باسمه تعالی

تحلیل فنی باج افزار AnimusLocker

مقدمه :

مشاهده و رصد فضای سایبری در زمینه باج‌افزار، از شروع فعالیت نمونه جدیدی به نام AnimusLocker خبر می‌دهد. بررسی‌ها نشان می‌دهد که فعالیت این باج‌افزار در اواخر ماه ژوئن سال ۲۰۱۸ میلادی شروع شده و به نظر می‌رسد تمرکز آن بیشتر بر روی کاربران انگلیسی زبان می‌باشد. این باج‌افزار از الگوریتم رمزنگاری AES ۲۵۶ بیتی برای رمزگذاری استفاده می‌کند و به جز فایل‌هایی با پسوندهای مشخص که در ادامه به آن‌ها اشاره خواهیم نمود، بقیه فایل‌ها را رمزگذاری می‌کند. طبق بررسی‌های انجام شده باج‌افزار AnimusLocker و باج‌افزار Aurora از یک خانواده می‌باشند و ریشه‌یابی آن‌ها به صورت زیر می‌باشد :

Aurora + Generic Malware > AnimusLocker

این باج‌افزار پس از رمزگذاری فایل‌ها پسوند آن‌ها را به animus. تغییر می‌دهد و همانند اکثر باج‌افزارها، از قربانیان تقاضای بیت‌کوین می‌کند و طبق اخبار دریافت شده، محققان امنیتی حوزه‌ی باج‌افزار موفق به رمزگشایی فایل‌های رمزگذاری شده توسط این باج‌افزار گردیده‌اند.

مشخصات فایل اجرایی :

نام فایل	Ransom.exe
MD۵	c۹c٤٧١١٣٥٥a٧٦d٥b٦٥٤٩cc٨٩٩٤٦a٩b٠٨
SHA-١	٢٥١٥٩fcc٥٠٣٢٨٨bfd٩٥٦٥٠٠٠b٩ae٢٤f١f١d٤e٥c٨
SHA-٢٥٦	٤١ff٣٧٨dcb٠c١eacc٣٧٦٦a٨٦٨c٨e٠٢٤٥٧٨٢c٧f٨٤٩d٦e٧٨٣٨٠c٧٧٩٩b٧٧٧١f٢e٢b
اندازه فایل	٣٨٠.٥ KB
کامپایلر	VC٨ -> Microsoft Corporation

فایل اجرایی این باج افزار دارای هفت بخش است :

نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	٦.٦١	٤٠٩٦	١٧٣٤٥٥	١٧٣٥٦٨
.rdata	٥.٥٩	١٨٠٢٢٤	٦٧٩١٢	٦٨٠٩٦
.data	٧.٧٩	٢٤٩٨٥٦	١٣٢٨٨٠	١٢٩٠٢٤
.gfids	٣.١٨	٣٨٥٠٢٤	٧٧٦	١٠٢٤

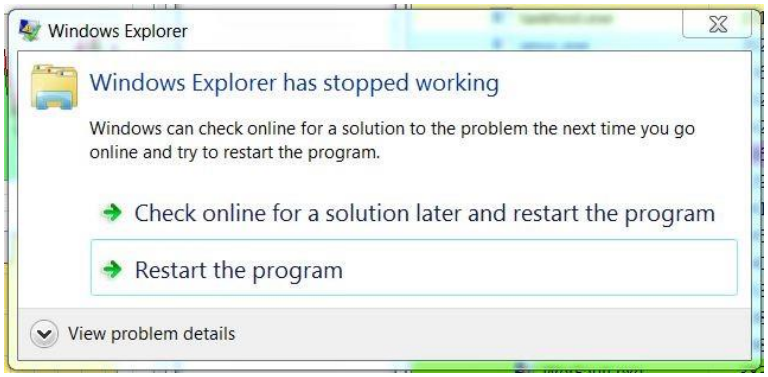
.tls	۰	۳۸۹۱۲۰	۵۰۲۹	۵۱۲۰
.rsrc	٤.۷۲	۳۹۷۳۱۲	٤۸۰	۵۱۲
.reloc	٦.۵۳	٤۰۱٤۰۸	۱۰۸۹۲	۱۱۲٦٤

تحلیل پویا :

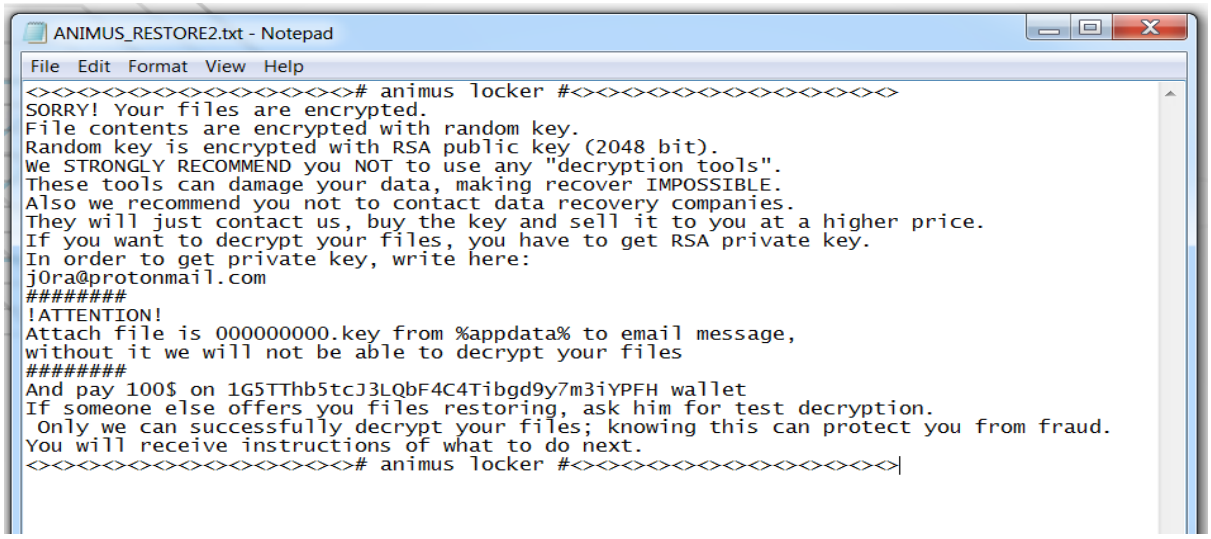
برای بررسی عمیق‌تر باج‌افزار AnimusLocker، فایل اجرایی آن را در محیط آزمایشگاهی اجرا کردیم تا عملکرد باج‌افزار را از نزدیک مورد بررسی قرار دهیم. نتایج حاصل از این بررسی نشان داد که باج‌افزار مورد اشاره در طول اجرا ۳ فایل متنی که محتوای هر یک از آن‌ها شامل پیغام باج‌خواهی می‌باشد را بر روی Desktop و در دایرکتوری‌های مختلف ایجاد می‌کند که نام این فایل‌ها به صورت زیر می‌باشد :

۱. ANIMUS_RESTORE.txt
۲. ANIMUS_RESTORE۲.txt
۳. ANIMUS_RESTORE۳.txt

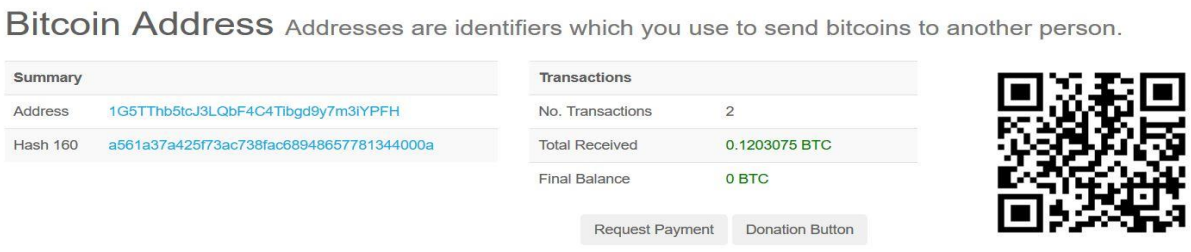
این باج‌افزار فایل‌ها را با استفاده از الگوریتم رمزنگاری AES ۲۵٦ بیتی رمزگذاری کرده و پسوند فایل‌ها را پس از رمزگذاری به animus. تغییر می‌دهد. در حین اجرای باج‌افزار، فعالیت Windows Explorer متوقف می‌شود و پیغام زیر به نمایش در می‌آید:



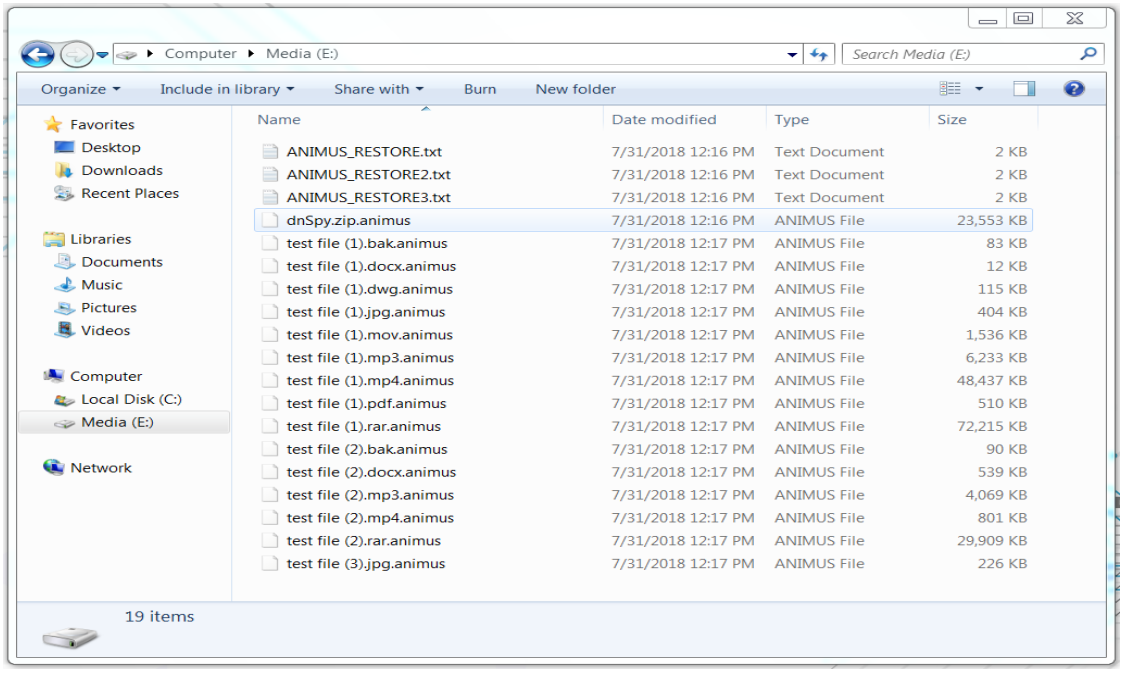
پس از آن، فرایند مربوط به اجرای باج‌افزار خاتمه پیدا می‌کند و پیغام باج‌خواهی به نمایش در می‌آید. تصویر زیر پیغام باج‌خواهی باج افزار AnimusLocker را نشان می‌دهد.



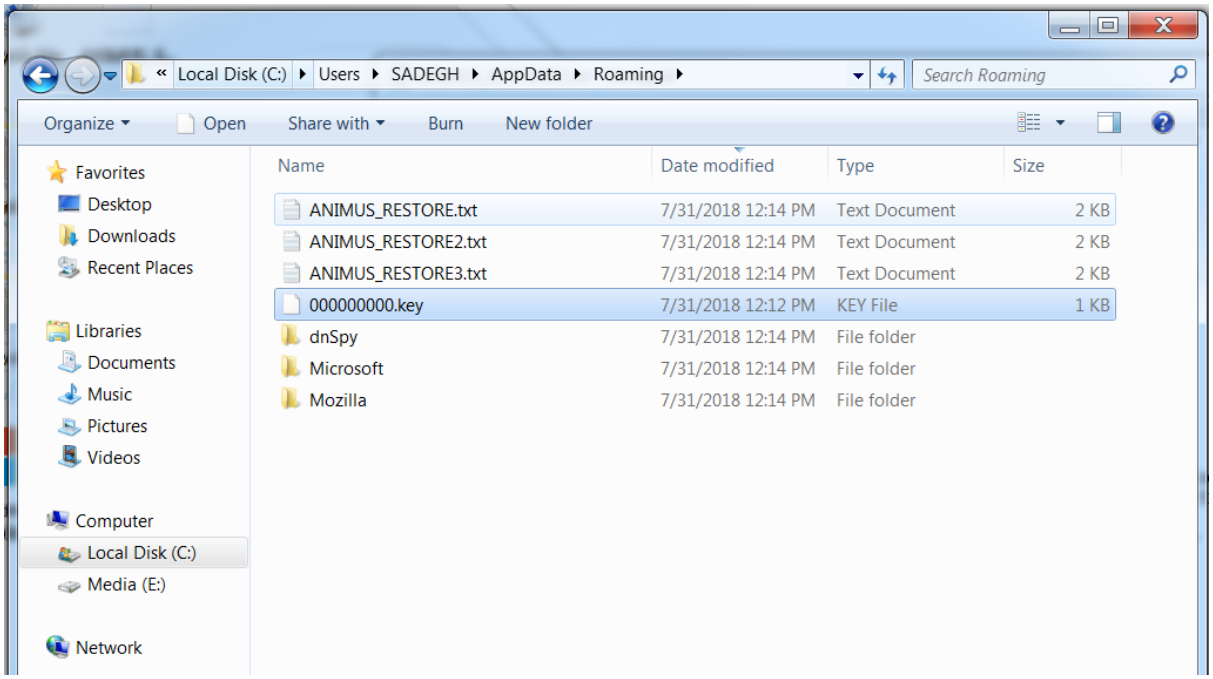
بر اساس پیغام باج‌خواهی مهاجمین اعلام نموده‌اند که فایل‌ها را با استفاده از یک کلید تصادفی رمزگذاری نموده‌اند و آن کلید تصادفی را با استفاده از الگوریتم رمزنگاری RSA ۲۰۴۸ بیتی رمزگذاری نموده‌اند و یادآور شده‌اند که قربانیان با استفاده از هیچ گونه ابزار رمزگشایی نمی‌توانند فایل‌ها را رمزگشایی نمایند، در غیر این صورت ممکن است به آن‌ها آسیب برسانند. قربانیان برای رمزگشایی فایل‌ها بایستی از طریق آدرس ایمیل j۰ra@protonmail.com با مهاجمین ارتباط برقرار نمایند تا کلید خصوصی را جهت رمزگشایی فایل‌ها خریداری نمایند. مهاجمین از قربانیان تقاضا نموده‌اند در ایمیل ارسالی یک فایل با نام ۰۰۰۰۰۰۰۰۰۰.key که در مسیر %appdata% قرار دارد و پس از اجرای باج‌افزار ایجاد شده است، را پیوست نمایند، در غیر این صورت آن‌ها قادر به رمزگشایی فایل‌ها نیستند. در انتهای پیغام باج‌خواهی مهاجمین به قربانیان اعلام نموده‌اند که معادل مبلغ ۱۰۰ دلار را به کیف پول بیت‌کوین به آدرس ۱G۵TThb۵tcJ۳LQbF٤C٤Tibgd۹y۷m۳iYPFH ارسال نمایند. طبق بررسی‌های انجام شده، در حال حاضر کیف پول مربوط به این باج‌افزار تاکنون تعداد ۲ تراکنش برابر با ۰.۱۲۰۳۰۷۵ BTC داشته است.



همانطور که پیشتر اشاره کردیم، این باج‌افزار پس از رمزگذاری فایل‌ها، پسوند آن‌ها را به animus. تغییر می‌دهد، تصویر زیر نشان‌دهنده فایل‌های رمزگذاری شده توسط این باج افزار می‌باشد :



فایل ••••••••••.key در تصویر زیر قابل مشاهده می‌باشد :



همانطور که اشاره شد این باج‌افزار به جز فایل‌هایی با پسوندهای مشخص، باقی فایل‌ها را رمزگذاری می‌نماید. در زیر لیست فایل‌هایی که توسط باج‌افزار رمزگذاری می‌شوند، قابل مشاهده می‌باشد :

.jnt, .1CD, .dt, .cf, .1c, .doc, .docx, .xls, .xlsx, .ppt, .pptx, .pst, .ost, .msg, .eml, .vsd, .vsdx, .txt, .csv, .rtf, .123, .wks, .wk1, .pdf, .dwg, .onetoc2, .snt, .jpeg, .jpg, .docb, .docm, .dot, .dotm, .dotx, .xlsm, .xlsb, .xlw, .xlt, .xlm, .xlc, .xltx, .xltm, .pptm, .pot, .pps, .ppsm, .ppsx, .ppam, .potx, .potm, .edb, .hwp, .602, .sxi, .sti, .sldx, .sldm, .vdi, .vmdk, .vmx, .gpg, .aes, .paq, .bz2,

.tbk, .bak, .tar, .tgz, .gz, .7z, .rar, .zip, .backup, .iso, .vcd, .bmp, .png, .gif, .raw, .cgm, .tif, .tiff, .nef, .psd, .ai, .svg, .djvu, .m4u, .m3u, .mid, .wma, .flv, .3g2, .mkv, .3gp, .mp4, .mov, .avi, .asf, .mpeg, .vob, .mpg, .wmv, .fla, .swf, .wav, .mp3, .sh, .class, .jar, .java, .rb, .asp, .php, .jsp, .brd, .sch, .dch, .dip, .p1, .vb, .vbs, .ps1, .bat, .cmd, .js, .asm, .pas, .cpp, .c, .cs, .suo, .sln, .idf, .mdf, .ibd, .myi, .myd, .frm, .odb, .odf, .db, .mdb, .accdb, .sql, .sqlitedb, .sqlite3, .asc, .lay6, .lay, .mml, .sxm, .otg, .odg, .uop, .std, .sxd, .otp, .odp, .wb2, .slk, .dif, .stc, .sxc, .ots, .ods, .3dm, .max, .3ds, .uot, .stw, .sxw, .ott, .odt, .pem, .p12, .csr, .crt, .pfx, .der

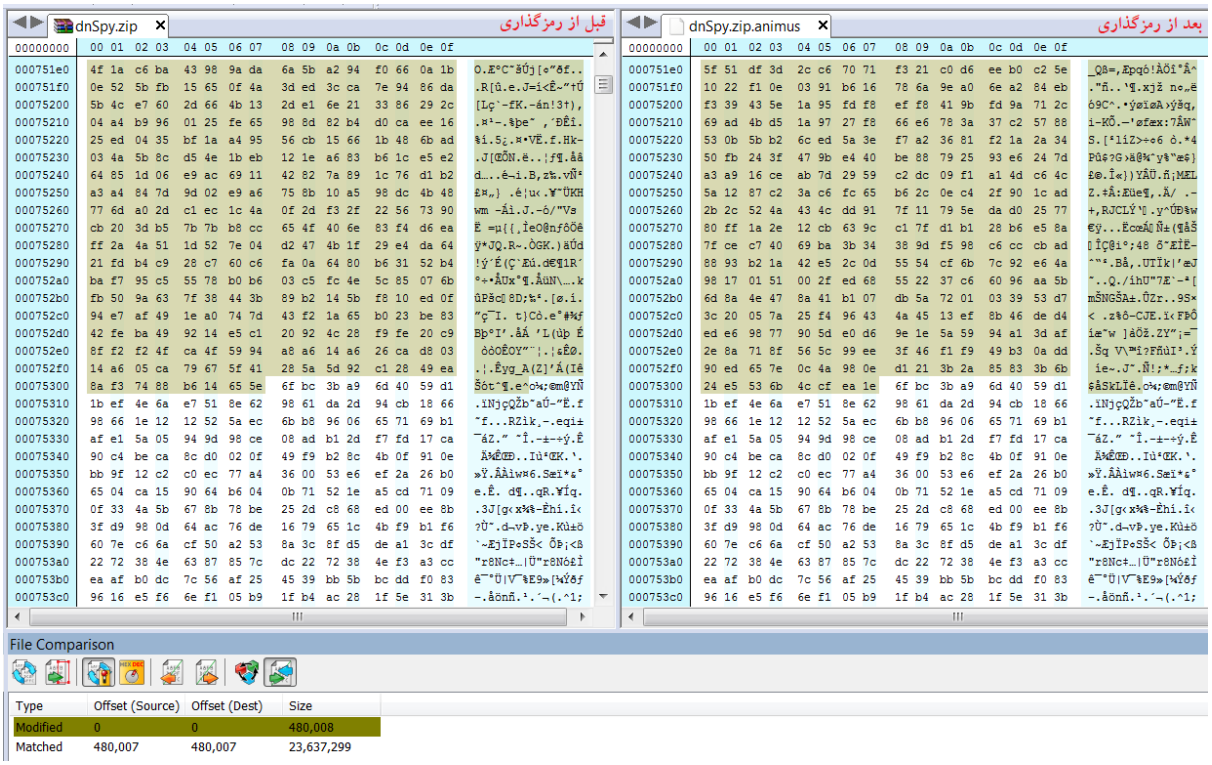
طبق مشاهدات صورت گرفته، هنگام اجرای باج‌افزار AnimusLocker به طور میانگین از ۳۵ الی ٤٥ درصد ظرفیت CPU، و ۲۰ درصد ظرفیت حافظه (RAM) استفاده می‌گردد.

بر اساس بررسی‌های انجام شده اکثر آنتی‌ویروس‌های معتبر، این باج‌افزار را به عنوان یک تروجان شناسایی نموده‌اند. لذا احتمال نفوذ باج‌افزار به سیستم از راه‌های متداول از جمله هرزنامه‌ها وجود دارد.

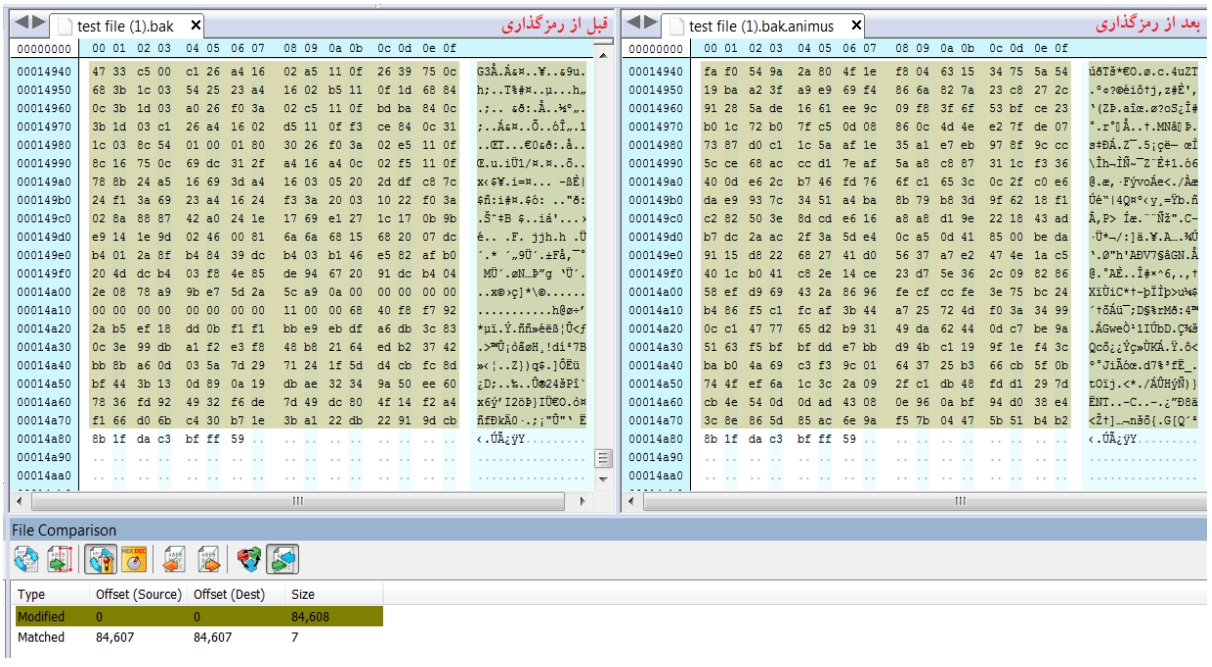
تحلیل ایستا:

پس از تحلیل کد باج‌افزار AnimusLocker به نتایج زیر دست پیدا کردیم.

طبق بررسی‌هایی که بر روی فایل‌های مختلف قبل و بعد از رمزگذاری انجام دادیم شاهد این بودیم که باج‌افزار AnimusLocker ساختار فایل‌ها را پس از رمزگذاری به طور کامل تغییر نمی‌دهد و با توجه به حجم فایل‌ها رفتار متفاوتی از خود نشان می‌دهد، به این صورت که تقریبا ۹۸ درصد ساختار فایل‌هایی که حجم آن‌ها کم‌تر از ۵۰۰ کیلوبایت می‌باشد را تغییر می‌دهد، اما فایل‌هایی که حجم آن‌ها بیشتر از ۵۰۰ کیلوبایت می‌باشند، فقط ٤٨٠٠٠٨ بایت ابتدایی آن‌ها را تغییر می‌دهد. تصاویر زیر نمونه‌ای از تغییرات ساختار فایل‌ها را نشان می‌دهد :

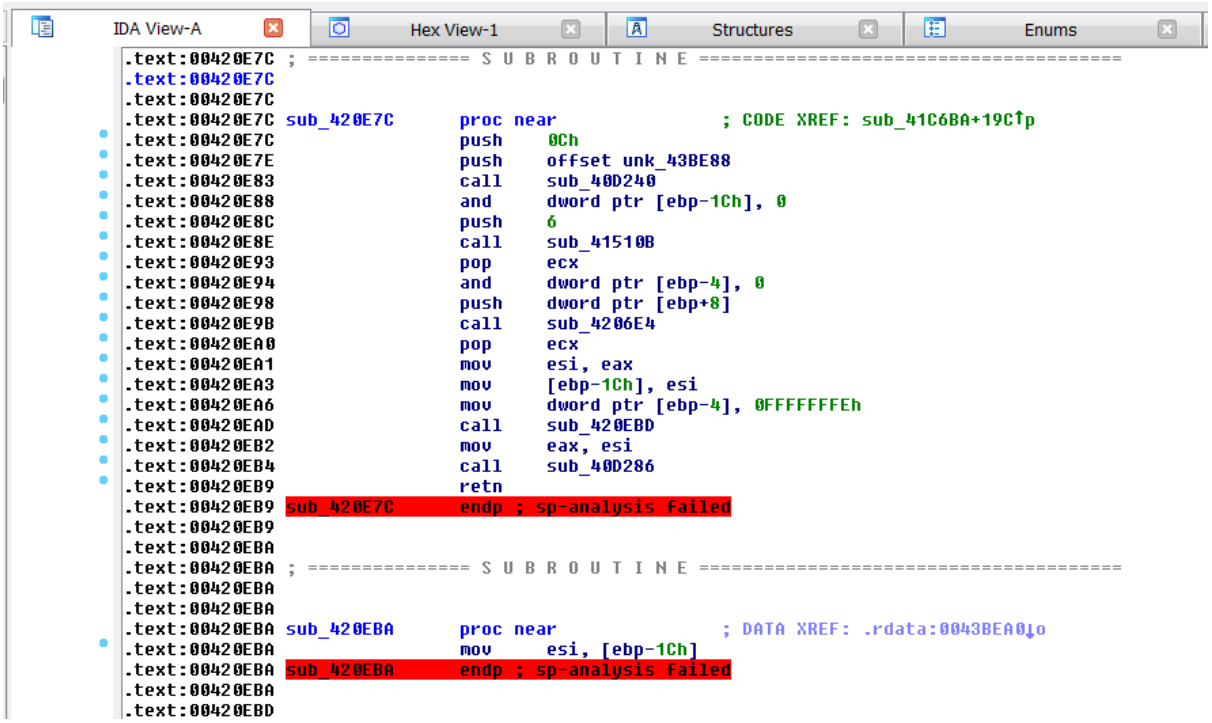


تصویر ۱: فایل با حجم بیشتر از ۵۰۰ کیلوبایت که ۴۸۰۰۰۸ بایت ابتدایی آن رمزگذاری شده است.

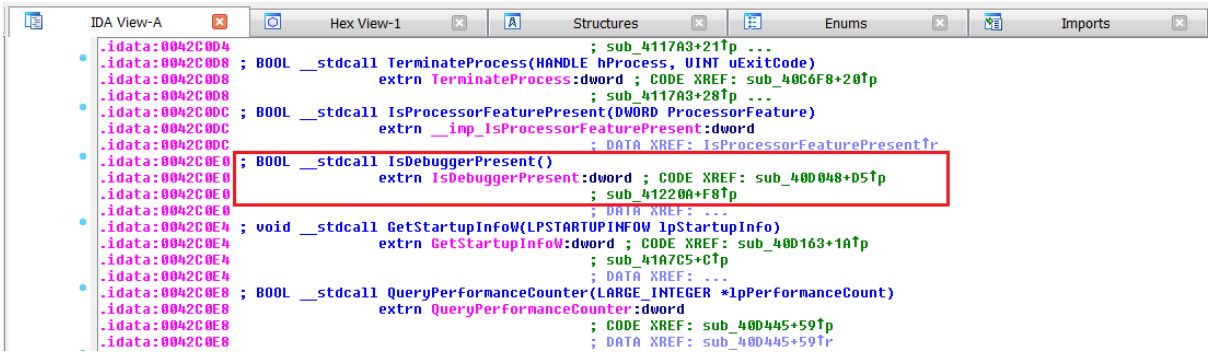


تصویر ۲: فایل با حجم کمتر از ۵۰۰ کیلوبایت که تقریبا ۹۸ درصد ساختار آن تغییر کرده است.

قطعه کد زیر مربوط به استفاده از روش‌های مختلف ضد دیس‌اسمبل جهت جلوگیری از بررسی و تحلیل‌های بیشتر توسط محققین می‌باشد :



تابع IsDebuggerPresent() که از توابع کتابخانه Kernel۳۲ می‌باشد برای جلوگیری از اجرای باج‌افزار در محیط‌های دیباگر استفاده می‌شود تا در هنگام تحلیل با ایجاد خطا در دیباگرها مانع فعالیت گردد. قطعه کد زیر مربوط به این فرایند می‌باشد :



در قطعه کد زیر با استفاده از تابع IsDebuggerPresent() اقدام به بررسی محیط دیباگر می‌کند. اگر نتیجه به دست آمده مثبت باشد (یعنی محیط اجرا دیباگر باشد)، با استفاده از تابع SetUnhandledExceptionFilter() باعث ایجاد خطا می‌شود و از ادامه‌ی فعالیت جلوگیری می‌نماید.

IDA View-A | Hex View-1 | Structures | Enums

```
.text:0040D086                  mov     [ebp+var_278], ecx
.text:0040D08C                  mov     [ebp+var_27C], edx
.text:0040D092                  mov     [ebp+var_280], ebx
.text:0040D098                  mov     [ebp+var_284], esi
.text:0040D09E                  mov     [ebp+var_288], edi
.text:0040D0A4                  mov     [ebp+var_25C], ss
.text:0040D0AB                  mov     [ebp+var_268], cs
.text:0040D0B2                  mov     [ebp+var_28C], ds
.text:0040D0B9                  mov     [ebp+var_290], es
.text:0040D0C0                  mov     [ebp+var_294], fs
.text:0040D0C7                  mov     [ebp+var_298], gs
.text:0040D0CE                  pushf
.text:0040D0CF                  pop     [ebp+var_264]
.text:0040D0D5                  mov     eax, [ebp+4]
.text:0040D0D8                  mov     [ebp+var_26C], eax
.text:0040D0DE                  lea     eax, [ebp+4]
.text:0040D0E1                  mov     [ebp+var_260], eax
.text:0040D0E7                  mov     [ebp+var_324], 10001h
.text:0040D0F1                  mov     eax, [eax-4]
.text:0040D0F4                  push    50h
.text:0040D0F6                  mov     [ebp+var_270], eax
.text:0040D0FC                  lea     eax, [ebp+var_58]
.text:0040D0FF                  push    esi
.text:0040D100                  push    eax
.text:0040D101                  call    sub_40F400
.text:0040D106                  mov     eax, [ebp+4]
.text:0040D109                  add     esp, 0Ch
.text:0040D10C                  mov     [ebp+var_58], 40000015h
.text:0040D113                  mov     [ebp+var_54], 1
.text:0040D11A                  mov     [ebp+var_4C], eax
.text:0040D11D                  call    ds:IsDebuggerPresent
.text:0040D123                  push    esi             ; lpTopLevelExceptionFilter
.text:0040D124                  lea     ebx, [eax-1]
.text:0040D127                  neg     ebx
.text:0040D129                  lea     eax, [ebp+var_58]
.text:0040D12C                  mov     [ebp+ExceptionInfo.ExceptionRecord], eax
.text:0040D12F                  lea     eax, [ebp+var_324]
.text:0040D135                  sbb     bl, bl
.text:0040D137                  mov     [ebp+ExceptionInfo.ContextRecord], eax
.text:0040D13A                  inc     bl
.text:0040D13C                  call    ds:SetUnhandledExceptionFilter
.text:0040D142                  lea     eax, [ebp+ExceptionInfo]
.text:0040D145                  push    eax             ; ExceptionInfo
.text:0040D146                  call    ds:UnhandledExceptionFilter
.text:0040D14C                  test    eax, eax
.text:0040D14E                  jnz     short loc_40D15D
.text:0040D150                  movzx   eax, bl
.text:0040D153                  neg     eax
.text:0040D155                  sbb     eax, eax
.text:0040D157                  and     dword_45CE78, eax
.text:0040D15D
.text:0040D15D loc_40D15D:                              ; CODE XREF: sub_40D048+106↑j
.text:0040D15D                  pop     esi
.text:0040D15E                  pop     ebx
.text:0040D15F                  mov     esp, ebp
.text:0040D161                  pop     ebp
.text:0040D162                  retn
.text:0040D162 sub_40D048       endp
0000C48C 0040D08C: sub_40D048+44
```

قطعه کد زیر مربوط به قرار دادن ۳ فایل TXT مربوط به پیغام باج‌خواهی در دایرکتوری‌های مختلف می‌باشد:

IDA View-A | Hex View-1 | Structures | Enums

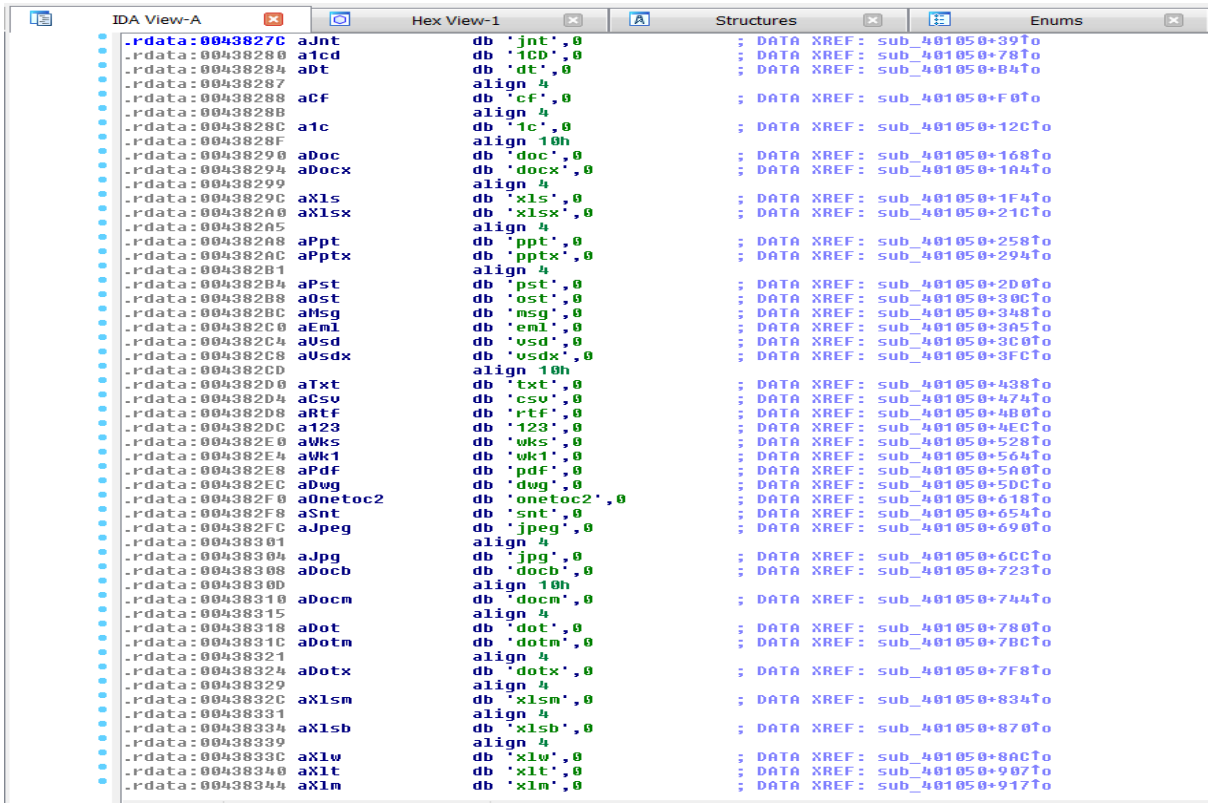
```
.text:00402510                push    ebp
.text:00402511                mov     ebp, esp
.text:00402513                push    0FFFFFFFFh
.text:00402515                push    offset sub_42A603
.text:0040251A                mov     eax, large fs:0
.text:00402520                push    eax
.text:00402521                sub     esp, 4Ch
.text:00402524                mov     eax, ___security_cookie
.text:00402529                xor     eax, ebp
.text:0040252B                push    eax
.text:0040252C                lea     eax, [ebp+var_C]
.text:0040252F                mov     large fs:0, eax
.text:00402535                lea     ecx, [ebp+var_58]
.text:00402538                call    sub_40A900
.text:0040253D                push    12h
.text:0040253F                push    offset aAnimus_restore ; "ANIMUS_RESTORE.txt"
.text:00402544                lea     ecx, [ebp+var_58]
.text:00402547                mov     [ebp+var_44], 0Fh
.text:0040254E                mov     [ebp+var_48], 0
.text:00402555                mov     [ebp+var_58], 0
.text:00402559                call    sub_408A60
.text:0040255E                lea     ecx, [ebp+var_40]
.text:00402561                mov     [ebp+var_4], 0
.text:00402568                call    sub_40A900
.text:0040256D                push    13h
.text:0040256F                push    offset aAnimus_resto_0 ; "ANIMUS_RESTORE2.txt"
.text:00402574                lea     ecx, [ebp+var_40]
.text:00402577                mov     [ebp+var_2C], 0Fh
.text:0040257E                mov     [ebp+var_30], 0
.text:00402585                mov     [ebp+var_40], 0
.text:00402589                call    sub_408A60
.text:0040258E                lea     ecx, [ebp+var_28]
.text:00402591                mov     byte ptr [ebp+var_4], 1
.text:00402595                call    sub_40A900
.text:0040259A                push    13h
.text:0040259C                push    offset aAnimus_resto_1 ; "ANIMUS_RESTORE3.txt"
.text:004025A1                lea     ecx, [ebp+var_28]
.text:004025A4                mov     [ebp+var_14], 0Fh
.text:004025AB                mov     [ebp+var_18], 0
.text:004025B2                mov     [ebp+var_28], 0
.text:004025B6                call    sub_408A60
.text:004025BB                push    ecx
.text:004025BC                mov     ecx, offset dword_45D6C4
.text:004025C1                mov     [ebp+var_4], 2
.text:004025C8                call    sub_40AE90
.text:004025CD                mov     byte ptr [ebp+var_10], 0
.text:004025D1                lea     eax, [ebp+var_10]
.text:004025D4                push    [ebp+var_10]
.text:004025D7                mov     dword_45D6C4, 0
.text:004025E1                push    eax
.text:004025E2                lea     eax, [ebp+var_58]
.text:004025E5                mov     dword_45D6C8, 0
.text:004025EF                push    eax
.text:004025F0                mov     dword_45D6CC, 0

00001910 00402510: sub_402510
```

تصاویر زیر مربوط به فرایند بررسی فایل‌ها جهت رمزگذاری و لیست فایل‌های هدف می‌باشد :

IDA View-A | Hex View-1 | Structures

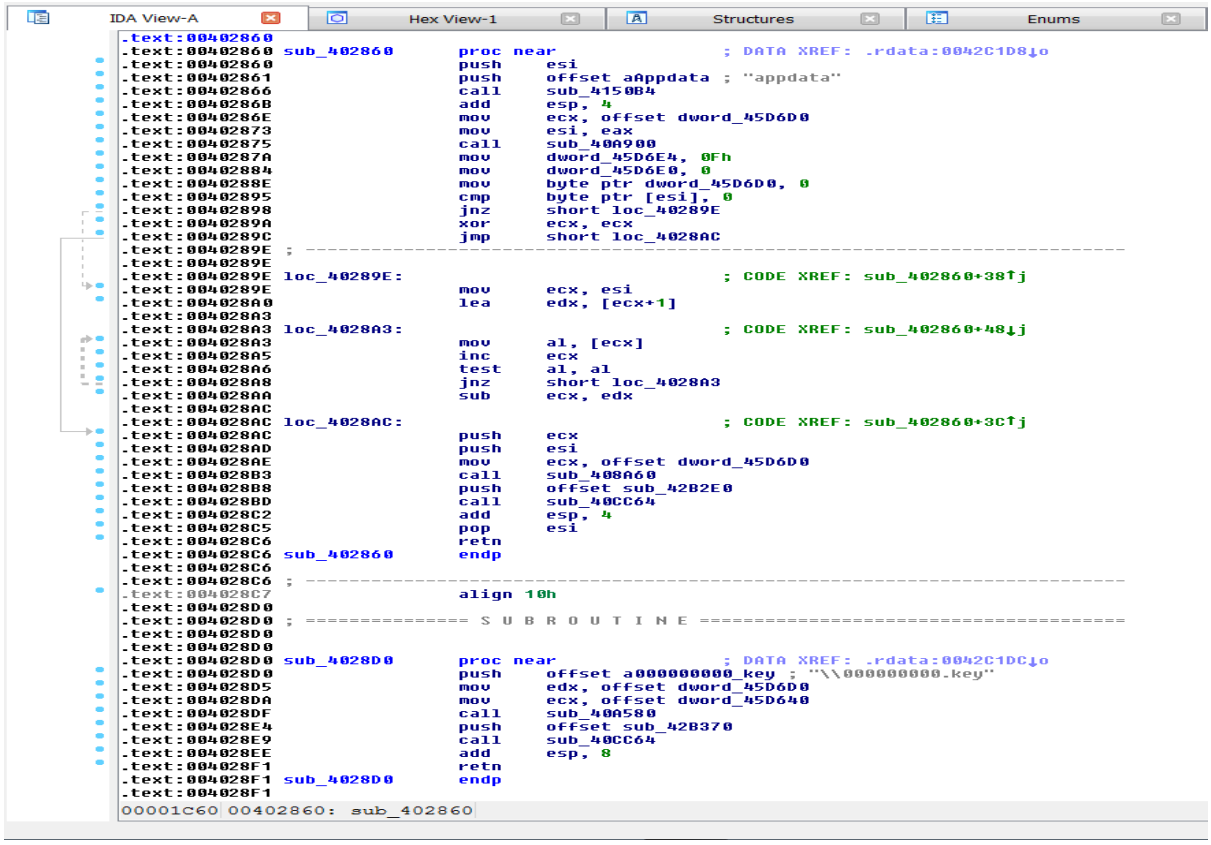
```
.text:00401050                push    ebp
.text:00401051                mov     ebp, esp
.text:00401053                push    0FFFFFFFFh
.text:00401055                push    offset sub_42A5C2
.text:0040105A                mov     eax, large fs:0
.text:00401060                push    eax
.text:00401061                mov     eax, 1128h
.text:00401066                call    sub_40D5B0
.text:0040106B                mov     eax, ___security_cookie
.text:00401070                xor     eax, ebp
.text:00401072                push    eax
.text:00401073                lea     eax, [ebp+var_C]
.text:00401076                mov     large fs:0, eax
.text:0040107C                lea     ecx, [ebp+var_1134]
.text:00401082                call    sub_40A900
.text:00401087                push    3
.text:00401089                push    offset aJnt       ; "jnt"
.text:0040108E                lea     ecx, [ebp+var_1134]
.text:00401094                mov     [ebp+var_1120], 0Fh
.text:0040109E                mov     [ebp+var_1124], 0
.text:004010A8                mov     [ebp+var_1134], 0
.text:004010AF                call    sub_408A60
.text:004010B4                lea     ecx, [ebp+var_111C]
.text:004010BA                mov     [ebp+var_4], 0
.text:004010C1                call    sub_40A900
.text:004010C6                push    3
.text:004010C8                push    offset a1cd       ; "1CD"
.text:004010CD                lea     ecx, [ebp+var_111C]
.text:004010D3                mov     [ebp+var_1108], 0Fh
.text:004010DD                mov     [ebp+var_110C], 0
.text:004010E7                mov     [ebp+var_111C], 0
.text:004010EE                call    sub_408A60
.text:004010F3                lea     ecx, [ebp+var_1104]
.text:004010F9                mov     byte ptr [ebp+var_4], 1
.text:004010FD                call    sub_40A900
.text:00401102                push    2
.text:00401104                push    offset aDt        ; "dt"
.text:00401109                lea     ecx, [ebp+var_1104]
.text:0040110F                mov     [ebp+var_10F0], 0Fh
.text:00401119                mov     [ebp+var_10F4], 0
.text:00401123                mov     [ebp+var_1104], 0
.text:0040112A                call    sub_408A60
.text:0040112F                lea     ecx, [ebp+var_10EC]
.text:00401135                mov     byte ptr [ebp+var_4], 2
.text:00401139                call    sub_40A900
.text:0040113E                push    2
.text:00401140                push    offset aCf        ; "cf"
.text:00401145                lea     ecx, [ebp+var_10EC]
.text:0040114B                mov     [ebp+var_10D8], 0Fh
.text:00401155                mov     [ebp+var_10DC], 0
.text:0040115F                mov     [ebp+var_10EC], 0
.text:00401166                call    sub_408A60
.text:0040116B                lea     ecx, [ebp+var_10D4]
.text:00401171                mov     byte ptr [ebp+var_4], 3

00000450 00401050: sub_401050
```

تصویر ۱

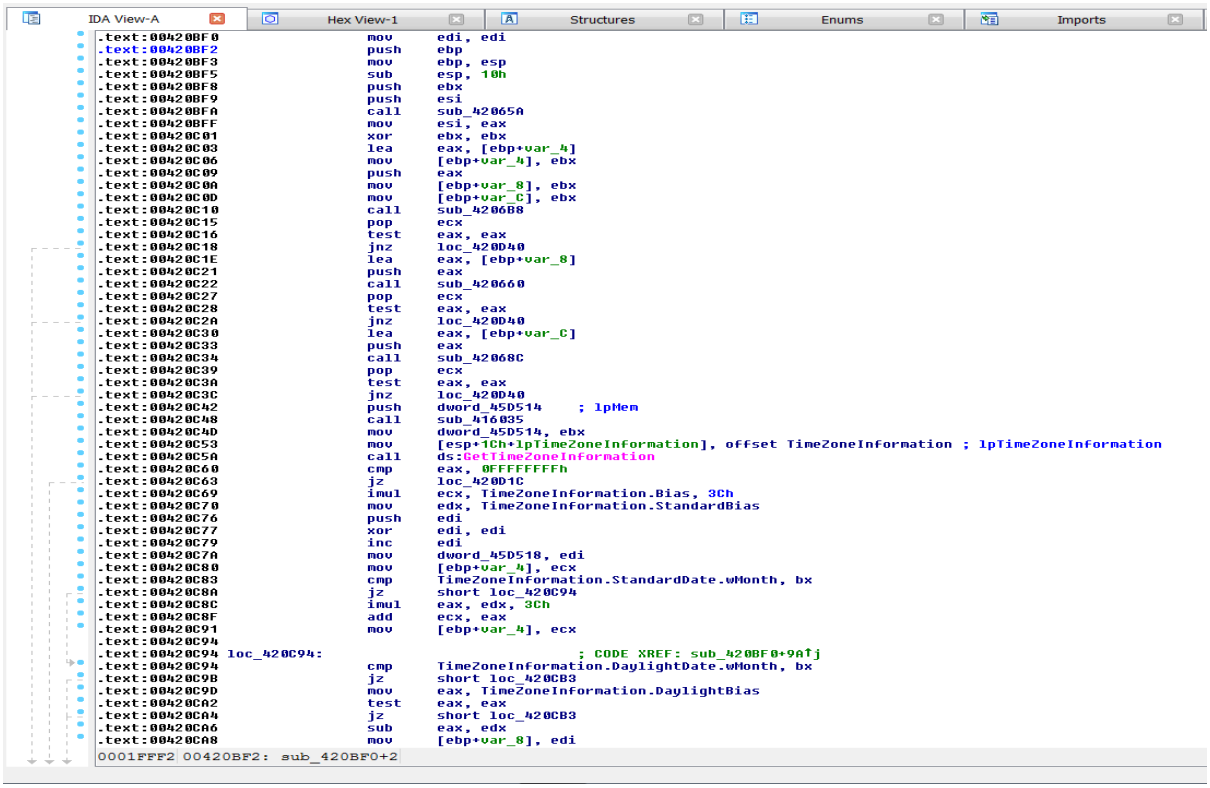


تصویر ۲: بخشی از فایل‌های مورد هدف باج‌افزار

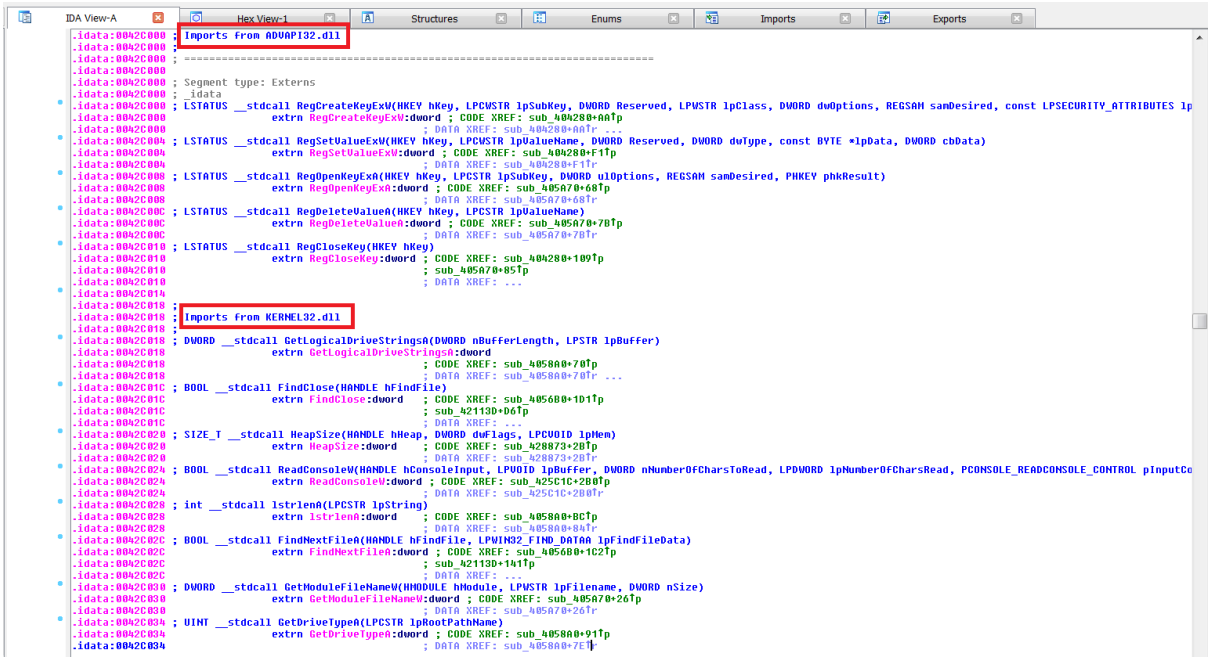
قطعه کد زیر مربوط به ایجاد فایل ••••••••••.key و قرار دادن آن در دایرکتوری %appdata% می‌باشد :



قطعه کد زیر مربوط به بررسی منطقه زمانی کاربران می‌باشد و به نظر می‌رسد باج‌افزار از آن برای هدف قرار دادن کاربرانی خاص استفاده می‌کند :



این باج‌افزار از کتابخانه‌های ویندوزی به همراه توابعی از هرکدام از کتابخانه‌ها استفاده می‌کند، در تصویر، استفاده از این کتابخانه‌ها به خوبی قابل مشاهده است، همچنین لیست کامل این کتابخانه‌ها به همراه توابع مورد استفاده نیز در ادامه‌ی متن آمده است.



ADVAPI۳۲.dll	KERNEL۳۲.dll	KERNEL۳۲.dll	KERNEL۳۲.dll	KERNEL۳۲.dll
RegCreateKeyExW	GetStdHandle	FindClose	TlsAlloc	HeapSize
RegOpenKeyExA	GetDriveTypeW	TlsGetValue	FlushFileBuffers	RaiseException
RegDeleteValueA	FileTimeToSystemTime	GetFullPathNameW	RtlUnwind	TlsFree
RegCloseKey	GetDriveTypeA	SetLastError	FreeLibrary	ReadFile
RegSetValueExW	SetEndOfFile	PeekNamedPipe	GetStartupInfoW	CloseHandle
	EncodePointer	IsDebuggerPresent	GetUserDefaultLCID	IsValidCodePage
	SystemTimeToTzSpecificLocal	HeapAlloc	GetProcessHeap	ResetEvent
	Time	GetModuleFileNameA	CompareStringW	FindNextFileA
	DeleteCriticalSection	EnumSystemLocalesW	FindFirstFileExA	IsValidLocale
	GetCurrentProcess	LoadLibraryExW	FindFirstFileA	GetProcAddress
	GetConsoleMode	MultiByteToWideChar	ExitProcess	CreateEventW
	UnhandledExceptionFilter	SetFilePointerEx	LeaveCriticalSection	CreateFileW
	FreeEnvironmentStringsW	MoveFileExW	GetLastError	GetFileType
	InitializeSListHead	DecodePointer	LCMapStringW	TlsSetValue
	GetLocaleInfoW	TerminateProcess	lstrlenA	SetStdHandle
	GetLogicalDriveStringsA	ReadConsoleW	GetConsoleCP	GetCPInfo
	GetTimeZoneInformation	GetCurrentThreadId	WaitForSingleObjectEx	WriteFile
	QueryPerformanceCounter	WriteConsoleW	GetCurrentDirectoryW	
	GetModuleHandleExW	InitializeCriticalSection	GetCurrentProcessId	
	SetEnvironmentVariableA	AndSpinCount	GetCommandLineW	
	IsProcessorFeaturePresent	HeapFree	WideCharToMultiByte	
	SetUnhandledExceptionFilter	EnterCriticalSection	GetSystemTimeAsFileTime	
	GetModuleFileNameW	SetEvent	HeapReAlloc	
	GetEnvironmentStringsW	GetModuleHandleW	GetStringTypeW	
	GetACP	GetCommandLineA	GetOEMCP	

بر اساس بررسی‌های صورت گرفته، این باج‌افزار پس از اجرا فقط یک فرایند ایجاد می‌کند :

- Ransom.exe

پس از خاتمه فرایند مربوط به باج‌افزار فرایند notepad.exe ایجاد می‌شود و پیغام باج‌خواهی به نمایش در می‌آید.

کلیدهای رجیستری زیر توسط باج‌افزار در سیستم باز می‌شوند :

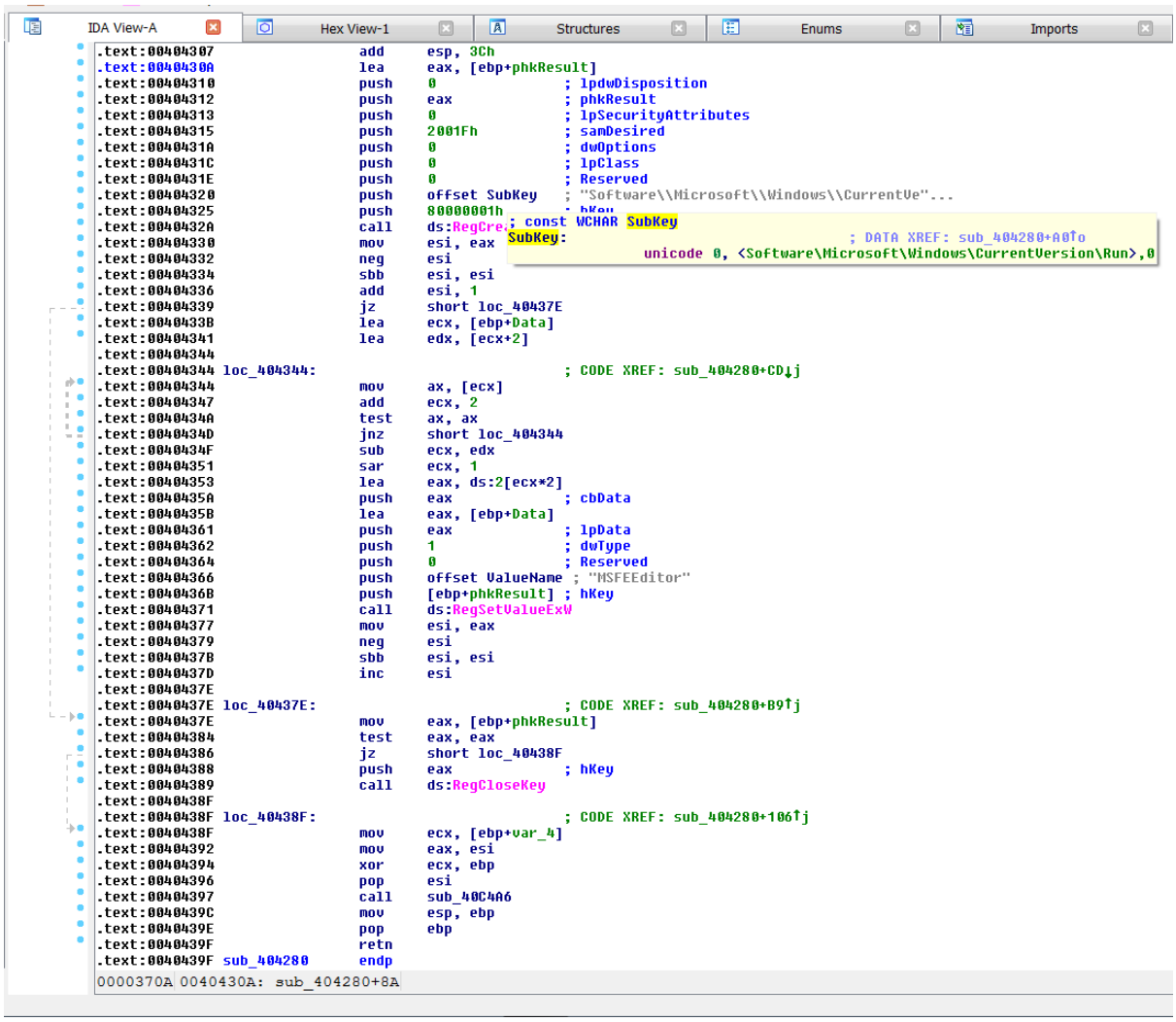
۱. *\Registry\Machine\Software\Microsoft\Windows NT\CurrentVersion\ImageFileExecutionOptions\996E.exe*
۲. *\Registry\MACHINE\System\CurrentControlSet\Control\SafeBoot\Option*
۳. *\Registry\Machine\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers*
٤. *\REGISTRY\MACHINE\SOFTWARE\Policies\Microsoft\Windows\Safer\CodeIdentifiers\TransparentEnabled*
٥. *\REGISTRY\USER\S-1-5-21-1482476501-1645522239-1417001333-500\Software\Policies\Microsoft\Windows\Safer\CodeIdentifiers*

طبق بررسی‌های انجام شده مهاجمین از کلید شماره ۱ جهت قابلیت توسعه دادن به باج‌افزار استفاده نموده‌اند و کلید شماره ۳ نیز جهت پیاده سازی سیاست‌های محدودیت نرم‌افزار استفاده شده است.

کلید رجیستری زیر توسط باج‌افزار در سیستم تنظیم می‌شود :

```
\REGISTRY\USER\S-1-5-21-1482476501-1645522239-1417001333-500\Software\Microsoft\Windows\CurrentVersion\Run\MSFEEditor
```

قطعه کد زیر مربوط به تنظیم شدن این کلید می‌باشد که برای اجرای دائمی و در هر بار اجرای سیستم از آن استفاده می‌شود و نام فایل مورد استفاده که در قسمت Run رجیستری ثبت می‌شود MSFEEditor می‌باشد :



تحلیل ترافیک شبکه :

پس از بررسی ترافیک شبکه، متوجه هیچ گونه درخواست DNS و تلاش برای برقراری ارتباط با میزبان در نقطه‌ی جغرافیایی خاص توسط باج‌افزار AnimusLocker نشدیم.

خروجی سامانه VirusTotal :

در حال حاضر تعداد ۵۰ مورد از ٦٧ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این باج افزار بوده وآن را حذف یا غیرفعال می‌کنند.

Ad-Aware	Trojan.GenericKD.31018237	AegisLab	W32.Troj.Ransom.Filecoder!c
ALYac	Trojan.Ransom.AnimusLocker	Antiy-AVL	Trojan[Dropper]/Win32.Dorgam
Arcabit	Trojan.Generic.D1D94CFD	Avast	Win32:Malware-gen
AVG	Win32:Malware-gen	Baidu	Win32.Trojan.WisdomEyes.16070401....
BitDefender	Trojan.GenericKD.31018237	CAT-QuickHeal	Trojan.Genasom
ClamAV	Win.Trojan.Agent-6595906-0	Comodo	TrojWare.Win32.Filecoder.~NNP
CrowdStrike Falcon	malicious_confidence_100% (W)	Cylance	Unsafe
Cyren	W32/Trojan.YKJD-1659	DrWeb	Trojan.Encoder.25651
Emsisoft	Trojan.GenericKD.31018237 (B)	Endgame	malicious (moderate confidence)
eScan	Trojan.GenericKD.31018237	ESET-NOD32	a variant of Win32/Filecoder.NNP
F-Secure	Trojan.GenericKD.31018237	Fortinet	W32/Filecoder.NNP!tr
GData	Win32.Trojan-Ransom.Filecoder.P@gen	Ikarus	Trojan-Ransom.FileCoder
Jiangmin	TrojanDropper.Dorgam.ry	K7AntiVirus	Trojan (005173491)
K7GW	Trojan (005173491)	Kaspersky	Trojan-Dropper.Win32.Dorgam.xiy
Malwarebytes	Ransom.FileCryptor	MAX	malware (ai score=94)
McAfee	Generic.dvr	McAfee-GW-Edition	BehavesLike.Win32.PUPXBE.fc
Microsoft	Trojan:Win32/Occamy.C	NANO-Antivirus	Trojan.Win32.Dorgam.feqpgk
Palo Alto Networks	generic.ml	Panda	Trj/GdSda.A
Qihoo-360	HEUR/QVM10.2.2661.Malware.Gen	Rising	Trojan.Filecoder!8.68 (CLOUD)
SentinelOne	static engine - malicious	Sophos AV	Mal/Generic-S
Sophos ML	heuristic	Symantec	Trojan Horse
Tencent	Win32.Trojan.Filecoder.Tbii	TrendMicro	Ransom_ANIMUS.THGOBAH
TrendMicro-HouseCall	Ransom_ANIMUS.THGOBAH	VBA32	TrojanDropper.Dorgam
Webroot	W32.Trojan.Gen	Yandex	Trojan.DR.Dorgam!tdFlMgVG6fk
Zillya	Dropper.Dorgam.Win32.951	ZoneAlarm	Trojan-Dropper.Win32.Dorgam.xiy

خروجی سامانه ویروس‌کاو مرکز ماهر :

در حال حاضر تعداد ۸ مورد از ۱۱ آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این باج افزار بوده وآن را حذف یا غیرفعال می‌کنند.

نتیجه اسکن AnimusLocker.exe

آنتی‌ویروس	نسخه آنتی‌ویروس	نتیجه اسکن	
پادویش	2.3.190.2675	✔	Clean
sophos	9.14.2	✔	Clean
f_secure	11.00	ii	Dangerous: Trojan.GenericKD.31018237
kaspersky	5.5	ii	Dangerous: Trojan-Dropper.Win32.Dorgam.Xiy
eset	4.5.3.38121	ii	Dangerous: Win32/Filecoder.NNP
drweb	11.0.1.1607061217	ii	Dangerous: Trojan.Encoder.25651
clam_av	0.99.2	ii	Dangerous: Win.Trojan.Agent-6595906-0
comodo	1.1.268025.1	ii	Dangerous: Malware
bitdefender	11.0.1.18	ii	Dangerous: Trojan.GenericKD.31018237
avast	2.1.2	✔	Clean
symantec	7.9.0.30	ii	Dangerous: Trojan Horse