بسمه تعالی

مرکز مدیریت امداد و هماهنگی
عملیات رخدادهای رایانه‌ای
سازمان فناوری اطلاعات ایران
معاونت امنیت فضای تولید و تبادل اطلاعات

گزارش تحلیلی بدافزار NEPHILIM Ransomware

گزارش تحلیل بدافزار

شناسه سند ..... Maher_13990424-3
نوع سند ..... گزارش فنی
شماره نگارش ۰,۱
تاریخ نگارش ۱۳۹۹/۰۴/۲۲
طبقه‌بندی سند ..... عادی

تهران- میدان آرژانتین- ابتدای بلوار بیهقی- نبش خیابان شانزدهم- ساختمان شماره ۱ سازمان فناوری اطلاعات ایران
(۰۲۱)۴۲۶۵۰۰۰۰
(۰۲۱)۴۲۶۵۰۰۰۰
cert.shahroodut.ac.ir

فهرست مطالب

۱ مقدمه

در چندین سال اخیر که شیوع بدافزارها در دنیای دیجیتال رشد زیادی داشته است باجافزارها رشد زیادی داشته‌اند و روزانه انواع مختلفی از آن ایجاد و انتشار می‌یابد. یکی از این باجافزارها NEPHILIM می‌باشد که با توجه به یافته‌ها و بررسی‌های محققین حوزه بدافزار اواخر March سال ۲۰۲۰ میلادی انتشار یافته است. هدف اولیه این باجافزار کاربران انگلیسی زبان بوده اما این توانایی را دارد که فایل‌هایی با عناوین فارسی را نیز بصورت رمزگذاری شده دربیارند.

این باجافزار با استفاده از الگوریتم‌های رمزنگاری پیچیده تمامی فایل‌های سیستم را به حالت رمزشده تبدیل کرده و پسوند NEPHILIM. را به انتهای آن‌ها اضافه می‌کند. بطور مثال فایلی با نام 123.jpg بصورت 123.jpg.NEPHILIM تبدیل می‌گردد. همچنین براساس آخرین اطلاعات موجود، این باجافزار نیز همانند باجافزارهای دیگر از طریق ایمیل‌های اسپم یا فایل‌های جعلی آپدیت نرم‌افزارهای قانونی انتشار می‌یابد و الگوریتم‌های مورد استفاده نیز با توجه به تحلیل و بررسی‌های صورت گرفته احتمال داده می‌شود RSA و AES باشد که در ادامه به آن‌ها اشاره خواهد شد.

مهاجمان بصورت مستقیم مبلغ باج را تعیین نکرده‌اند و کاربران قربانی شده باید با استفاده از آدرس‌های ایمیلی که در فایل راهنما قرار دارند با مهاجمان ارتباط برقرار کنند تا مقدار و نحوه پرداخت باج مشخص گردد. آدرس‌های ایمیل بصورت Deanlivermore@protonmail.com، robertatravels@mail.com و Bernardocarlos@tutanota.com می‌باشند. همچنین برای اطمینان از اینکه فایل‌های رمزشده به حالت قبلی و قابل استفاده باز می‌گردند مهاجمان درخواست دو فایل دارند که می‌توان به این آدرس‌ها ارسال کرد.

۲ مشخصات و ریزجزئیات فایل باج‌افزار

جداول و نمودارهای موجود در این بخش نشان دهنده ریزجزئیات فایل اجرایی باج‌افزار می‌باشند که در طول تحلیل‌های استاتیک و پویا توسط ابزارهای مختلف بدست آمده‌اند. این اطلاعات شامل مواردی همچون اندازه فایل، مقادیر هش فایل، آنتروپی، وضعیت شناسایی فایل در ویروس‌توتال و ویروس‌کاو و غیره می‌باشد.

۲-۱ مشخصات فایل

همانطور که قبلا ذکر گردید این بدافزار از خانواده باج‌افزار و رمزگذار فایل می‌باشد که با استفاده از زبان برنامه‌نویسی C++ طراحی و پیاده‌سازی شده است. نام آن نیز با استفاده از پسوندی که به انتهای فایل‌ها اضافه می‌گردد NEPHILIM نامگذاری شده است.

جدول ۱ - ریزجزئیات مربوط به باج‌افزار

3BEB3D466BCC0977EC2DD66D72AB6BB3	**هش md5**
E94089137A41FD95C790F88CC9B57C2B4D5625BA	**هش SHA1**
B227FA0485E34511627A8A4A7D3F1ABB6231517BE62D022916273B7A51B80A17	**هش SHA256**
Ransomware, Crypto Locker	**نوع بدافزار**
NEPHILIM	**نام بدافزار**
weeli.exe	**نام فایل اجرایی**
.NEPHILIM	**پسوند**
Email Spam, Etc.	**نحوه انتشار**
Mar/25/2020	**زمان کامپایل**
Microsoft Visual C++	**کامپایلر**
18424 bytes	**حجم فایل**
6.542	**آنتروپی کلی فایل**
32 bits	**معماری فایل**
4	**تعداد بخش**
-	**آدرس فایل Pdb**

۲-۲ بخش‌های مختلف فایل

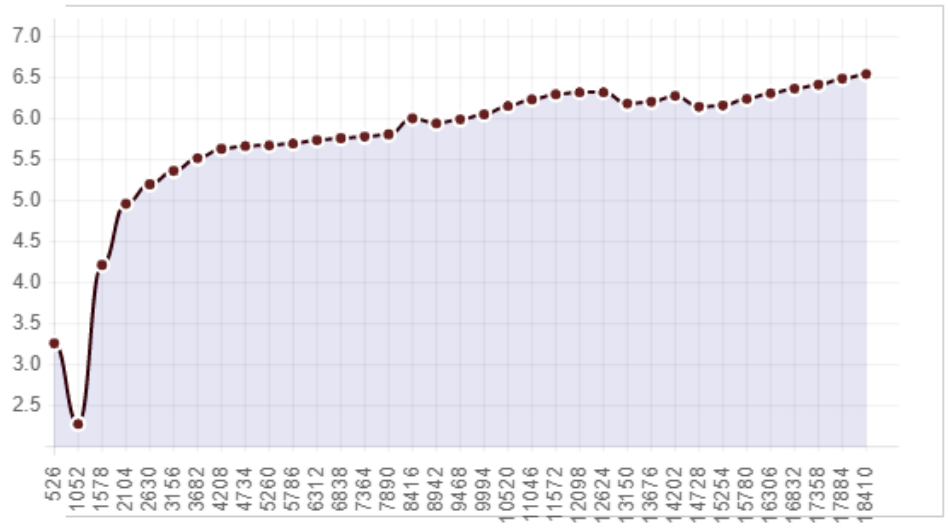
جدول موجودر در زیر نیز بخش‌های مختلف تشکیل دهنده فایل باجافزار را با جزئیات کامل مانند مقدار آنتروپی، اندازه خام، اندازه مجازی هر بخش و غیره نشان می‌دهد. این فایل متشکل از چهار بخش بصورت text، rdata، data و reloc بصورت زیر می‌باشد.

جدول ۲- بخش‌های و مشخصات مربوط به آن‌ها

ردیف	نام	آدرس مجازی	اندازه مجازی	اندازه خام	آنتروپی	بایت‌های اولیه
1	.text	00001000	000019BF	00001A00	6.034	55 8B EC 83 EC 24 53 56 57
2	.rdata	00003000	0000137E	00001400	6.090	B0 41 00 00 BE 41 00 00 D0
3	.data	00005000	00000120	00000200	0.170	98 3F 40 00 88 3F 40 00 60
4	.reloc	00006000	00000560	00000600	4.842	00 10 00 00 CC 01 00 00 A3

۲-۳ آنتروپی کلی فایل

شکل زیر وضعیت آنتروپی کلی فایل را در حالت عادی بصورت نموداری نشان می‌دهد. مقدار این آنتروپی برابر با 6.542 می‌باشد که مقدار آن کمتر از هفت می‌باشد.

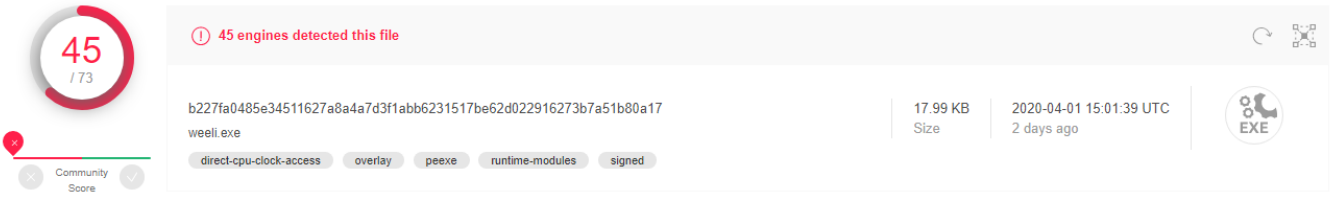


شکل ۱- مقدار و وضعیت آنتروپی کلی فایل

با توجه به جدول شماره ۲ و شکل ۱، مقدار آنتروپی کلی فایل نزدیک هفت و بصورت صعودی می‌باشدو همچنین آنتروپی بخش data. نیز تقریبا صفر می‌باشد. مقادیر بالای هفت و روند صعودی و همچنین مقدار صفر آنتروپی دگردیسی و چندریختی و مشکوک بودن فایل را نشان می‌دهد.

۴-۲ وضعیت شناسایی فایل در ویروس‌توتال

شکل زیر وضعیت شناسایی فایل را در ویروس‌توتال نشان می‌دهد. در این سامانه از بین ۷۳ موتور تحلیل ۴۵ موتور قادر به شناسایی فایل بعنوان یک فایل بدافزار شده‌اند و درصورت استفاده از نسخه‌های بروزشده این موتورهای آنتی‌ویروس در سیستم می‌توان از انتقال و اجرای آن جلوگیری کرد.



شکل ۲- وضعیت تشخیص فایل در ویروس‌توتال

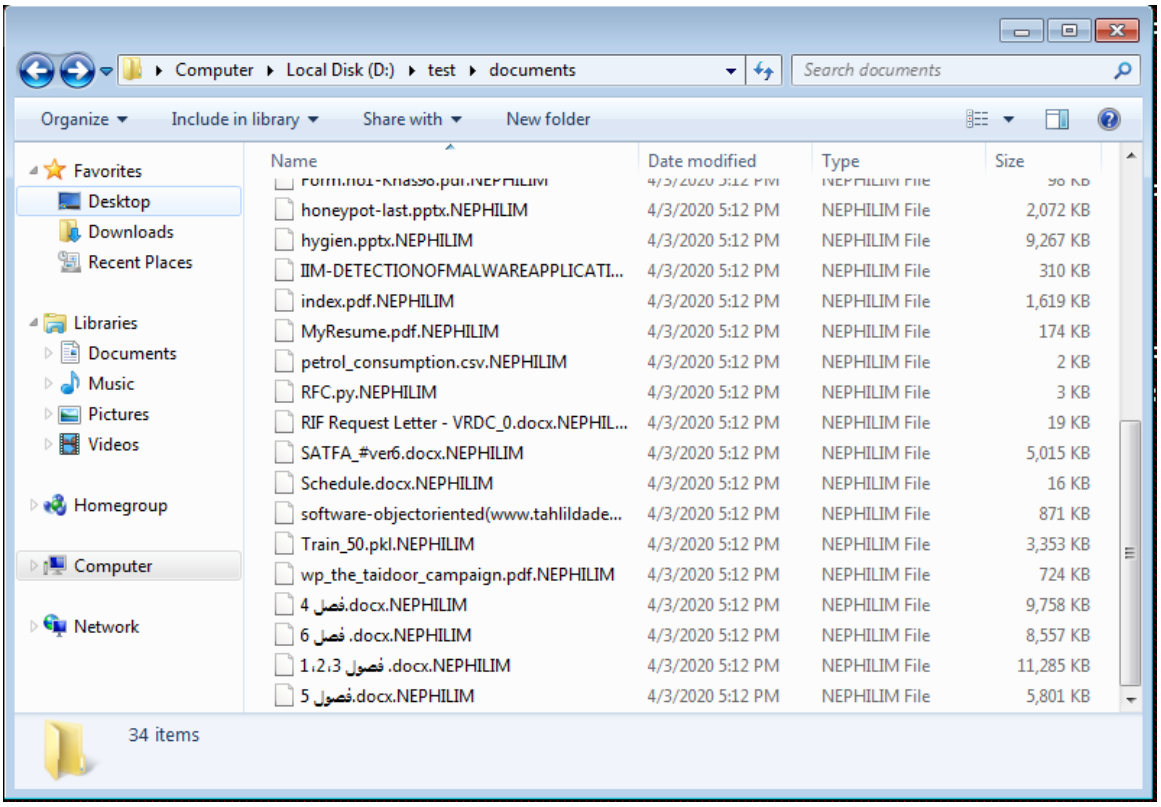
۵-۲ وضعیت شناسایی فایل در ویروس‌کاو

شکل زیر نیز وضعیت شناسایی فایل را در سامانه بومی ویروس‌کاو نشان می‌دهد. از بین ۳۰ موتور موجود تعداد ۱۲ موتور قادر به شناسایی بعنوان فایل مخرب و بدافزار شده‌اند. از بین این موتورها، موتورهای بومی ستفا و پادویش قادر به شناسایی فایل بعنوان یک فایل مخرب می‌باشند.

شکل ۳ - وضعیت تشخیص فایل در ویروس‌کاو

۳ فرایند آلوده‌سازی

این باج‌افزار بعد از انتقال به سیستم با استفاده از الگوریتم‌های رمزنگاری پیچیده تمامی فایل‌های سیستم را به حالت رمزشده تبدیل کرده و پسوند NEPHILIM. را به انتهای آن‌ها اضافه می‌کند. بعد از رمزگذاری فایل‌ها یک فایل متنی راهنما را در پوشه‌های مختلفی از سیستم ایجاد کرده و یک فایل عکس را در مسیر C:\Users\Tahlilgar\AppData\Local\Temp را با نام god.jpg (شکل شماره ۵) ایجاد کرده و تصویر زمینه سیستم را با این عکس تغییر می‌دهد. در این عکس محتویات فایل متنی راهنما قرار دارد که برای کاربر برای انجام مراحل بعدی از در برقراری ارتباط با مهاجمان نشان داده می‌شود. براساس آخرین اطلاعات موجود، این باج‌افزار نیز همانند باج‌افزارهای دیگر از طریق ایمیل‌های اسپم یا فایل‌های جعلی آپدیت نرم‌افزارهای قانونی انتشار می‌یابد و الگوریتم‌های مورد استفاده نیز با توجه به تحلیل و بررسی‌های صورت گرفته احتمال داده می‌شود RSA و AES باشد که در ادامه به آن‌ها اشاره خواهد شد.



شکل ۴ – نمونه فایل‌های رمزشده توسط باج‌افزار

شکل ۵ – تصویر مربوط به صفحه اصلی سیستم

۴ شرح تحلیل

این بخش از گزارش نتیجه تحلیل و بررسی فایل باج‌افزار را توسط ابزارهای تحلیل در قسمت‌های مختلف نشان می‌دهد و شامل مواردی مانند کتابخانه و توابع، رشته‌ها، فعالیت‌های شبکه و غیره می‌باشند.

۴-۱ کتابخانه و توابع مورد استفاده

فایل اجرایی باج‌افزار با استفاده از تکنیک‌های مبهم‌سازی، توابع و رشته‌های آن را تغییر داده که هنگام دیس‌اسمبل کردن فایل، تعداد کتابخانه و توابع را محدود نشان داده و رشته‌ها را بصورت کاراکترهای ناخوانا نشان می‌دهد. لذا هنگام فرایند دیس‌اسمبل کتابخانه و توابع موجود در جدول زیر بدست می‌آید.

جدول ۳ – کتابخانه و توابع مورد استفاده از باج‌افزار

کتابخانه و توابع	Kernel32.dll
	ExitProcess, FindFirstFileW, lstrlenA, GetDriveTypeW, HeapAlloc, SetFilePointerEx, HeapFree, WaitForSingleObject, GetLogicalDrives, GetProcessHeap, WriteFile, Sleep, ReadFile, CreateFileW, GetFileSizeEx, GetLastError, SetLastError, MoveFileW, FindClose, lstrcmpiW, lstrcatW, FindNextFileW, CloseHandle, lstrcpyW, CreateThread, GetTempPathW, GetProcAddress, LoadLibraryA, CreateMutexA, GetCommandLineW

از بین این توابع به موارد مشکوکی مانند CreateFileW، CreateThread، FindFirstFileW، FindNextFileW، GetCommandLineW، GetDriveTypeW، GetFileSizeEx، GetProcAddress، GetTempPathW، LoadLibraryA، Sleep، WriteFile اشاره گرد.

رشته‌های موجود و قابل دسترس هنگام دیس‌اسمبل نیز در جدول زیر قابل مشاهده می‌باشد که در برخی موارد با استفاده از عملیات مبهم‌سازی[۱] به رشته‌های ناخوانا که معنی و مفهوم خاصی ندارد، تبدیل شده‌اند.

جدول ۴ - رشته‌های قابل استخراج از فایل باج‌افزار

رشته‌های قابل دریافت	https://sectigo.com/CPS0C 6NEPHILIM-DECRYPT.txt $RECYCLE.BIN NTDETECT.COM MSDOS.SYS AUTOEXEC.BAT ntuser.dat \god.jpg http://ocsp.sectigo.com0 http://ocsp.usertrust.com0 appdata advapi32.dll user32.dll shell32.dll gdi32.dll KERNEL32.dll 2http://crl.sectigo.com/SectigoRSACodeSigningCA.crl0s 2http://crt.sectigo.com/SectigoRSACodeSigningCA.crt0# New Jersey1 Jersey City1

[۱] Obfuscation

The USERTRUST Network1.0,

%USERTrust RSA Certification Authority0

Sectigo RSA Code Signing CA0

?http://crl.usertrust.com/USERTrustRSACertificationAuthority.crl0v

3http://crt.usertrust.com/USERTrustRSAAddTrustCA.crt0%

.NEPHILIM

6NEPHILIM-DECRYPT.txt

$RECYCLE.BIN

NTDETECT.COM

MSDOS.SYS

AUTOEXEC.BAT

ntuser.dat

\god.jpg

http://ocsp.sectigo.com0

http://ocsp.usertrust.com0

appdata

advapi32.dll

user32.dll

shell32.dll

gdi32.dll

KERNEL32.dll

http://crl.sectigo.com/SectigoRSACodeSigningCA.crl0s

http://crt.sectigo.com/SectigoRSACodeSigningCA.crt0#

New Jersey1

Jersey City1

The USERTRUST Network1.0,

%USERTrust RSA Certification Authority0

Sectigo RSA Code Signing CA0

http://crl.usertrust.com/USERTrustRSACertificationAuthority.crl0v

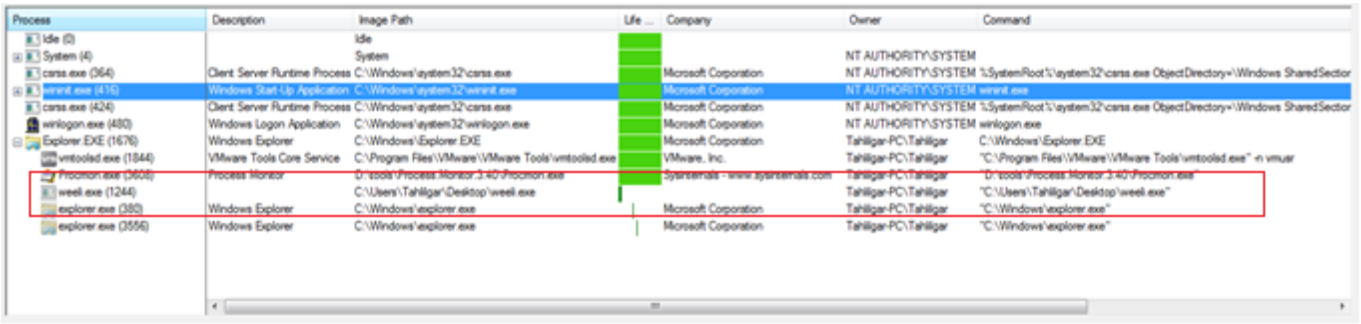
.NEPHILIM

https://sectigo.com/CPS0C http://crl.sectigo.com/SectigoRSACodeSigningCA.crl http://ocsp.sectigo.com http://crl.usertrust.com/USERTrustRSACertification http://crt.usertrust.com/USERTrustRSAAddTrustCA.cr http://ocsp.usertrust.com

از بین رشته‌ها می‌توان به مواردی همانند آدرس سایت‌های مختلف، پسوند اضافه شده به فایل‌ها، نام فایل تصویر ایجاد شده، کتابخانه‌های سیستمی مختلف، آدرس مسیرهایی از سیستم و غیره اشاره کرد.

۴-۲ پروسس‌های ایجاد شده توسط باج‌افزار

شکل زیر پروسس‌ها و فرایندهای ایجاد شده در سیستم را در طول اجرا و فعالیت باج‌افزار را نشان می‌دهد. باج‌افزار بدون اجرای هیچ زیرپروسسی با نام weeli.exe به فعالیت خود ادامه داده و اقدام به رمزگذاری فایل‌های سیستم می‌کند.



شکل ۶- ساختار درختی پروسس‌های اجرای در طول فعالیت باج‌افزار

۴-۳ فایل‌های ایجاد شده

همانطور که قبلا نیز ذکر گردید باج‌افزار در طول فعالیت خود فایل‌هایی را در مسیرهایی از سیستم ایجاد می‌کند. اما این فایل‌ها مخرب نبوده و هیچ خرابی را در سیستم بوجود نمی‌آورند. فایل‌های ایجاد شده به همراه مسیر آن‌ها بصورت موارد زیر می‌باشند.

1. C:\Users\Tahlilgar\AppData\Local\Temp\god.jpg
2. C:\NEPHILIM-DECRYPT.txt
3. C:\Windows\Fonts\comic.ttf
4. C:\Windows\Fonts\StaticCache.dat

۴-۴ تغییرات رجیستری

باج‌افزار بعد از اجرا و رمزکردن فایل‌های سیستم باعث ایجاد تغییراتی در رجیستری سیستم شده و کلیدهایی ثبت یا تغییر و یا حذف می‌کند. آدرس‌های رجیستری زیر این موارد را نشان می‌دهد که بصورت لیست‌وار آورده شده‌اند.

- کلیدهای رجیستری ایجاد شده:

1. HKCU\Control Panel\Desktop\Wallpaper

- کلیدهای رجیستری باز وخوانده شده:

1. HKLM\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Strong Cryptographic Provider
2. HKLM\SYSTEM\CurrentControlSet\Policies\Microsoft\Cryptography\Configuration
3. HKLM\Software\Microsoft\Cryptography
4. HKLM\Software\Policies\Microsoft\Windows NT\Rpc

- کلیدهای رجیستری تغییر داده شده:

1. HKU\S-1-5-21-2137028633-1772422641-3009315880-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache\C:\Users\Tahlilgar\Desktop\weeli.exe: "weeli"
2. HKU\S-1-5-21-2137028633-1772422641-3009315880-1001_Classes\Local Settings\Software\Microsoft\Windows\Shell\MuiCache\C:\Users\Tahlilgar\Desktop\weeli.exe: "weeli"

۵ بررسی الگوریتم‌های رمزگذاری

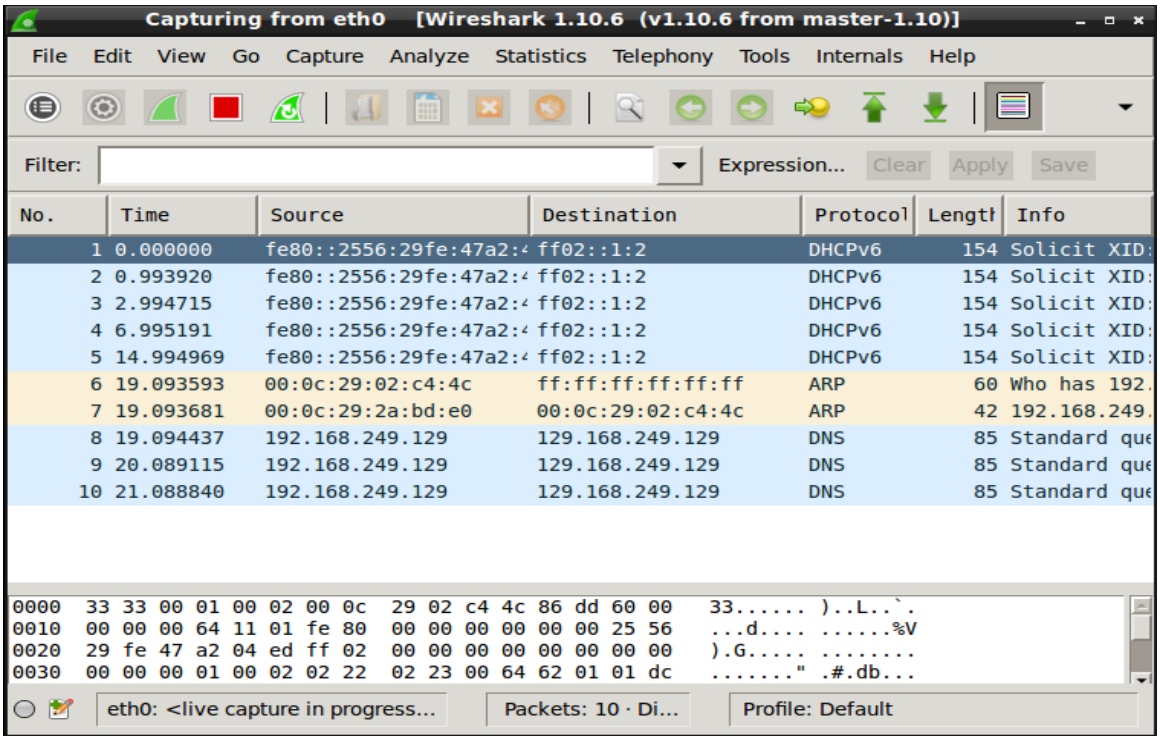
در این بخش به بررسی الگوریتم‌های و توابع رمزگذاری استفاده شده در فایل اجرایی باج‌افزار می‌پردازیم که با استفاده از ابزار SignSearch بدست آمده است. ایست زیر این الگوریتم‌ها و توابع را نشان می‌دهند.

1. AES Rijndael S / ARIA S1
2. AES Rijndael Si / ARIA X1
3. Windows CryptDecrypt
4. Windows CryptAcquireContext
5. Windows CryptImportKey
6. Windows CryptDeriveKey
7. Windows CryptCreateHash
8. Windows CryptHashData
9. SSH RSA id-sha1 OBJ.ID. oiw(14) secsig(3) algorithms(2) 26

از بین این ۹ مورد می‌توان به الگوریتم‌های AES و RSA اشاره کرد که احتمال داده می‌گردد از این دو برای رمزگذاری فایل‌ها استفاده شده باشد. همچنین توابعی را مشاهده می‌کنیم که در داخل بدنه آن‌ها از فرایندهای رمزگذاری استفاده شده است.

۵-۱ ارتباطات شبکه

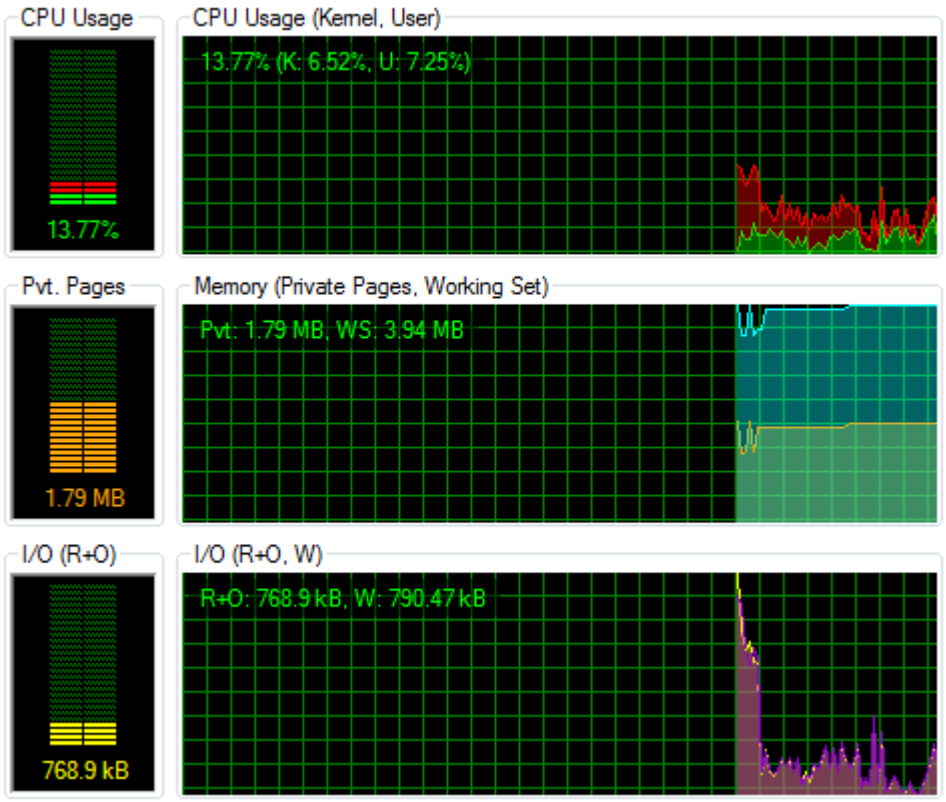
در طول اجرای باج‌افزار در سیستم هیچ نوع فعالیتی مبنی بر ارتباط شبکه مشاهده نگردید. که در شکل زیر می‌توان این فعالیت را مشاهده کرد.



شکل ۷ - بررسی فعالیت شبکه در طول اجرای باج‌افزار

۵-۲ وضعیت منابع سیستم

شکل زیر وضعیت منابع سیستم را فقط برای فایل اجرایی باج‌افزار نشان می‌دهد که در حال مصرف است. با توجه به شکل می‌توان مشاهده کرد که مدت زمان اجرایی فایل کمتر بوده و بعد از مدت کوتاهی باعث Terminate شدن پروسس اجرایی می‌شود. همچنین در شکل مشاهده می‌گردد که میزان استفاده از Memory و I/O بدلیل خواندن و نوشتن فایل‌های سیستم بیشتر می‌باشد.



شکل ۸- وضعیت منابع سیستم در طول اجرای باج‌افزار

۶ تحلیل کد

۶-۱ بارگذاری کتابخانه

با توجه به شکل زیر باج‌افزار با استفاده از تابع LoadLibrary سعی در بارگذاری کتابخانه Shell32.dll داشته تا با استفاده از توابعی همچون CommandLineToArgv اقدام به اجرا و انجام عملیاتی در سیستم داشته باشد.

```
push    ebp
mov     ebp, esp
and     esp, 0FFFFFFF8h
sub     esp, 210h
push    esi
push    edi
call    sub_40279A
xor     edi, edi
mov     [esp+218h+var_20C], edi
call    ds:GetCommandLineW
mov     esi, eax
cmp     dword_4050FC, edi
jnz     short loc_402911
push    offset aShell32Dll ; "shell32.dll"
call    ds:LoadLibraryA
mov     dword_4050FC, eax

                        ; CODE XREF: start+27↑j
mov     eax, dword_4050F8
cmp     eax, edi
jnz     short loc_402930
push    offset aCommandlinetoa ; "CommandLineToArgvW"
push    dword_4050FC    ; hModule
call    ds:GetProcAddress
mov     dword_4050F8, eax
```

شکل ۹- بارگذاری کتابخانه shell32.dll

یا در شکل زیر می‌توان بارگذاری کتابخانه User32.dll را مشاهده کرد.

```
cmp     dword_40508C, ebx
jnz     short loc_4024B2
push    offset aUser32Dll ; "user32.dll"
call    ds:LoadLibraryA
mov     dword_40508C, eax

                        ; CODE XREF: sub_402056+44A↑j
mov     eax, dword_405088
cmp     eax, ebx
jnz     short loc_4024D1
push    offset aReleasedc ; "ReleaseDC"
push    dword_40508C    ; hModule
call    ds:GetProcAddress
mov     dword_405088, eax
```

شکل ۱۰ – بارگذاری کتابخانه user32.dll

باج‌افزار علاوه بر کتابخانه Kernell32.dll از کتابخانه‌های Shell32.dll، User32.dll و gdi32.dll با استفاده از بارگذاری در طول فایل باج‌افزار استفاده می‌کند.

۲-۶ نوشتن فایل

باج‌افزارها بعد از رمزگذاری فایل‌های سیستم اقدام به نوشتن فایل رمزشده با پسوند جدید می‌کنند. برای نوشتن فایل از تابع WriteFile استفاده می‌گردد که یک مورد از استفاده از این تابع را در شکل زیر می‌توان مشاهده کرد.

```
call    sub_401D9B
pop     ecx
pop     ecx
push    ebx             ; lpOverlapped
test    eax, eax
jz      short loc_40138A
lea     eax, [ebp+NumberOfBytesWritten]
push    eax             ; lpNumberOfBytesWritten
push    [ebp+dwBytes]   ; nNumberOfBytesToWrite
push    [ebp+lpBuffer]  ; lpBuffer
push    [ebp+hFile]     ; hFile
call    ds:WriteFile
```

شکل ۱۱- استفاده از تابع WriteFile برای نوشتن فایل

۶-۳ ایجاد فایل جدید

همانطور که قبلا نیز ذکر گردید باج‌افزار در برخی از مواقع اقدام به ایجاد فایل‌هایی همانند god.jpg در سیستم می‌کند. این کار با استفاده از تابع CreateFile صورت می‌گیرد که نمونه‌ای از این استفاده را می‌توان در شکل زیر مشاهده کرد.

```
call    eax ; dword_405088
push    ebx             ; hTemplateFile
push    80h             ; dwFlagsAndAttributes
push    2               ; dwCreationDisposition
push    ebx             ; lpSecurityAttributes
push    ebx             ; dwShareMode
push    40000000h       ; dwDesiredAccess
lea     eax, [ebp+Buffer]
push    eax             ; lpFileName
call    ds:CreateFileW
mov     edi, eax
cmp     edi, 0FFFFFFFFh
jz      loc_4025C9
mov     esi, ds:WriteFile
push    ebx             ; lpOverlapped
lea     eax, [ebp+NumberOfBytesWritten]
push    eax             ; lpNumberOfBytesWritten
push    0Eh             ; nNumberOfBytesToWrite
```

شکل ۱۲ - استفاده از تابع CreateFile برای ایجاد فایل جدید

۶-۴ بدست آوردن محیط cmd

باج‌افزار بعد از اجرای فایل اقدام به بارگذاری کتابخانه Shell32.dll کرده و تابع GetCommandLine را اجرا می‌کند. با اجرای این تابع محیط cmd را در اختیار گرفته و می‌تواند دستوراتی را اجرا کند. اما در طول تحلیل با ابزارها هیچ عملکردی مبنی بر اجرای دستورات مشاهده نگردید.

```
push    ebp
mov     ebp, esp
and     esp, 0FFFFFFF8h
sub     esp, 210h
push    esi
push    edi
call    sub_40279A
xor     edi, edi
mov     [esp+218h+var_20C], edi
call    ds:GetCommandLineW
mov     esi, eax
cmp     dword_4050FC, edi
jnz     short loc_402911
```

شکل ۱۳ - بدست آوردن محیط cmd با استفاده از تابع GetCommandLine

۷ توصیه‌های امنیتی برای پیشگیری

۱) گرفتن فایل پشتیبان بصورت دوره‌ای از فایل‌های سیستم و ذخیره آن در محل دیگر
۲) استفاده از آنتی‌ویروس قوی و بروزرسانی مداوم آن
۳) خودداری از بازکردن و اجرا فایل‌های مشکوک و ناشناس
۴) خودداری از بازکردن ایمیل‌های مشکوک و ناشناس
۵) اطمینان از سالم بودن دستگاه‌های جانبی مانند فلش
۶) استفاده از رمزعبور قوی بر روی درایوهای سیستم
۷) استفاده از سیستم‌عامل جدید و بروزرسانی شده
۸) بروزرسانی مداوم سیستم عامل
۹) پیکربندی مناسب پروتکل‌های مورد استفاده در شبکه متناسب با محیط کار