Travnet 殭屍網路分析簡介(MacAfee)

原文出處：McAfee (CERT 譯)

http://blogs.mcafee.com/mcafee-labs/travnet-botnet-steals-huge-amount-of-sensitive-data

Vikas Taneja 上個月討論了惡意軟體 Travnet，從那時我們持續分析不同的樣本，因為控制代碼的出現，可以確定 Travnet 是殭屍網路而不是個木馬，此惡意軟體有能力等待惡意服務器的進一步指示。

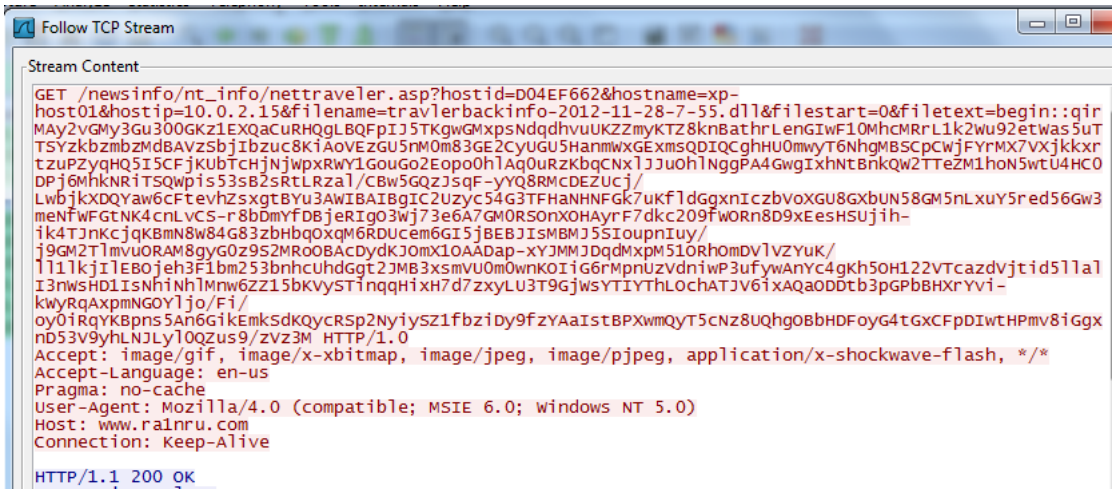
Travnet 殭屍網路不只從受害機器竊取敏感性資料也偷取文件檔案。一般來說，我們將敏感性資料儲存在 Office 檔案、PDF 等。Travnet 使用資料壓縮與資料加密的方法來偷取資料其中也包含大檔案。

Bot 一開始會收集受害者的敏感性資料，接下來搜尋文件檔案 (doc.docx.xls.xlsx.txt.rtf.pdf)。以下是片斷代碼：

```
.text:10001E9F          push    offset unk_10016504 ; name
.text:10001EA4          call    GET_IP_ADDRESS
.text:10001EA9          push    esi             ; Size
.text:10001EAA          push    offset unk_100163F0 ; Dest
.text:10001EAF          call    GET_VOLUME_INFORMATION
.text:10001EB4          mov     edi, dword_10016B6C
.text:10001EBA          add     esp, 14h
.text:10001EBD          lea     eax, [ebp+Buffer]
.text:10001EC3          imul    edi, 0EA60h
.text:10001EC9          push    esi             ; uSize
.text:10001ECA          push    eax             ; lpBuffer
.text:10001ECB          xor     ebx, ebx
.text:10001ECD          call    ds:GetSystemDirectoryA
.text:10001ED3          mov     esi, ds:sprintf
.text:10001ED9          lea     eax, [ebp+Buffer]
.text:10001EDF          push    eax
.text:10001EE0          lea     eax, [ebp+FileName]
.text:10001EE6          push    offset aSSystem_t_dll ; "%s\\system_t.dll"
.text:10001EEB          push    eax             ; Dest
.text:10001EEC          call    esi ; sprintf
.text:10001EEE          add     esp, 0Ch
.text:10001EF1          call    FIND_COMPUTER_INFORMATION
.text:10001EF6          call    FIND_RUNNING_PROCESSES
.text:10001EFB          call    FIND_IP_CONFIG_DETAILS
.text:10001F00          lea     eax, [ebp+SystemTime]
.text:10001F03          push    eax             ; lpSystemTime
.text:10001F04          call    ds:GetLocalTime
.text:10001F0A          movzx   eax, [ebp+SystemTime.wMinute]
.text:10001F0E          push    eax
.text:10001F0F          movzx   eax, [ebp+SystemTime.wHour]
.text:10001F13          push    eax
.text:10001F14          movzx   eax, [ebp+SystemTime.wDay]
.text:10001F18          push    eax
.text:10001F19          movzx   eax, [ebp+SystemTime.wMonth]
.text:10001F1D          push    eax
.text:10001F1E          movzx   eax, [ebp+SystemTime.wYear]
.text:10001F22          push    eax
.text:10001F23          lea     eax, [ebp+var_21C]
.text:10001F29          push    offset aTravlerbackinf ; "travlerbackinfo-%d-%d-%d-%d-%d.dll"
.text:10001F2E          push    eax             ; Dest
.text:10001F2F          call    esi ; sprintf
.text:10001F31          call    CREATE_CONFIG_DAT_FILE
.text:10001F36          lea     eax, [ebp+var_21C]
.text:10001F3C          push    eax             ; int
.text:10001F3D          lea     eax, [ebp+FileName]
.text:10001F43          push    eax             ; lpFileName
.text:10001F44          call    START_SEND_INFO
.text:10001F49          add     esp, 24h
```

先前的代碼包含了電腦名稱、IP 位址、使用者名稱、作業系統、運行程序清單、IP 設定細節與系統上目前的帳戶。惡意軟體建立檔案 system_t.dll 來儲存純文字。同時也建立檔案 travelbackinfo-(SystemTime).dll，此檔案使用在 HTTP GET 要求中。

存在檔案的資料可以很大，全憑執行的程序與 IP 設定細節。殭屍網路將使用資料壓縮與加密方法來將資料傳送至遠端服務器。封包記錄看起來像是：



Bot 用參數"&filetext" 由"begin::."開始來送出竊取的資料，但是壓縮的檔案可能會太大而不能透過 HTTP 傳送，所以會以 1024bytes 的大小傳送。為了要追蹤資料，它透過參數"&filestart."夾帶著字串"::end"來暗示檔案結束。

資料壓縮與加密技術

Bot 用兩個動作來處理原始資料：

- 第一步，使用像是 LZSS(Lempel–Ziv–Storer–Szymanski)壓縮方法來壓縮原始資料。
- 第二步，使用自製的 Base64 加密壓縮的檔案。

第一步資料壓縮

資料壓縮保持先前看見資料的 sliding window 相似 LZSS 的資料壓縮。

透過相似的方法保持 sliding window 大小(來實現高壓縮率)但是輸出變數長度"長度偏移"配對(用來呈現教字 bits 的數量)。我們還不曾見過任何引用或實施輸出變數長度與變數偏移，目前我們將稱呼此為 LZSS 資料壓縮演算法的一種變種。

Bot 透過讀入 65536bytes 的大小來壓縮(如此必須保持 sliding window 的大小)，最後的壓縮輸出將會呈現如下格式：

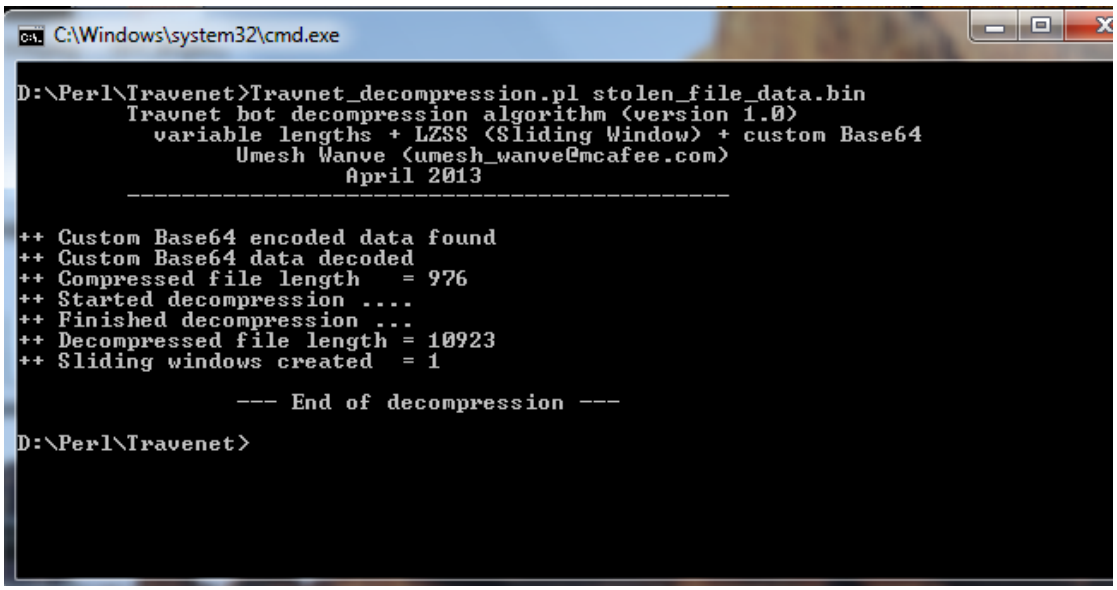
Original Length (2 bytes) + Compressed Length (2 bytes) + Compressed Data

此方法實現了高度壓縮率並減少了原始資料的大小，允許 bot 上傳檔案到遠端伺服器。解壓程序則非常簡單，因為不需要搜尋最長配對但只需要注意變數長度值。

第二步 自製的 Base64 加密

Travnet 使用自製的 Base64 加密來對壓縮的二元資料加密。使用在標準的 Base64 的金鑰與字集為 “ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/” 並用 “=” 來填充；bot 用的則是 ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789-/” 並用 “*” 來填充。

我們寫了一個小工具來解壓縮 Travnet 竊取的資料。



看著輸出，我們看見解壓檔案(原始資料)的大小高出壓縮的檔案許多。現在看看解壓的資料：

```
[¼ÆËã»úÐÅÏ¢]
¼ÆËã»ú: XP-HOST01
ÓÃ»§Ãû: SYSTEM
IpµØÖ·: [redacted]
²Ù×÷ÏµÍ³: Microsoft Windows XP Professional, Service Pack 2 (Build 2600)

´ÅÅÌ¿Õ¼ä: ×Ü´ÅÅÌ¿Õ¼äÎª:9GB,Ê£Óà´ÅÅÌ¿Õ¼äÎª5GB(Õ¼55.56%)
CPU: GenuineIntel x86 Family 6 Model 47 Stepping 2 2251MHZ
ÎïÀíÄÚ´æ: ×ÜÎïÀíÄÚ´æ:1023MB,¿ÉÓÃÄÚ´æ:812MB (Õ¼79.40%)

[½ø³ÌÁÐ±í]
   0 [System Process]
   4 System
 416 smss.exe
 472 csrss.exe
 496 winlogon.exe
 540 services.exe
 552 lsass.exe
 700 vservice.exe
 740 svchost.exe
 824 svchost.exe
 892 svchost.exe
 932 svchost.exe
 960 svchost.exe
1280 explorer.exe
1376 spoolsv.exe
1536 jqs.exe
1956 IEXPLORE.EXE
1992 clicker.exe
 332 alg.exe
1148 python.exe
```

上面的是受害者被竊取的檔案，有趣的是無法辨識的字是中文。當將敏感性資料寫入到 DLL 檔案，Bot 作者寫入的硬碼是中文。如果我們轉換成英文，看起來像是：

```
[Computer Information]
Computer: XP-HOST01
User name: SYSTEM
IP Address: [redacted]
Operating System: Microsoft Windows XP Professional, Service Pack 2 (Build 2600)

Disk Space: Total disk space: 9GB free disk space to 5GB (55.56%)
CPU: GenuineIntel x86 Family 6 Model 47 Stepping 2 2251MHZ
Physical Memory: Total Physical Memory: 1023MB, available memory: 812MB (79.40%)

[List of processes]
   0 [System Process]
```

竊取檔案

Bot 不會停止；並偷的更多。接下來我們看看 bot 呼叫的功能：

```
.text:10001FB0                                          ; StartAddress+F3j ...
.text:10001FB0                  call    FIND_ALL_FILES
.text:10001FB5                  call    SEND_FILES_LIST
.text:10001FBA                  test    eax, eax
.text:10001FBC                  jz      short loc_10001FC9
.text:10001FBE                  push    offset FileName ; lpFileName
.text:10001FC3                  call    ds:DeleteFileA
.text:10001FC9
.text:10001FC9 loc_10001FC9:                            ; CODE XREF: StartAddress+134j
.text:10001FC9                  call    FIND_DOCUMENTS
.text:10001FCE                  call    SEND_DOCUMENTS_FILE
.text:10001FD3                  test    eax, eax
.text:10001FD5                  jz      short loc_10001FE2
.text:10001FD7                  push    offset byte_100161A0 ; lpFileName
.text:10001FDC                  call    ds:DeleteFileA
.text:10001FE2
.text:10001FE2 loc_10001FE2:                            ; CODE XREF: StartAddress+14Dj
.text:10001FE2                  call    FIND_DESKTOP_FILES
```

Bot 會傳出以下資訊：

- 含有系統所有檔案名稱的清單
- 為 doc、docx、xls、xlsx、txt、rtf 與 pdf 的檔案。
- 所有來自受害電腦的檔案。

一旦它送出所有檔案到遠端服務器，Bot 將沉睡並等待進一步指令。

服務器指令

- UNINSTALL
- UPDATE
- RESET
- UPLOAD

接下來，我們看到服務器要求上傳更多檔案的指令：

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 50
Content-Type: text/html
Server: Microsoft-IIS/7.0
X-Powered-By: ASP.NET
Date: Sun, 21 Apr 2013 13:51:33 GMT

78D71C31_Success:UPLOAD
2878D244_Success:UPLOAD
```

雖然殭屍網路使用簡單的手法感染與竊取資訊，一些原因讓 Travnet 殭屍網路變得特別：

- 使用無損資料壓縮竊取大資料檔案。
- 竊取副檔名為.doc,docx,xls,xlsx,txt,rtf 與 pdf 的檔案。
- 竊取系統上的所有檔案。

這些特別的特徵與中文字的呈現，讓我們有了以下的結論。Travnet 殭屍網路可能以敏感性資料為目標。我們懷疑攻擊者使用初始數據–電腦資訊、IP 等來竊取特定團體或個體的資料。我們也相信資料上傳到惡意伺服器受到攻擊者的監控。我們已經發現新的用來執行攻擊已註冊的網域。我們相信受 Travnet 感染受害者有大量的資料被竊取。