باسمه تعالی

تحلیل فنی باج‌افزار

Nefilim

تاریخ نگارش :

۱۳۹۹/۰۱/۲۷

فهرست مطالب

۱. مقدمه :

در اواخر ماه فوریه سال ۲۰۲۰ میلادی، اخباری مبنی بر فعالیت باج‌افزاری به‌نام Nefilim درفضای سایبری منتشر شد. طبق شواهد موجود، کد این باج‌افزار با باج‌افزار Nemty شباهت بسیار زیادی دارد. این باج‌افزار پس از اتمام فرآیند رمزگذاری، در صورت نپرداختن مبلغ باج، قربانی را تهدید به انتشار فایل‌های رمزگذاری شده می‌کند. باج‌افزار Nefilim از الگوریتم رمزنگاری AES-128 برای رمزگذاری فایل‌ها استفاده کرده است.

۲. مشخصات فایل اجرایی آنپک شده:

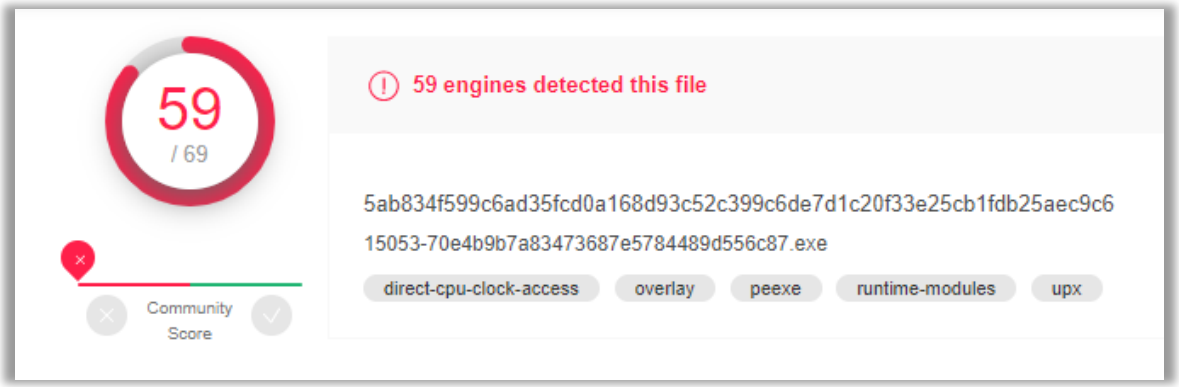
نام فایل	ransom
MD5	70e4b9b7a83473687e5784489d556c87
SHA-1	1f594456d88591d3a88e1cdd4e93c6c4e59b746c
SHA-256	5ab834f599c6ad35fcd0a168d93c52c399c6de7d1c20f33e25cb1fdb25aec9c6
نوع فایل	Win32 EXE
اندازه فایل	36 کیلوبایت

۳. شجره‌نامه

براساس گزارش‌های منتشر شده، باج‌افزار Nefilim ظاهراً نسخه‌ای توسعه یافته از باج‌افزار Nemty 2.5 می‌باشد. مهمترین تفاوت این دو باج‌افزار در این است که Nefilim از شیوه RaaS برای انتشار خود استفاده نمی‌کند.

٤. میزان تهدید فایل باج‌افزار

درحال حاضر تعداد ۵۹ مورد از ٦۹ ضدبدافزار سامانه VirusTotal، قادر به شناسایی، توقف و یا حذف این باج‌افزار می‌باشند.



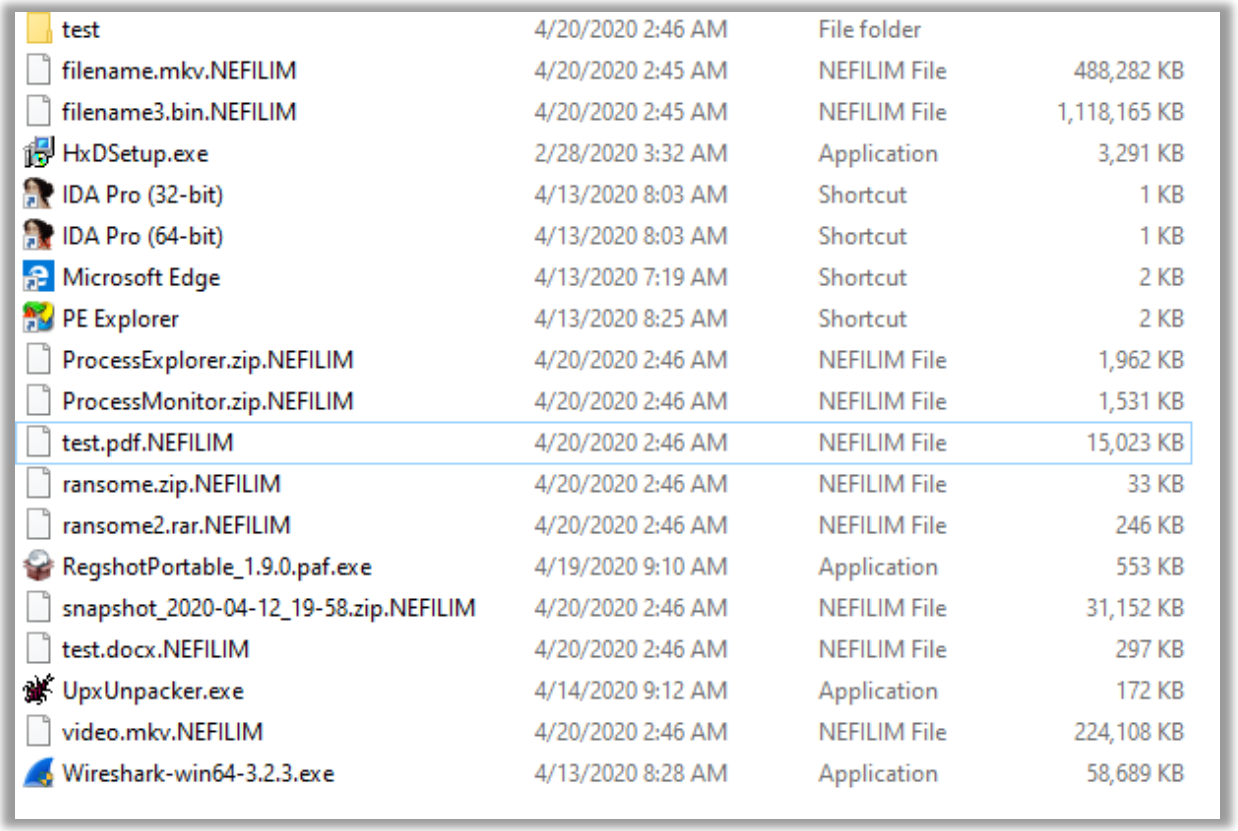
شکل ۱ – نرخ شناسایی باج‌افزار در سامانه VT

۵. تحلیل پویا

۵-۱ آناتومی حمله:

باج‌افزار Nefilim در مدت زمان کوتاهی پس اجرا و شناسایی فایل‌ها و پوشه‌های سیستم قربانی، شروع به رمزگذاری آن‌ها می‌کند. نسخه بررسی شده این باج‌افزار در محیط آزمایشگاهی هیچ‌گونه پیغام باج‌خواهی در پوشه فایل‌های رمزشده ایجاد نمی‌کند اما طبق گزارشات موجود، در برخی از نسخه‌ها، باج‌افزار فایلی به نام NEFILIM-DECRYPT.txt ایجاد می‌کند که حاوی اطلاعاتی برای ارتباط می‌باشد.

سپس فرآیند رمزگذاری شروع شده و باج‌افزار در مدت زمان کوتاهی تمام فایل‌های مورد نظر خود در سیستم قربانی را رمزگذاری می‌نماید. همانطور که در تصویر زیر قابل مشاهده است، تقریبا تمام انواع فایل‌ها (به جز فایل‌های exe) رمزگذاری و به انتهای آن‌ها پسوند NEFILIM. اضافه شده است.



test	4/20/2020 2:46 AM	File folder	
filename.mkv.NEFILIM	4/20/2020 2:45 AM	NEFILIM File	488,282 KB
filename3.bin.NEFILIM	4/20/2020 2:45 AM	NEFILIM File	1,118,165 KB
HxDSetup.exe	2/28/2020 3:32 AM	Application	3,291 KB
IDA Pro (32-bit)	4/13/2020 8:03 AM	Shortcut	1 KB
IDA Pro (64-bit)	4/13/2020 8:03 AM	Shortcut	1 KB
Microsoft Edge	4/13/2020 7:19 AM	Shortcut	2 KB
PE Explorer	4/13/2020 8:25 AM	Shortcut	2 KB
ProcessExplorer.zip.NEFILIM	4/20/2020 2:46 AM	NEFILIM File	1,962 KB
ProcessMonitor.zip.NEFILIM	4/20/2020 2:46 AM	NEFILIM File	1,531 KB
test.pdf.NEFILIM	4/20/2020 2:46 AM	NEFILIM File	15,023 KB
ransome.zip.NEFILIM	4/20/2020 2:46 AM	NEFILIM File	33 KB
ransome2.rar.NEFILIM	4/20/2020 2:46 AM	NEFILIM File	246 KB
RegshotPortable_1.9.0.paf.exe	4/19/2020 9:10 AM	Application	553 KB
snapshot_2020-04-12_19-58.zip.NEFILIM	4/20/2020 2:46 AM	NEFILIM File	31,152 KB
test.docx.NEFILIM	4/20/2020 2:46 AM	NEFILIM File	297 KB
UpxUnpacker.exe	4/14/2020 9:12 AM	Application	172 KB
video.mkv.NEFILIM	4/20/2020 2:46 AM	NEFILIM File	224,108 KB
Wireshark-win64-3.2.3.exe	4/13/2020 8:28 AM	Application	58,689 KB

شکل ۲ – نمونه فایل‌های رمزگذاری شده توسط باج‌افزار Nefilim

باج‌افزار Nefilim پس از اجرا، فایل اجرایی خود را پاک می‌کند و به پروسه خود خاتمه می‌دهد.

۵-۲ روش انتشار:

براساس گزارش TrendMicro، احتمال نفوذ باج‌افزار Nefilim از طریق پروتکل RDP بسیار زیاد است.

۵-۳ روش جلوگیری:

با توجه به روش انتشار باج‌افزار توصیه می‌گردد در صورت استفاده از سرویس دسترسی از راه دور، علاوه بر مانیتور کردن لاگین‌های کاربران در ساعات مختلف شبانه‌روز، اقدامات مربوط به امن‌سازی RDP حتماً در دستور کار قرار گیرد.

٦. تحلیل ایستا

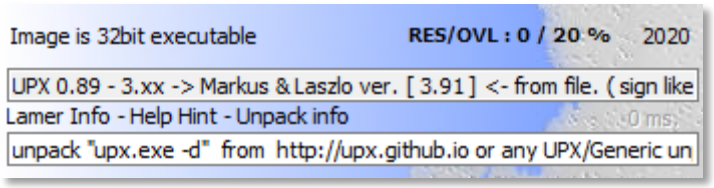
بررسی‌های اولیه بر روی فایل اجرایی این باج‌افزار نشان می‌دهد که باج‌افزار Nefilim بر روی تمامی نسخه‌های سیستم‌عامل ویندوز از ویندوز XP به بعد، اجرا خواهد شد.

Section Alignment	00001000h	
File Alignment	00000200h	
Operating System Version	00010005h	5.1
Image Version	00000000h	0.0
Subsystem Version	00010005h	5.1
Win32 Version Value	00000000h	Reserved

شکل ٣ – نسخه‌های سیستم‌عامل

٦-١ تحلیل کد:

با توجه به section های این باج‌افزار می‌توان فهمید که این یک فایل پک شده است.



شکل ٤ – نمایش نوع پکر باج‌افزار

بخش‌های کد و داده در فایل اجرایی باج‌افزار Nefilim با پکر UPX پک شده‌اند. پس از آنپک کردن فایل اجرایی، نتایج زیر حاصل شد.

در تابع اصلی کد این باج‌افزار، یک رشته کاراکتری که حاوی عبارتی به زبان روسی است، مشاهده می‌گردد که ترجمه آن به انگلیسی در زیر آمده است:

> Den gi plyvut v karmany rekoy. My khodim po krayu nozha...
>
> [floating in the pockets of the river. We walk along the edge of the knife]

باتوجه به اینکه در قطعه کد زیر زیر از تابع Mutex استفاده شده است، می‌توان پی برد که این باج‌افزار بصورت Multithread نوشته شده است.

```
; Attributes: noreturn bp-based frame

; int __cdecl main(int argc, const char **argv, const char **envp)
_main proc near

var_40= dword ptr -40h
var_3C= byte ptr -3Ch
var_20= dword ptr -20h
var_10= dword ptr -10h
var_C= dword ptr -0Ch
var_4= dword ptr -4
argc= dword ptr  8
argv= dword ptr  0Ch
envp= dword ptr  10h

push    ebp
mov     ebp, esp
sub     esp, 40h
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_4], eax
push    ebx
mov     ebx, [ebp+argv]
push    esi
push    edi
push    offset Name     ; "Den'gi plyvut v karmany rekoy. My khodi"...
xor     esi, esi
push    esi             ; bInitialOwner
push    esi             ; lpMutexAttributes
mov     [ebp+var_40], ebx
call    ds:CreateMutexA
push    esi             ; dwMilliseconds
push    eax             ; hHandle
call    ds:WaitForSingleObject
call    ds:GetLastError
```

1

2

شکل ۵ – تابع main

در ادامه، باج‌افزار با فراخوانی زیرروال دیگری، عملیات مربوط بـه مقـداردهی توابـع رمزنگـاری را انجـام می‌دهد. (نام این توابع به منظور سهولت در درک آن تغییر داده شده‌اند)

```
loc_402DB0:
call    initial_crypt
push    offset aNefilim ; "NEFILIM"
lea     eax, [ebp+var_20]
call    sub_402190
push    [ebp+var_10]    ; unsigned int
call    ??_U@YAPAXI@Z   ; operator new[](uint)
cmp     [ebp+var_C], 8
pop     ecx
mov     ecx, [ebp+var_20]
mov     lpBuffer, eax
jnb     short loc_402DDC
```

شکل ۶ – تابع main

باج‌افزار Nefilim دریکی از زیرروال‌ها، با استفاده از تابع FindFisrtFileW فایل‌های موجود در سیستم قربانی را فهرست گذاری (indexing) کرده و آن‌ها را با اسامی موجود در تصویر ۹ مقایسه می‌کند. سپس پسوند آن‌ها را نیز با تابع PathFindExtensionW مقایسه کرده (تصویر ۸) و اگر با فرمت‌های موجود در تصویر ۱۰ برابر بود فرایند رمزگذاری را انجام نمی‌دهد.

```
loc_4015CE:
lea     ecx, [esp+2F8h+FindFileData]
push    ecx             ; lpFindFileData
push    eax             ; lpFileName
call    ds:FindFirstFileW
mov     [esp+2F8h+hFindFile], eax
cmp     eax, 0FFFFFFFFh
jz      loc_401B0B
```

شکل ۷ – تابع main

```
loc_4018DA:
mov     esi, ds:lstrcmpiW
push    offset aExe     ; ".exe"
push    eax             ; lpString1
call    esi ; lstrcmpiW
test    eax, eax
jz      loc_401ADA
```

شکل ۸ – تابع main

```
; const WCHAR aWindows
aWindows:                               ; DATA XREF: FindFile+12C↑o
                text "UTF-16LE", 'windows',0
; const WCHAR aRecycleBin
aRecycleBin:                            ; DATA XREF: FindFile+143↑o
                text "UTF-16LE", '$RECYCLE.BIN',0
                align 4
; const WCHAR aRsa
aRsa:                                   ; DATA XREF: FindFile+15A↑o
                text "UTF-16LE", 'rsa',0
; const WCHAR aNtdetectCom
aNtdetectCom:                           ; DATA XREF: FindFile+171↑o
                text "UTF-16LE", 'NTDETECT.COM',0
                align 4
; const WCHAR aNtldr
aNtldr:                                 ; DATA XREF: FindFile+188↑o
                text "UTF-16LE", 'ntldr',0
; const WCHAR aMsdosSys
aMsdosSys:                              ; DATA XREF: FindFile+19F↑o
                text "UTF-16LE", 'MSDOS.SYS',0
; const WCHAR aIoSys
aIoSys:                                 ; DATA XREF: FindFile+1B6↑o
                text "UTF-16LE", 'IO.SYS',0
                align 4
; const WCHAR aBootIni
aBootIni:                               ; DATA XREF: FindFile+1CD↑o
                text "UTF-16LE", 'boot.ini',0
                align 4
; const WCHAR aAutoexecBat
aAutoexecBat:                           ; DATA XREF: FindFile+1E4↑o
                text "UTF-16LE", 'AUTOEXEC.BAT',0
                align 4
; const WCHAR aNtuserDat
aNtuserDat:                             ; DATA XREF: FindFile+1FB↑o
                text "UTF-16LE", 'ntuser.dat',0
```

شکل ۹ – تابع main

```
aMp4_0:                                 ; DATA XREF: .data:0040EC20↓o
                text "UTF-16LE", 'mp4',0
aMp3_0:                                 ; DATA XREF: .data:0040EC1C↓o
                text "UTF-16LE", 'mp3',0
aPif_0:                                 ; DATA XREF: .data:0040EC18↓o
                text "UTF-16LE", 'pif',0
aTtf_0:                                 ; DATA XREF: .data:0040EC14↓o
                text "UTF-16LE", 'ttf',0
aUrl_0:                                 ; DATA XREF: .data:0040EC10↓o
                text "UTF-16LE", 'url',0
aDll_0:                                 ; DATA XREF: .data:0040EC0C↓o
                text "UTF-16LE", 'dll',0
aIni_0:                                 ; DATA XREF: .data:0040EC08↓o
                text "UTF-16LE", 'ini',0
aCpl_0:                                 ; DATA XREF: .data:0040EC00↓o
                text "UTF-16LE", 'cpl',0
aCom_0:                                 ; DATA XREF: .data:0040EBFC↓o
                text "UTF-16LE", 'com',0
aCmd_0:                                 ; DATA XREF: .data:0040EBF8↓o
                text "UTF-16LE", 'cmd',0
aCab_0:                                 ; DATA XREF: .data:0040EBF4↓o
                text "UTF-16LE", 'cab',0
aLog_0:                                 ; DATA XREF: .data:0040EBF0↓o
                text "UTF-16LE", 'log',0
aExe_0:                                 ; DATA XREF: .data:0040EBEC↓o
                                        ; .data:0040EC04↓o
                text "UTF-16LE", 'exe',0
aLnk_0:                                 ; DATA XREF: .data:0040EBE8↓o
                text "UTF-16LE", 'lnk',0
```

شکل ۱۰ – تابع main

لیست کامل این پسوندها به صورت زیر است:

.mp4, .mp3, .piff, .ttf, .url, .dll, .ini, .cpl, .com, .cmd, .cab, .log, .exe, .lnk

همچنین برچسب درایوها با استفاده از تابع GetLogicalDrives به‌دست می‌آید.

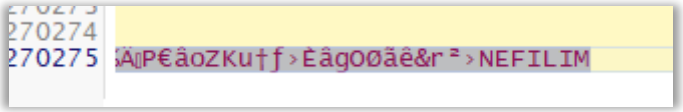
```
; Attributes: bp-based frame

sub_40206A proc near

hHandle= dword ptr -84h
var_1C= dword ptr -1Ch
var_18= dword ptr -18h
lpParameter= dword ptr -14h
var_10= dword ptr -10h
RootPathName= word ptr -0Ch
var_A= word ptr -0Ah
var_8= word ptr -8
var_6= word ptr -6
var_4= dword ptr -4

push    ebp
mov     ebp, esp
sub     esp, 84h
mov     eax, ___security_cookie
xor     eax, ebp
mov     [ebp+var_4], eax
push    ebx
push    esi
xor     esi, esi
push    edi
mov     [ebp+var_10], esi
call    ds:GetLogicalDrives
mov     edi, ds:HeapAlloc
mov     ebx, ds:GetProcessHeap
xor     ecx, ecx
mov     [ebp+var_1C], eax
mov     [ebp+var_18], ecx
```

شکل ۱۱ – تابع main

در ادامه کتابخانه‌های استفاده شده در کد باج‌افزار بارگذاری می‌شوند.

طبق بررسی‌های انجام شده، بعد از آنپک کردن فایل اجرایی باج‌افزار، برخی از آدرس‌های درون کد برنامه خراب می‌شوند و توانایی اجرا ندارند (خطای Access Violation). لذا احتمال وجود باگ در کد باج‌افزار بسیار زیاد است. طبق بررسی‌های صورت گرفته، باج‌افزار Nefilim از الگوریتم رمزنگاری AES-128 جهت رمزگذاری فایل‌های موردنظر خود در سیستم قربانی استفاده می‌کند.

نتایج بررسی‌ها بر روی نمونه فایل‌های رمزشده با نمونه سالم آن‌ها پس از پایان فعالیت باج‌افزار در سیستم قربانی نشان می‌دهد که این باج‌افزار، محدودیتی در اندازه فایل برای رمزگذاری ندارد. همچنین با توجه به بررسی چندین فایل، می‌بینیم که در انتهای همه آن‌ها عبارت NEFILIM را اضافه می‌کند.



شکل ۱۲ – تابع main

۶-۲ تحلیل ترافیک شبکه:

پس از بررسی ترافیک شبکه ضبط شده در حین اجرای باج‌افزار مورد مشکوکی مشاهده نگردید.

۶-۳ رمزگشایی:

در حال حاضر، هیچ‌گونه ابزاری جهت رمزگشایی فایل‌های رمزشده توسط این باج‌افزار، ارائه نشده است.