PowerShell介紹

北區ASOC 二線工程師 林宜進
E-mail：tjline01@asoc.cc.ntu.edu.tw

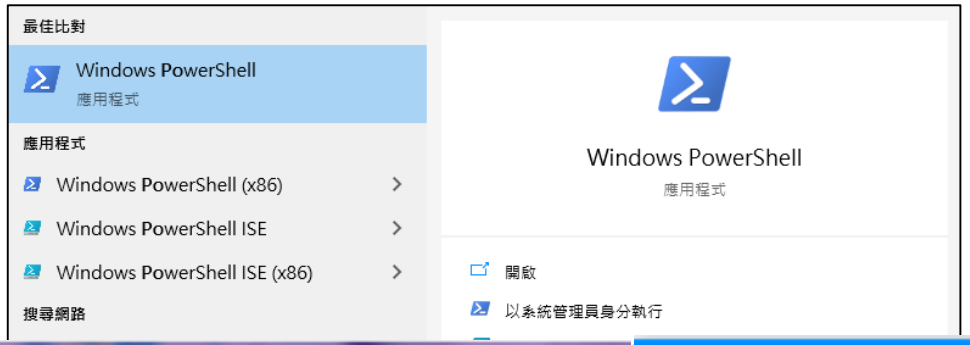
大綱

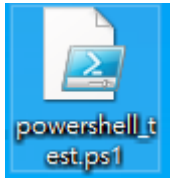
- 前言
- 偵測說明
- 案例分析
- DEMO
- 補充內容
- 參考資料

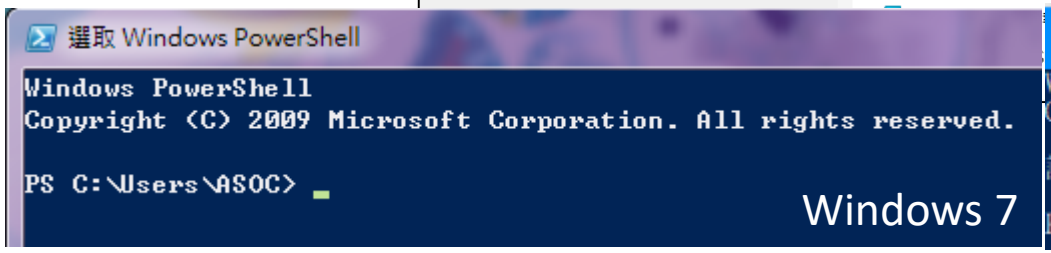
前言

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)

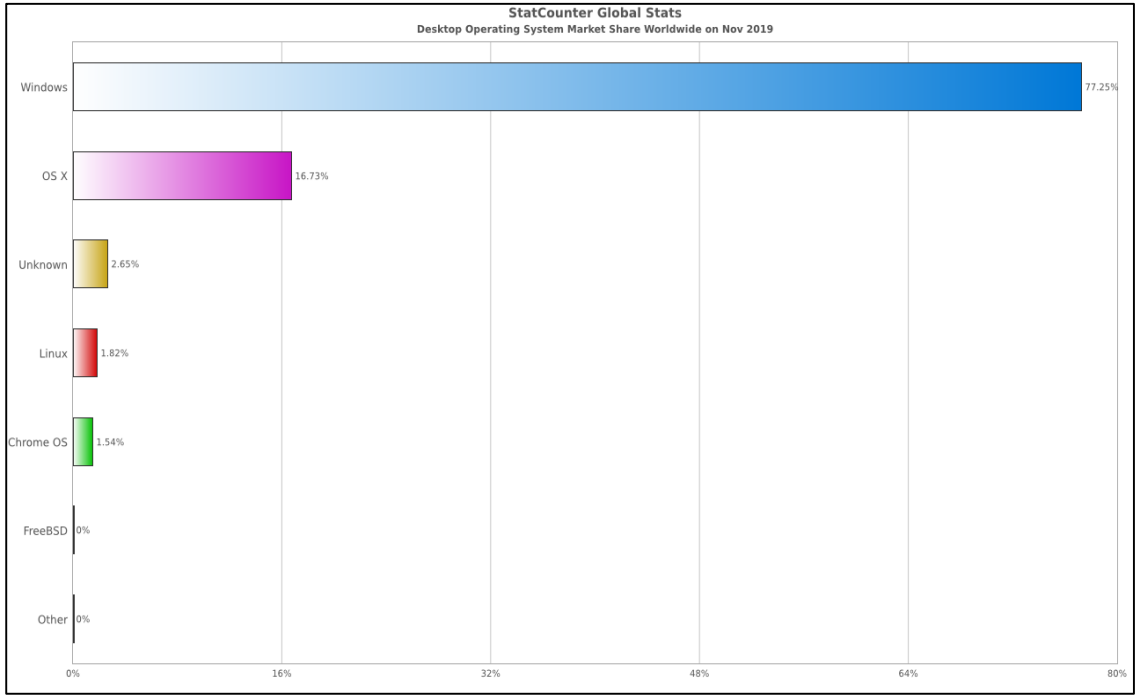






為何用PowerShell攻擊?

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

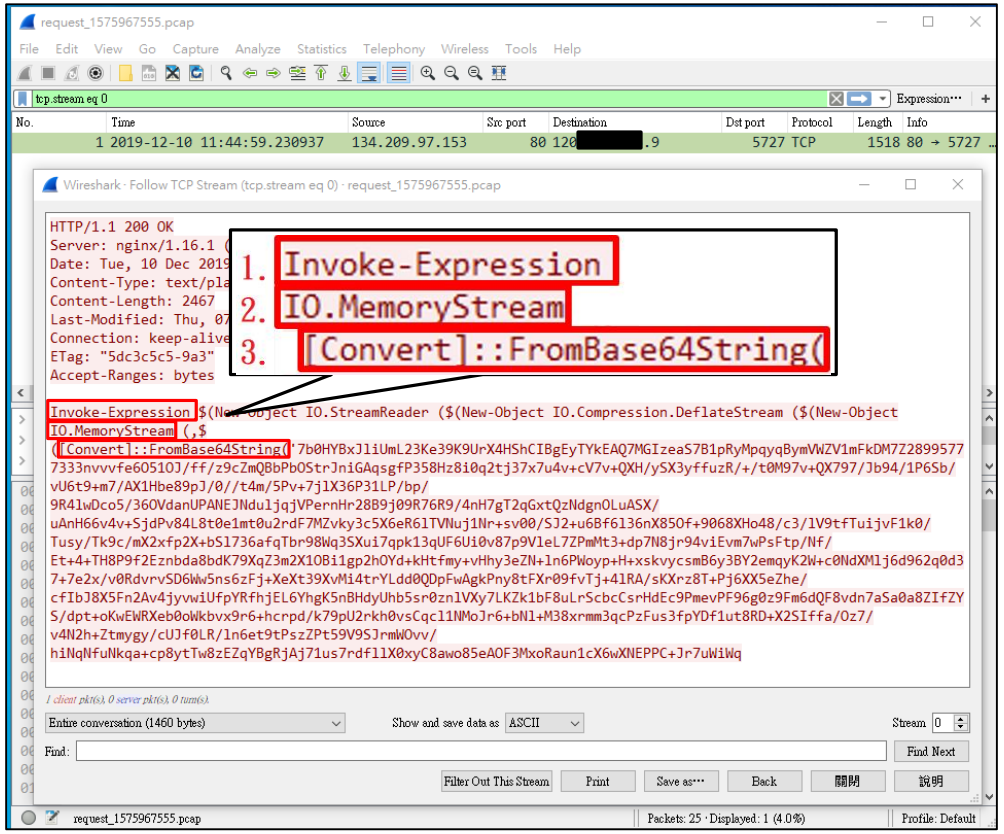
偵測說明

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)

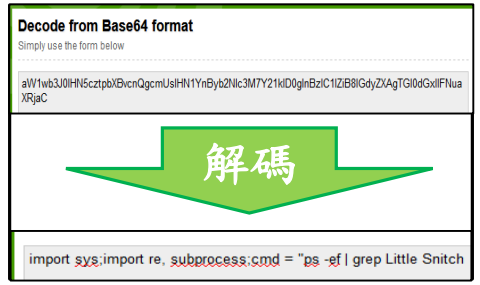
封包內容包含特定關鍵字(http標頭相關)

傳送檔案內容包含特定關鍵字(記憶體位置)

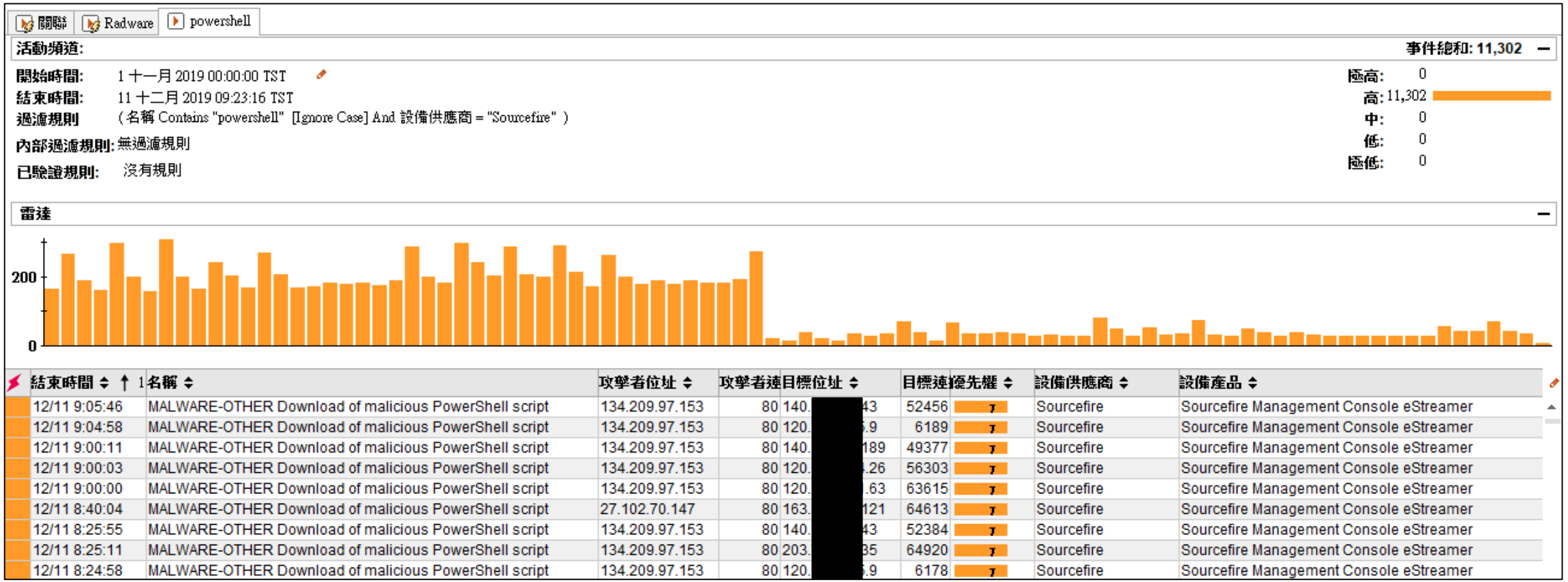
傳送檔案時包含“bypass”字串

傳送檔案時包含PowerSell指令

封包內容包含特定關鍵字(http標頭相關)



ArcSight紀錄



關聯 | Radware | powershell

活動頻道: 事件總和: 11,302

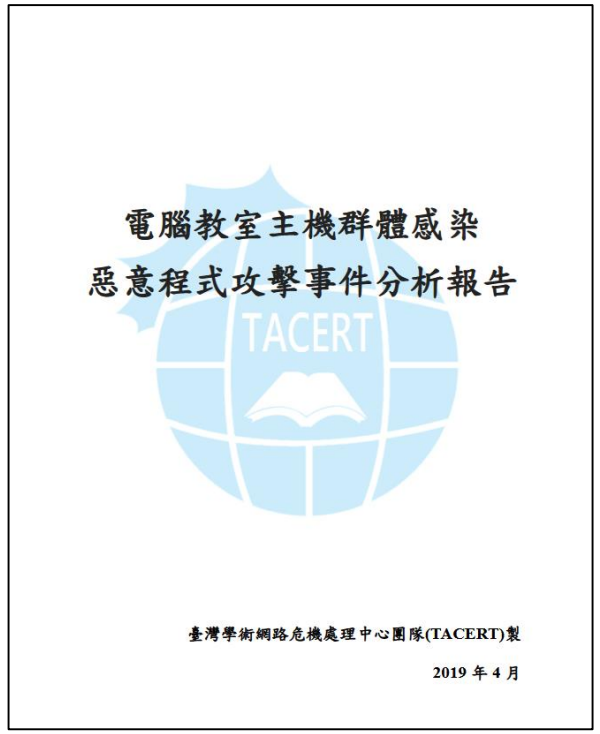
開始時間: 1 十一月 2019 00:00:00 TST
結束時間: 11 十二月 2019 09:23:16 TST
過濾規則 (名稱 Contains "powershell" [Ignore Case] And 設備供應商 = "Sourcefire")
內部過濾規則: 無過濾規則
已驗證規則: 沒有規則

極高: 0
高: 11,302
中: 0
低: 0
極低: 0

雷達

結束時間	名稱	攻擊者位址	攻擊者連	目標位址	目標連	優先權	設備供應商	設備產品
12/11 9:05:46	MALWARE-OTHER Download of malicious PowerShell script	134.209.97.153	80	140.[redacted]43	52456	7	Sourcefire	Sourcefire Management Console eStreamer
12/11 9:04:58	MALWARE-OTHER Download of malicious PowerShell script	134.209.97.153	80	120.[redacted].9	6189	7	Sourcefire	Sourcefire Management Console eStreamer
12/11 9:00:11	MALWARE-OTHER Download of malicious PowerShell script	134.209.97.153	80	140.[redacted]189	49377	7	Sourcefire	Sourcefire Management Console eStreamer
12/11 9:00:03	MALWARE-OTHER Download of malicious PowerShell script	134.209.97.153	80	120.[redacted].26	56303	7	Sourcefire	Sourcefire Management Console eStreamer
12/11 9:00:00	MALWARE-OTHER Download of malicious PowerShell script	134.209.97.153	80	120.[redacted].63	63615	7	Sourcefire	Sourcefire Management Console eStreamer
12/11 8:40:04	MALWARE-OTHER Download of malicious PowerShell script	27.102.70.147	80	163.[redacted]121	64613	7	Sourcefire	Sourcefire Management Console eStreamer
12/11 8:25:55	MALWARE-OTHER Download of malicious PowerShell script	134.209.97.153	80	140.[redacted]43	52384	7	Sourcefire	Sourcefire Management Console eStreamer
12/11 8:25:11	MALWARE-OTHER Download of malicious PowerShell script	134.209.97.153	80	203.[redacted]35	64920	7	Sourcefire	Sourcefire Management Console eStreamer
12/11 8:24:58	MALWARE-OTHER Download of malicious PowerShell script	134.209.97.153	80	120.[redacted].9	6178	7	Sourcefire	Sourcefire Management Console eStreamer

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

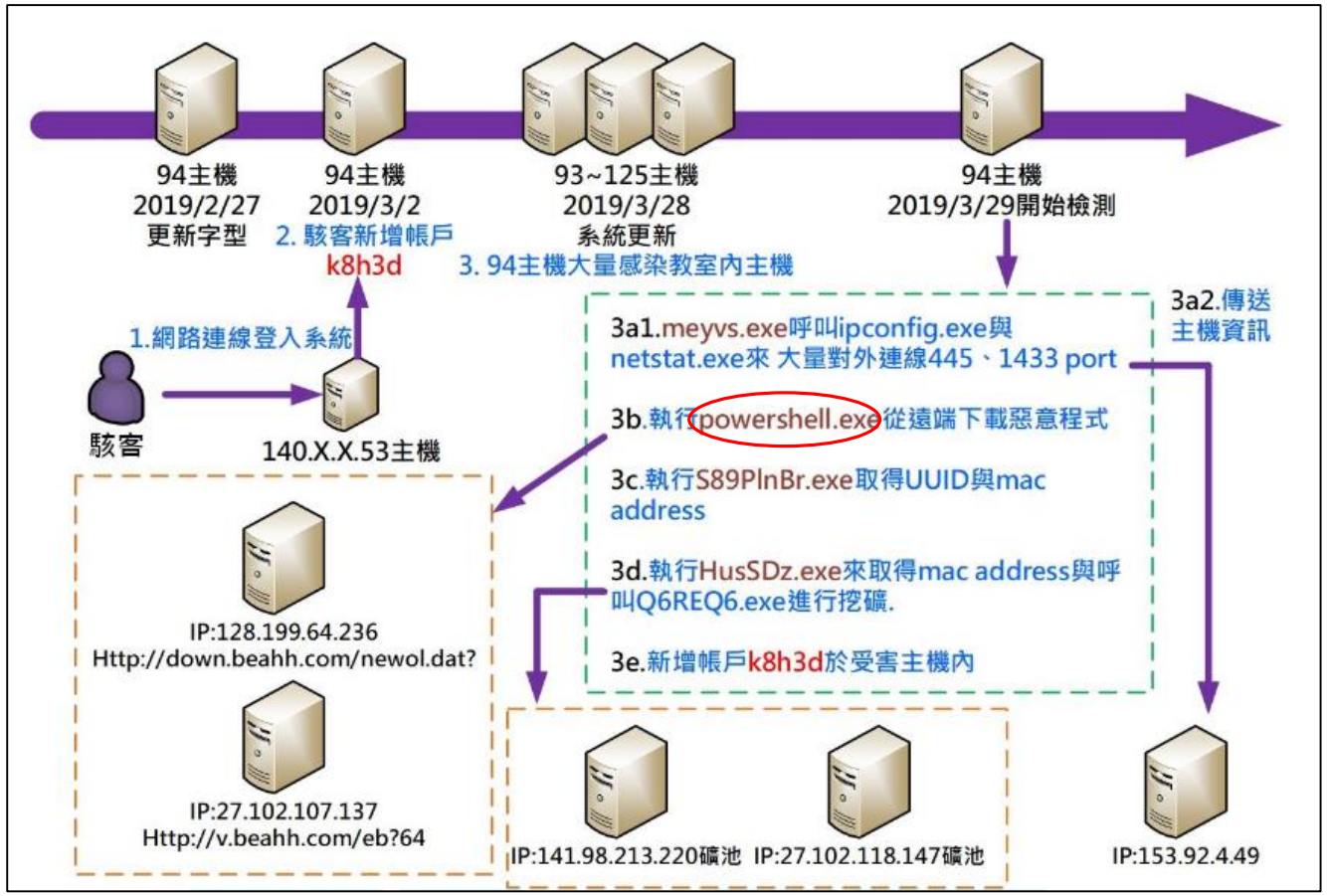
圖片來源：https://fortiguard.com/encyclopedia/ips/32978/malicious-file-downloading

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

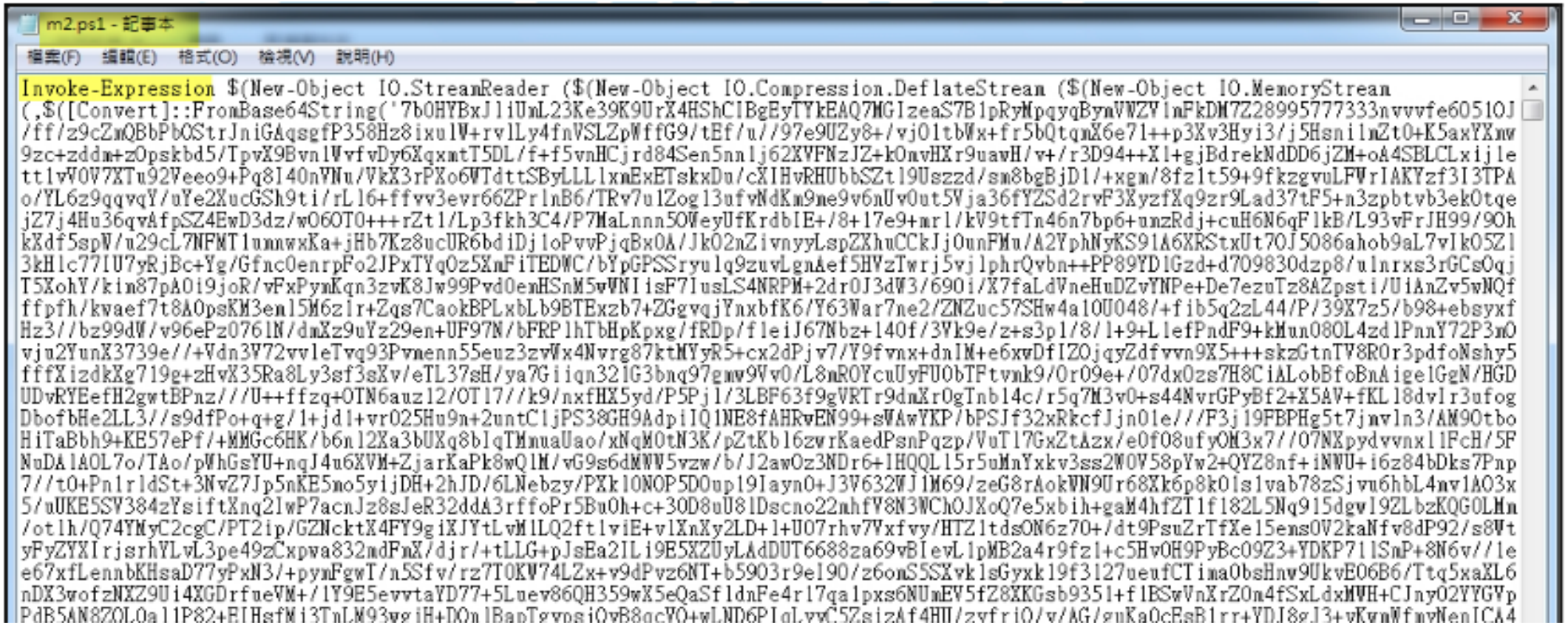
圖片來源：https://portal.cert.tanet.edu.tw/docs/pdf/20190425010432325668479617618 43.pdf

時間軸



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

PowerShell指令



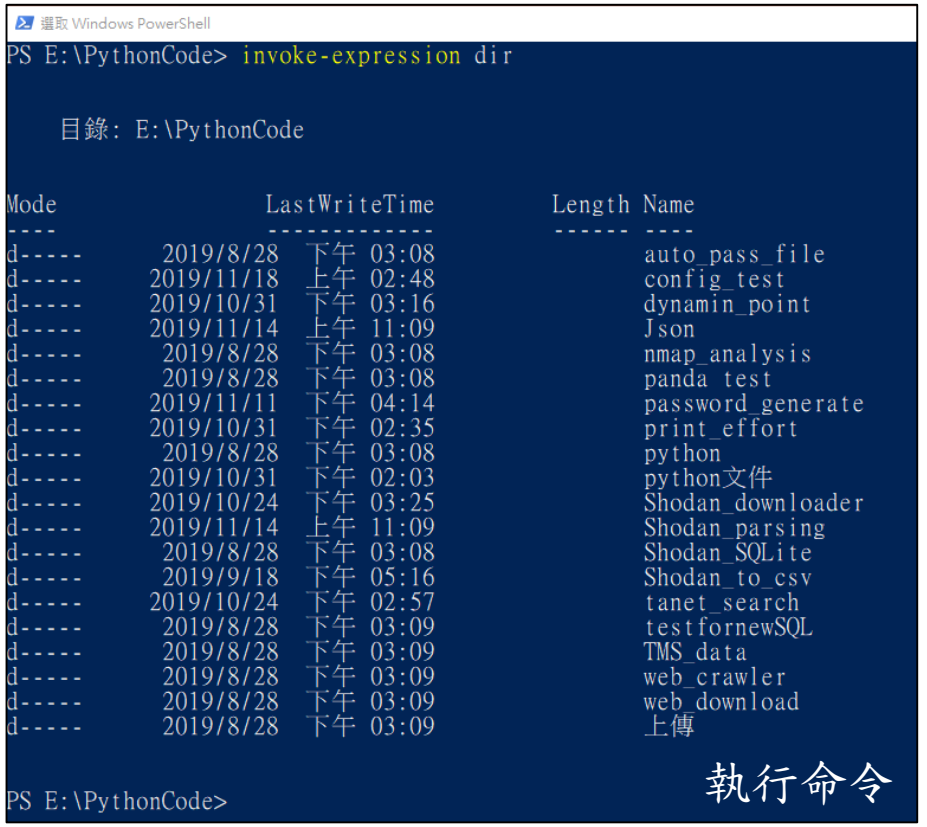
圖片來源：https://portal.cert.tanet.edu.tw/docs/pdf/2019042501043232566847961761843.pdf

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

◆ Example

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



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

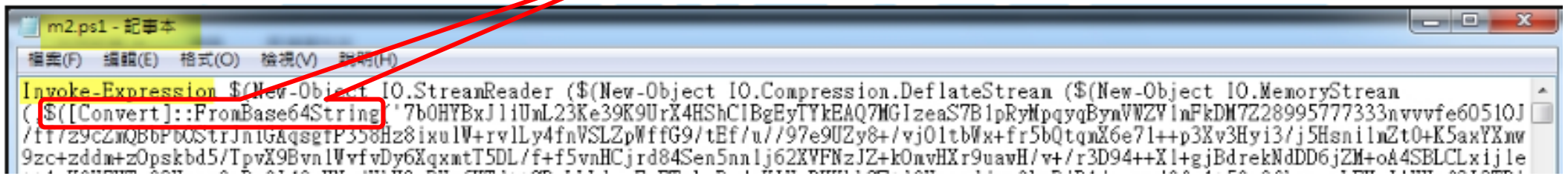
建議措施

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



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

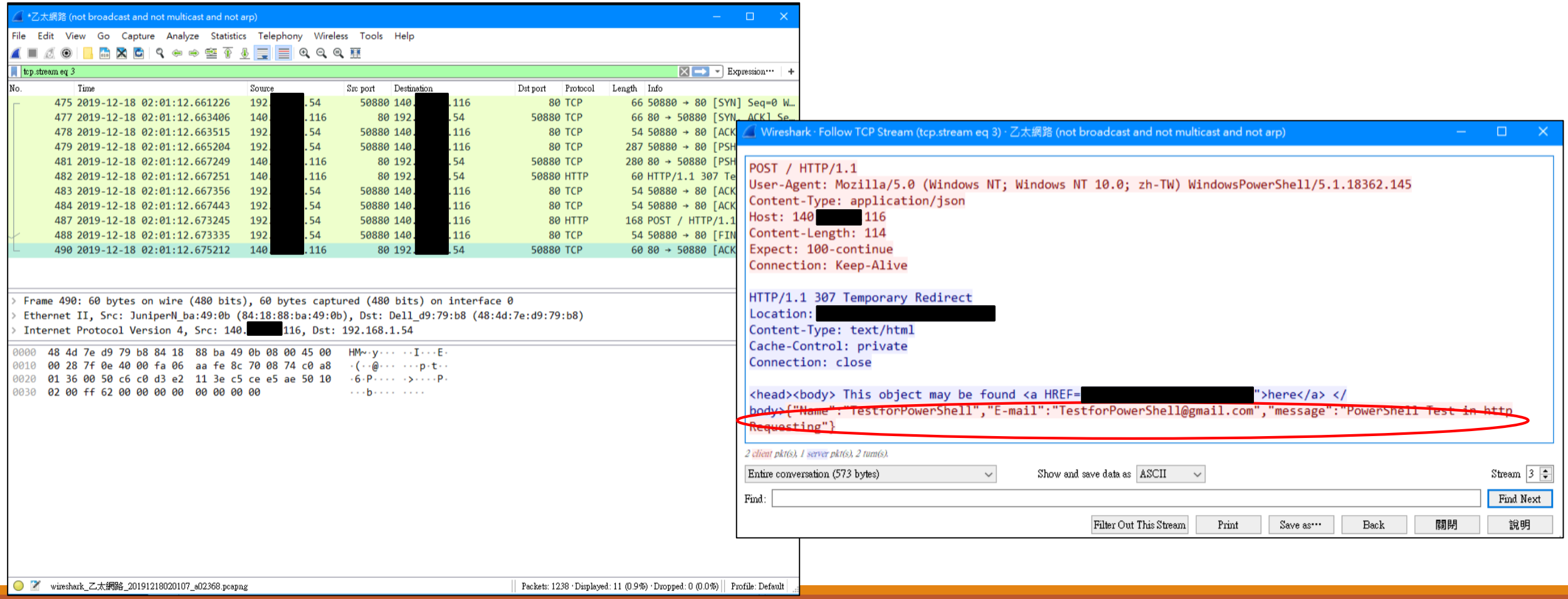
DEMO1：發送 HTTP Requesting

- 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
```

DEMO1：封包內容

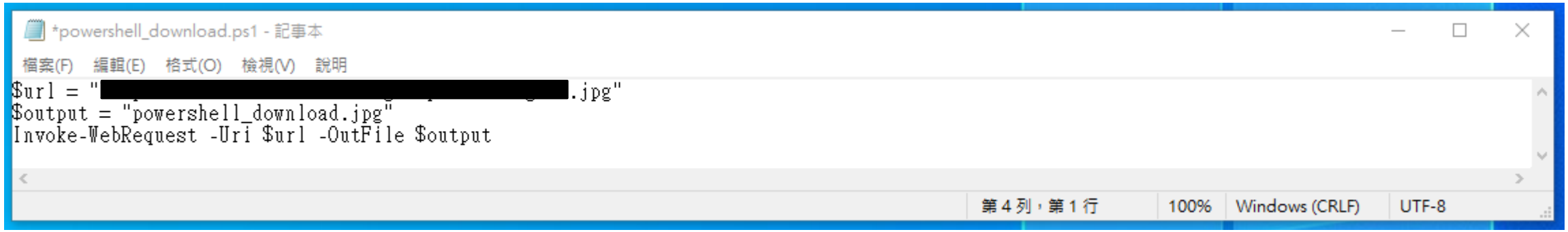

*乙太網路 (not broadcast and not multicast and not arp)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp.stream eq 3
No. Time Source Src port Destination Dst port Protocol Length Info
475 2019-12-18 02:01:12.661226 192.[redacted].54 50880 140.[redacted].116 80 TCP 66 50880 → 80 [SYN] Seq=0 W…
477 2019-12-18 02:01:12.663406 140.[redacted].116 80 192.[redacted].54 50880 TCP 66 80 → 50880 [SYN, ACK] Se…
478 2019-12-18 02:01:12.663515 192.[redacted].54 50880 140.[redacted].116 80 TCP 54 50880 → 80 [ACK
479 2019-12-18 02:01:12.665204 192.[redacted].54 50880 140.[redacted].116 80 TCP 287 50880 → 80 [PSH
481 2019-12-18 02:01:12.667249 140.[redacted].116 80 192.[redacted].54 50880 TCP 280 80 → 50880 [PSH
482 2019-12-18 02:01:12.667251 140.[redacted].116 80 192.[redacted].54 50880 HTTP 60 HTTP/1.1 307 Te
483 2019-12-18 02:01:12.667356 192.[redacted].54 50880 140.[redacted].116 80 TCP 54 50880 → 80 [ACK
484 2019-12-18 02:01:12.667443 192.[redacted].54 50880 140.[redacted].116 80 TCP 54 50880 → 80 [ACK
487 2019-12-18 02:01:12.673245 192.[redacted].54 50880 140.[redacted].116 80 HTTP 168 POST / HTTP/1.1
488 2019-12-18 02:01:12.673335 192.[redacted].54 50880 140.[redacted].116 80 TCP 54 50880 → 80 [FIN
490 2019-12-18 02:01:12.675212 140.[redacted].116 80 192.[redacted].54 50880 TCP 60 80 → 50880 [ACK
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.[redacted]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 ···b···· ····
wireshark_乙太網路_20191218020107_a02368.pcapng
Packets: 1238 · Displayed: 11 (0.9%) · Dropped: 0 (0.0%)
Profile: Default
Wireshark · Follow TCP Stream (tcp.stream eq 3) · 乙太網路 (not broadcast and not multicast and not arp)
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[redacted]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 <a HREF=[redacted]">here </
body>{"Name":"TestforPowerShell","E-mail":"TestforPowerShell@gmail.com","message":"PowerShell Test in http
Requesting"}
2 client pkt(s), 1 server pkt(s), 2 turn(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

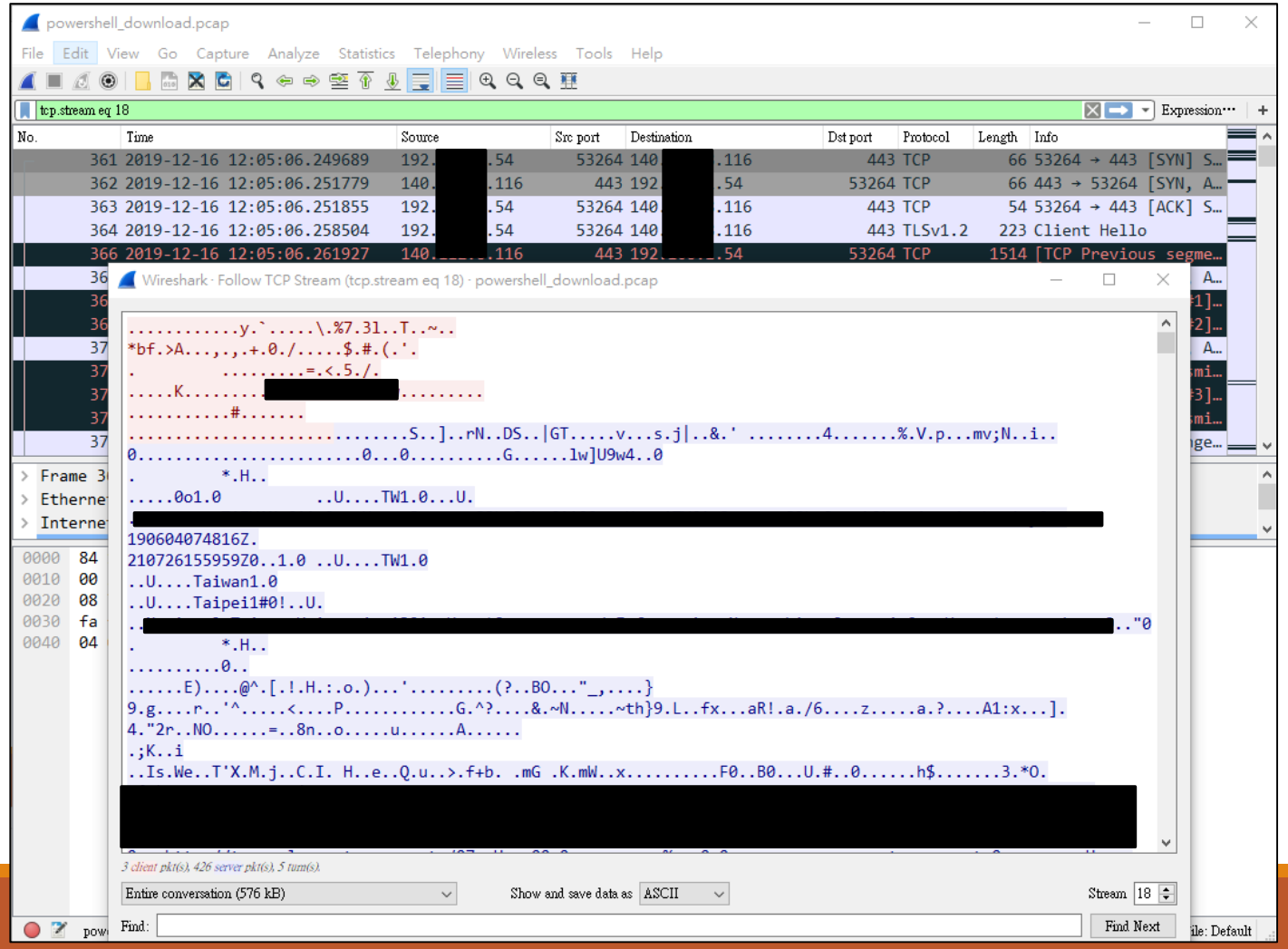
- 此次DEMO是透過PowerShell下載圖片

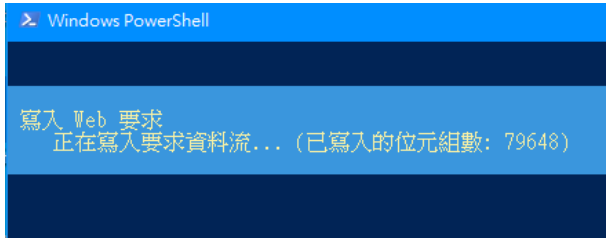
◆ Example

```
$url = "任意圖片的url"
$output = "powershell_download.jpg"
Invoke-WebRequest -Uri $url -OutFile $output
```



DEMO2：執行後的結果和封包內容





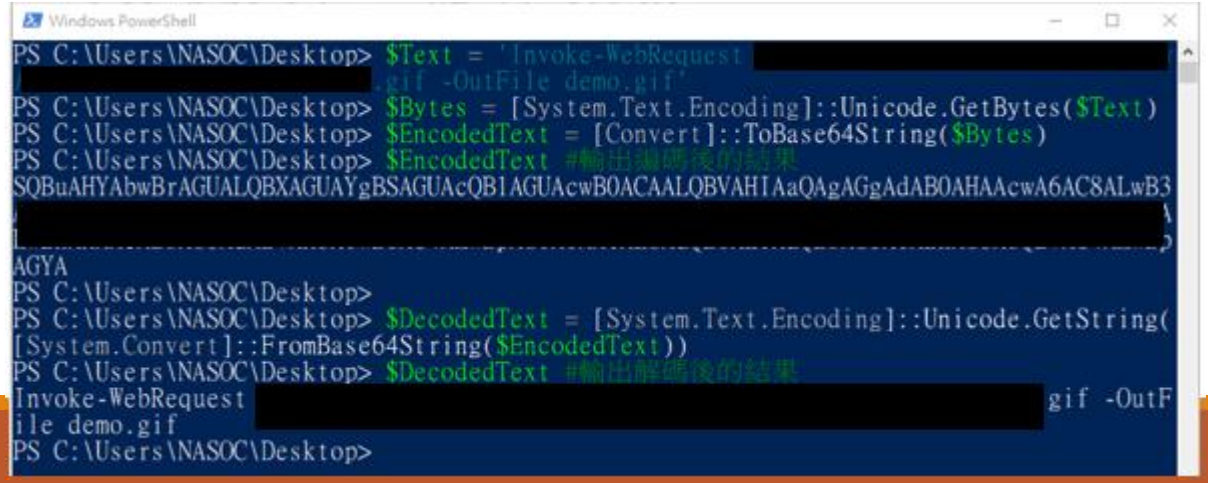
DEMO3：指令加密與解密

- 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*



補充內容

通用指令

PowerShell內可用的cmd.exe和UNIX指令如下圖：

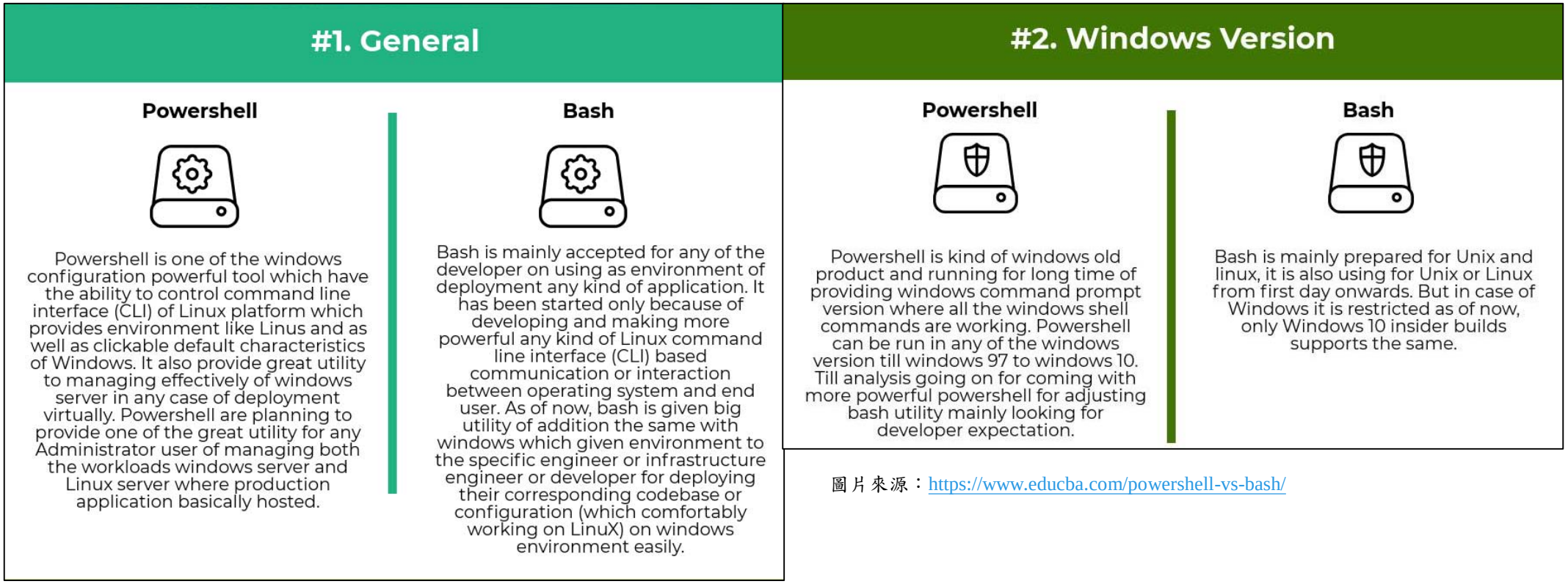
cat	dir	mount	rm
cd	echo	move	rmdir
chdir	erase	popd	sleep
clear	h	ps	sort
cls	history	pushd	tee
copy	kill	pwd	type
del	lp	r	write
diff	ls	ren	

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

PowerShell與Bash的比較



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/201904250104323256684796176184 3.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