、以及各類主題的CTF競賽同步舉行。會場內跨領域專家不分晝夜地分享實務經驗，展現了對技術極致追求的熱忱。這不僅是一場資訊技術的交流會議，更見證資安社群透過集體智慧建構安全防禦邊界的具體實踐。

筆者從DEF CON 33體悟到現在資安防禦除了面臨AI科技帶來的新技術與挑戰外，還有利用既有合法的服務作為攻擊媒介的威脅。隨著AI技術的迅速發展，傳統的資安防護設備已不再是攻擊者試圖突破的目標，而是利用AI優化社交工程、壓縮攻擊週期，並讓惡意流量完全隱藏在日常使用的瀏覽器或是Google SecOps等合法服務中；甚至利用自動化工具，加速入侵攻擊者的目標、取得權限並造成危害。

同樣的，防禦端要打破舊有的思維架構，傳統的邊界防護系統已經沒辦法有效抵禦AI及「寄生服務」攻擊，勢必要結合AI監控和自動化等「行為導向」的防禦機制並可快速抵禦攻擊行為。除此之外，應落實端點治理與認證強化：針對瀏覽器實施嚴格的擴充程式「白名單管理」以防範資訊外洩；並執行強化的身分存取管理（IAM），要求密碼符合複雜度規範與定期更換，結合多因素驗證建構全方位的零信任防禦體系。

總而言之，資安攻防是一場永無止境的進化賽跑，面對AI與寄生式攻擊的雙重挑戰，防禦端已無退路，必須從被動修補轉向主動防禦。從DEF CON 33的交流中我們看到，技術的巔峰不在於單一漏洞的攻克，而在於如何運用自動化與行為監測構建全面防線。唯有落實嚴密的治理策略，才能在科技高度發達的今日，守住資訊安全的最後堡壘。