باسمه تعالی

تحلیل فنی باج افزار PainLocker

مقدمه :

مشاهده و رصد فضای سایبری در زمینه باج‌افزار، از شروع فعالیت نمونه جدیدی از خانواده‌ی Everbe به نام PainLocker خبر می‌دهد که پس از رمزگذاری فایل‌ها، به انتهای آن‌ها عبارت [Pain@cock.lu].pain. را اضافه می‌کند. بررسی‌ها نشان می‌دهد که فعالیت این باج‌افزار در تاریخ ۲۵ می ۲۰۱۸ میلادی شروع شده و به نظر می‌رسد تمرکز آن بیشتر بر روی کاربران انگلیسی زبان می‌باشد. این باج‌افزار به جز فایل‌های موجود در برخی از دایرکتوری‌های سیستم قربانی که در ادامه به آن‌ها اشاره خواهیم نمود، تمامی فایل‌های موجود بر روی سیستم قربانی را رمزگذاری می‌کند. طبق بررسی‌های انجام شده در حال حاضر، باج‌افزار EVEREST LOCKER آخرین عضو خانواده‌ی Everbe می‌باشد و ترتیب انتشار باج‌افزارهای عضو این خانواده به صورت زیر می‌باشد :

Everbe> Embrace> PainLocker> eV۳rbe> EvilLocker> HYENA> thunder> divine> EvilLocker۲>

NOT_OPEN LOCKER v۱> NOT_OPEN LOCKER v۲> EVEREST LOCKER

طبق بررسی‌های صورت گرفته فایل‌های رمزگذاری شده توسط این نسخه از خانواده‌ی Everbe، قابل رمزگشایی می‌باشند و قربانیان می‌توانند با استفاده از ابزار رمزگشایی منتشر شده، به راحتی فایل‌های خود را رمزگشایی نمایند.

مشخصات فایل اجرایی :

نام فایل	PainLocker.exe
MD۵	b۰٤۷a٤ab۱۷۲٦fe۷٤۱٤٤۸٤٤۹۳c٦۰d۵۸۷۰
SHA-۱	e۹a۵٤٤bac٤۳۰۲٦۱۰۰۸af۹fbf۸۰ab٦fba٦۳۸۵c۲cb
SHA-۲۵٦	d۹ea۳۷b٤۳b۵۹۷۳٤۰d۱٤۳۰f۳b۹۷cb۷۲d۸٤ef۸۸aeb۵۵۹acf۸٦۸dfd۰a۷۳de٤f٤۵f۸
اندازه فایل	۳۰.۵ KB
کامپایلر	Microsoft visual c++ ۸

فایل اجرایی این باج‌افزار دارای سه بخش است :

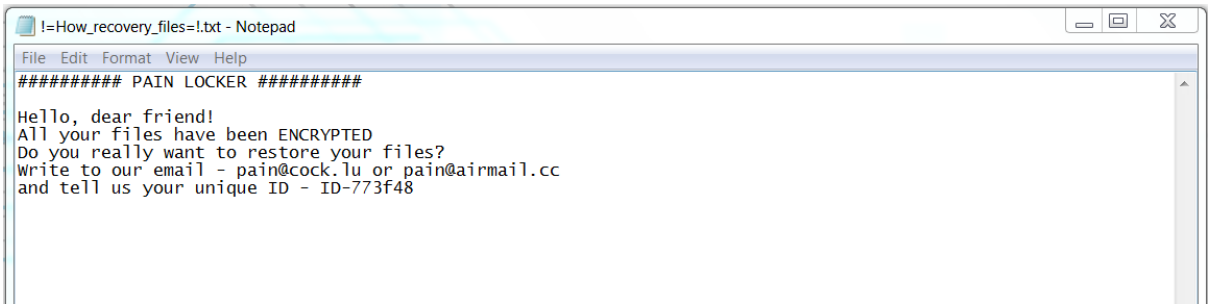
نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام

UPX۰	0	4096	61440	0
UPX۱	7.86	65536	32768	29184
.rsrc	4.74	98304	4096	1024

تحلیل پویا :

برای بررسی عمیق‌تر باج‌افزار PainLocker، 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 بر خلاف برخی از نسخه‌های قبلی این خانواده که فقط بر روی سیستم‌هایی با ویندوز ٦٤ بیتی قابل اجاره بودند، بر روی سیستم‌هایی با ویندوز ۳۲ بیتی نیز قابل اجرا می‌باشد و پس از اجرا، از ادامه‌ی فعالیت برخی از فرایندها که در ادامه به آن‌ها اشاره خواهیم نمود، جلوگیری می‌کند. همچنین مشاهده گردید باج‌افزار با اجرای فرایند vssadmin.exe نسخه‌های shadowcopy را حذف می‌کند و پس از آن شروع به رمزگذاری فایل‌ها می‌کند. این باج‌افزار در طول اجرای خود یک فایل متنی تحت عنوان !_How_recovery_files_!.txt که محتوای آن شامل پیغام باج‌خواهی می‌باشد را بر روی Desktop و در دایرکتوری‌های مختلف ایجاد می‌کند و پس از پایان فرایند رمزگذاری فایل‌ها، فرایند مربوط به اجرای باج‌افزار پایان می‌یابد و فایل اجرایی آن نیز حذف می‌شود.

تصویر زیر پیغام باج‌خواهی باج افزار PainLocker را نشان می‌دهد.



بر اساس پیغام باج‌خواهی مهاجمین اعلام نموده‌اند که تمام فایل‌های قربانیان رمزگذاری شده و در صورتی که تمایل به رمزگشایی فایل‌های خود دارند بایستی از طریق یکی از آدرس ایمیل‌های pain@cock.lu و یا pain@airmail.cc با آن‌ها ارتباط برقرار نمایند و در ایمیل ارسالی کدشناسایی مربوط به خود را ذکر نمایند. پس از برقراری ارتباط به صورت ناشناس، پاسخی از سوی مهاجمین دریافت نکردیم.

طبق بررسی‌های انجام شده بر روی کدمنبع باج‌افزار PainLocker شاهد این بودیم که این باج‌افزار به جز فایل‌های موجود در دایرکتوری‌های زیر، باقی فایل‌های موجود در سیستم قربانیان را رمزگذاری می‌کند و به

علت رمزگذاری دایرکتوری‌های مربوط به نرم‌افزارهای نصب شده بر روی سیستم، هیچ یک از آن‌ها قابل استفاده نیستند.

Windows, Program files, Program files (x۸۶), System volume information

همچنین این باج‌افزار فایل‌های موجود در Recycle Bin را حذف می‌کند.

باج‌افزار PainLocker ساختار تمام فایل‌ها را به یک شکل تغییر نمی‌دهد و طبق بررسی‌های صورت گرفته ساختار فایل‌ها با پسوندهای زیر و موجود در دایرکتوری‌های اشاره شده زیر را به طور کامل تغییر می‌دهد :

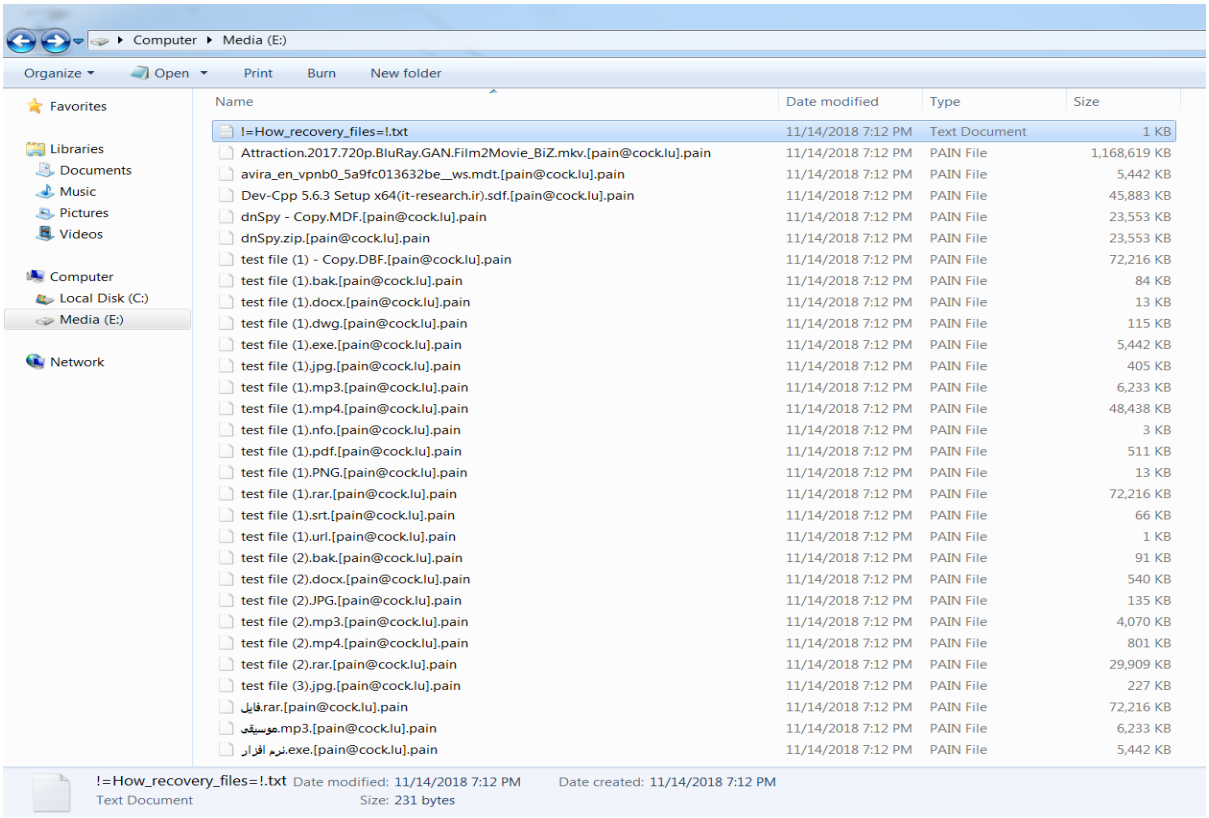
.sql, .mdf, .txt, .dbf, .ckp, .dacpac, .db۳, .dtxs, .mdt, .sdf, .MDF, .DBF

همانطور که قابل مشاهده است این پسوندها، مربوط به فایل‌های پایگاه داده می‌باشد و به نظر می‌رسد به علت تغییر یافتن تمام ساختار فایل‌های مربوطه، این فایل‌ها توسط نرم‌افزارهای بازیابی اطلاعات پایگاه داده قابل بازیابی نمی‌باشند.

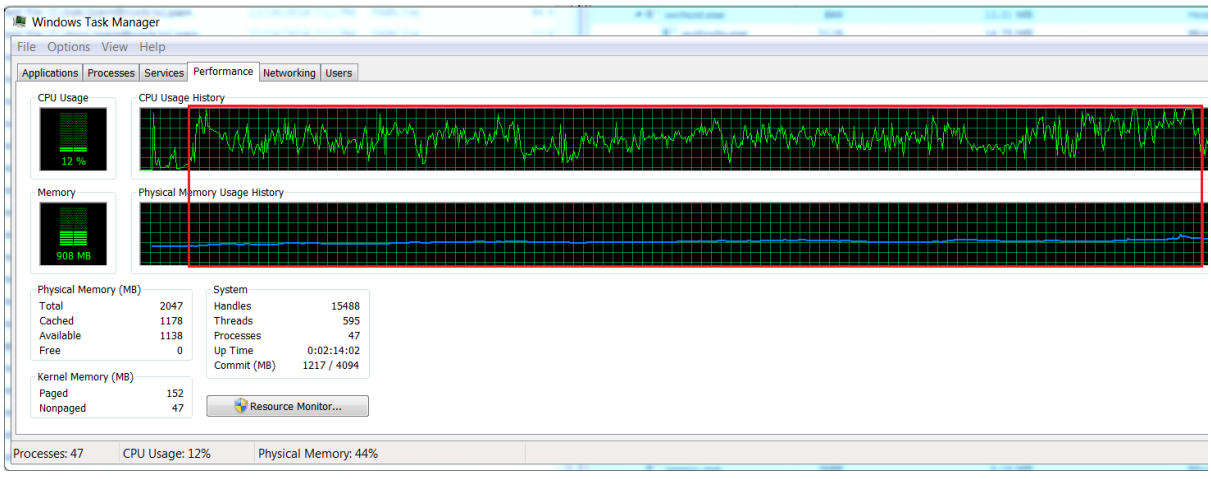
باج‌افزار PainLocker پس از اجرا از ادامه‌ی فعالیت برخی فرایندها جلوگیری کرده و همچنین مانع اجرای مجدد آن‌ها می‌شود. لیست کامل این فرایندها در ذیل قابل مشاهده می‌باشد :

sqlserv.exe, oracle.exe, ntdbsmgr.exe, sqlservr.exe, sqlwriter.exe, fdhost.exe, MsDtsSrvr.exe, msmdsrv.exe, ReportingServecesService.exe, fdlauncher.exe

این باج‌افزار در صورتی که عنوان فایل‌ها به زبان فارسی نیز باشد، آن‌ها را رمزگذاری می‌کند و پسوند فایل‌ها پس از رمزگذاری به [Pain@cock.lu].pain. تغییر پیدا می‌کند. تصویر زیر نشان‌دهنده فایل‌های رمزگذاری شده توسط این باج‌افزار می‌باشد :



طبق مشاهدات صورت گرفته، در صورت بالا بودن ظرفیت منابع سیستم قربانی، سرعت رمزگذاری فایل‌ها نیز بالاتر خواهد بود. هنگام اجرای باج‌افزار PainLocker شاهد بودیم که این باج‌افزار به طور میانگین از ۴۰ الی ۴۵ درصد ظرفیت CPU، و ۵ الی ۱۰ درصد ظرفیت حافظه (RAM) استفاده می‌کند. همچنین مدت زمان رمزگذاری فایل‌ها بستگی به حجم داده‌های موجود بر روی سیستم قربانیان دارد. به طور مثال طبق بررسی‌های صورت گرفته در محیط آزمایشگاه، مدت زمان لازم جهت رمزگذاری یک هارد دیسک با ظرفیت ۲۵ گیگابایت، ۱۲ دقیقه بود. تصویر زیر مربوط به نمودار مصرف منابع سیستم توسط باج‌افزار، از لحظه‌ی شروع تا انتهای فرایند رمزگذاری می‌باشد :

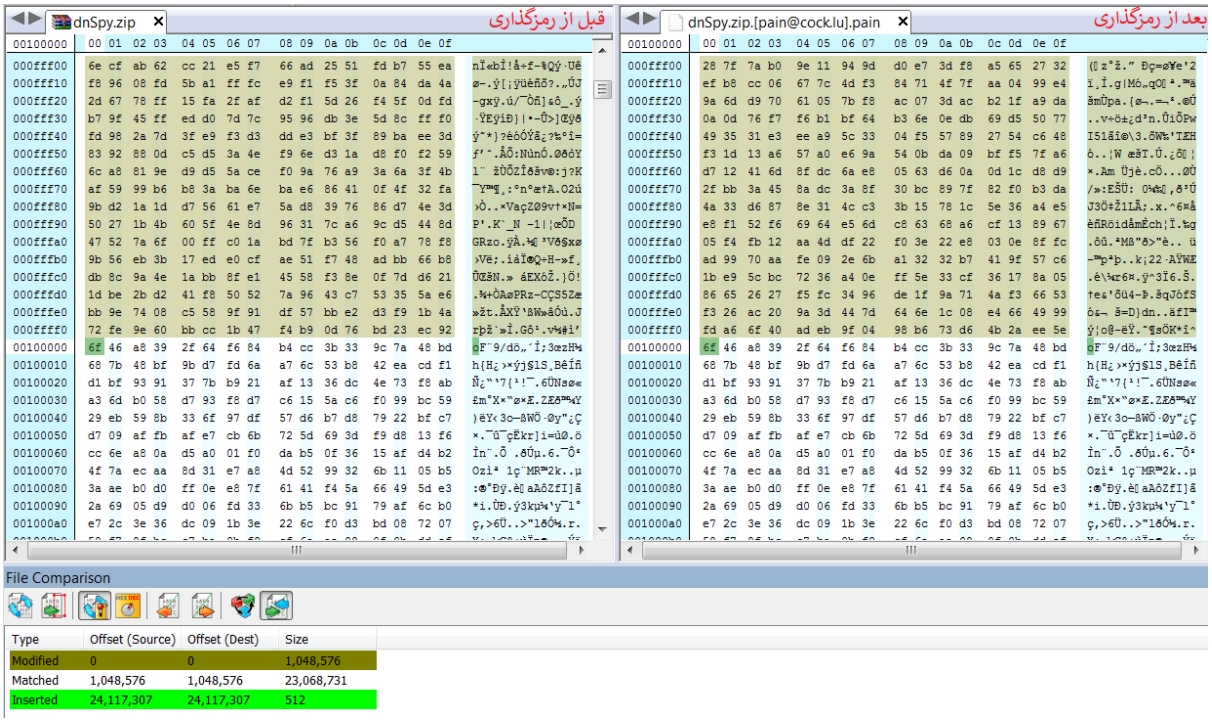


بر اساس مشاهدات صورت گرفته و سوابق نسخه‌های قبلی خانواده‌ی باج‌افزار Everbe، این نسخه نیز به طور معمول از طریق هک کردن کلمه عبور سرویس ریموت دسکتاپ (RDP) و همچنین هرزنامه ها منتشر می گردد. لذا به مدیران و راهبران شبکه در سازمان ها توصیه می گردد نسبت به امن سازی شبکه خصوصاً RDP اقدام نمایند و کاربران از باز نمودن هرگونه ایمیل حاوی پیوست مشکوک و بازدید از سایت‌های نامعتبر جداً خودداری نمایند.

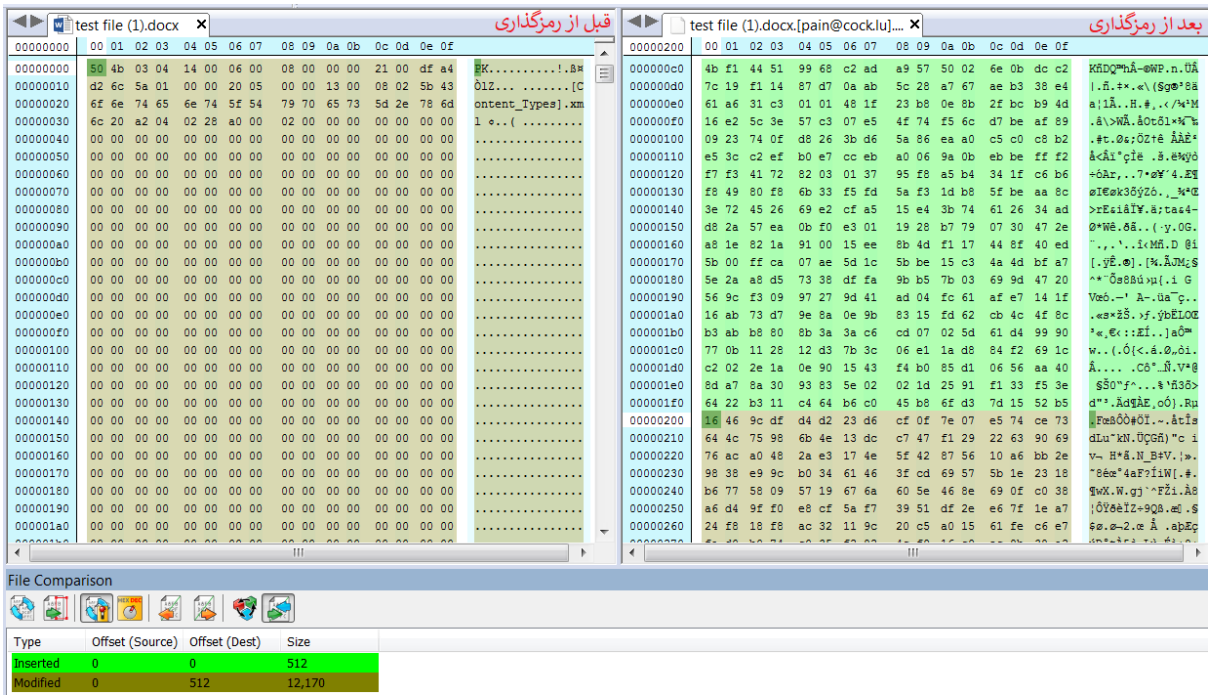
تحلیل ایستا:

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

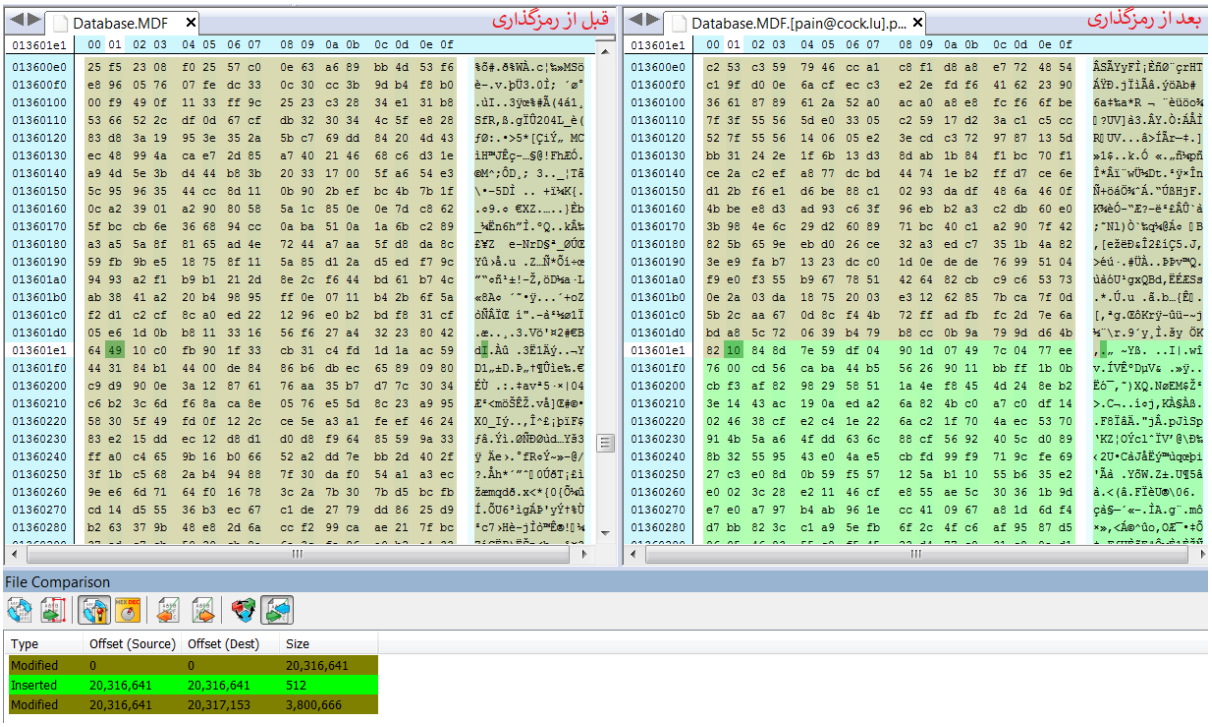
طبق بررسی‌هایی که بر روی فایل‌های مختلف قبل و بعد از رمزگذاری انجام دادیم شاهد این بودیم که باج‌افزار PainLocker ساختار فایل‌هایی را که حجم آن‌ها کمتر از ۱۰۴۸۵۷۶ بایت است و برخی از فایل‌ها با پسوند مشخص که در بخش تحلیل پویا به آن‌ها اشاره نمودیم را به طور کامل تغییر می‌دهد و فایل‌هایی که حجم آن‌ها از این مقدار بیشتر است، تنها ۱۰۴۸۵۷۶ بایت ابتدایی آن‌ها را تغییر می‌دهد. همچنین این باج‌افزار مقدار ۵۱۲ بایت را به ساختار تمامی فایل‌هایی که اشاره شد، پس از رمزگذاری اضافه می‌کند. تصاویر زیر نمونه‌ای از تغییرات ساختار فایل‌ها را نشان می‌دهد :



تصویر ۱: فایل با حجم بیشتر از ۱۰۴۸۵۷۶ بایت که ۱۰۴۸۵۷۶ بایت ابتدایی آن رمزگذاری شده است.

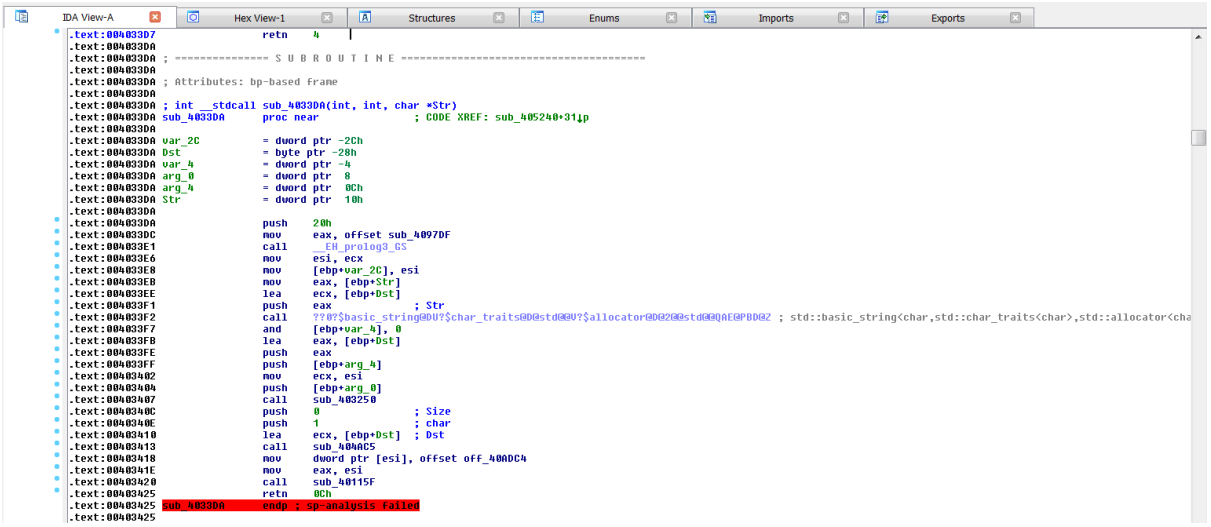


تصویر ۲: فایل با حجم کمتر از ۱۰۴۸۵۷۶ بایت که تمام ساختار آن تغییر کرده است.

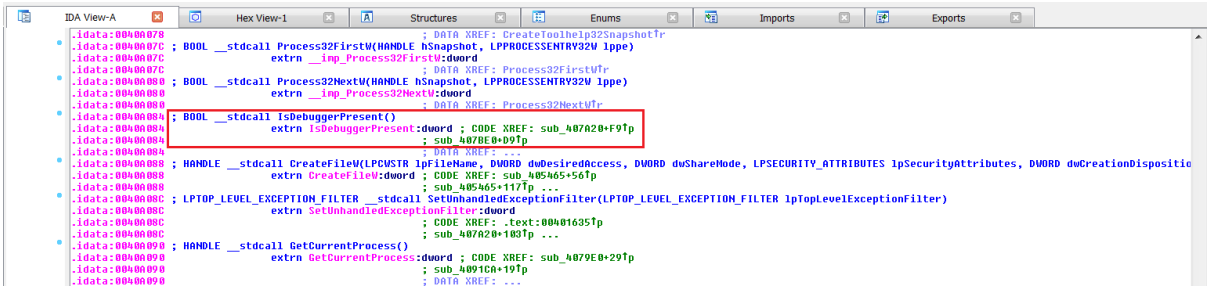


تصویر ۳: فایل با پسوندهای مشخص که ساختار آن به طور کامل تغییر نموده است.

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.

Sample_5b28bc5ec036f72574b3ceb6.c

```
7465    int v21; // [esp+108h] [ebp-228h]
7466    int v22; // [esp+10Ch] [ebp-224h]
7467    int *v23; // [esp+110h] [ebp-220h]
7468    int v24; // [esp+114h] [ebp-21Ch]
7469    int v25; // [esp+118h] [ebp-218h]
7470    __int16 v26; // [esp+11Ch] [ebp-214h]
7471    unsigned int v27; // [esp+120h] [ebp-210h]
7472    int *v28; // [esp+124h] [ebp-20Ch]
7473    __int16 v29; // [esp+128h] [ebp-208h]
7474    int savedregs; // [esp+330h] [ebp+0h]
7475    int retaddr; // [esp+334h] [ebp+4h]
7476
7477    if ( a3 != -1 )
7478      sub_407D00();
7479    memset(&Dst, 0, 0x50u);
7480    memset(&v13, 0, 0x2CCu);
7481    ExceptionInfo.ExceptionRecord = (PEXCEPTION_RECORD)&Dst;
7482    ExceptionInfo.ContextRecord = (PCONTEXT)&v13;
7483    v23 = &v13;
7484    v22 = v5;
7485    v21 = v6;
7486    v20 = a1;
7487    v19 = a3;
7488    v18 = a2;
7489    v29 = __SS__;
7490    v26 = __CS__;
7491    v17 = __DS__;
7492    v16 = __ES__;
7493    v15 = __FS__;
7494    v14 = __GS__;
7495    v7 = __readeflags();
7496    v27 = v7;
7497    v25 = retaddr;
7498    v28 = &retaddr;
7499    v13 = 65537;
7500    v24 = savedregs;
7501    Dst = a4;
7502    v11 = a5;
7503    v12 = retaddr;
7504    v8 = IsDebuggerPresent();
7505    SetUnhandledExceptionFilter(0);
7506    if ( !UnhandledExceptionFilter(&ExceptionInfo) && !v8 && a3 != -1 )
7507      sub_407D00();
7508  }
```

قطعه کد زیر مربوط به پیغام باج‌خواهی و کلید عمومی آن در کدمنبع باج‌افزار می‌باشد :

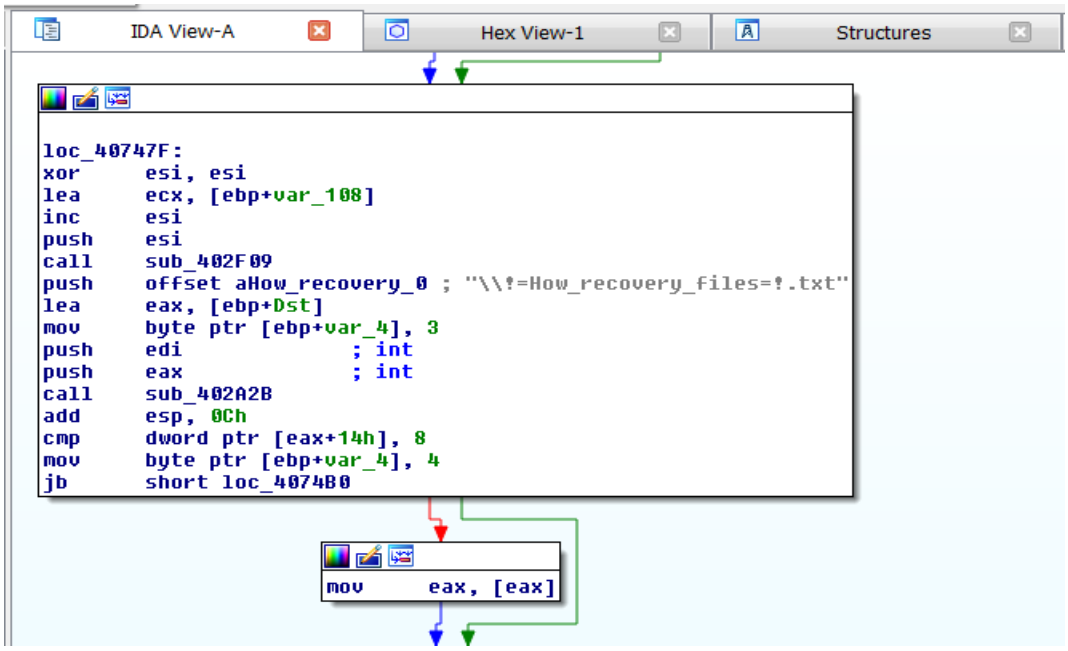
IDA View-A | Hex View-1 | Structures | Enums

```
.rdata:0040A820 aPainLockerHell db '########## PAIN LOCKER ##########',0Ah
.rdata:0040A820                                         ; DATA XREF: .text:00401FAB↑o
.rdata:0040A820                 db 0Ah
.rdata:0040A820                 db 'Hello, dear friend!',0Ah
.rdata:0040A820                 db 'All your files have been ENCRYPTED',0Ah
.rdata:0040A820                 db 'Do you really want to restore your files?',0Ah
.rdata:0040A820                 db 'Write to our email - pain@cock.lu or pain@airmail.cc',0Ah
.rdata:0040A820                 db 'and tell us your unique ID - ID-${CODE}',0Ah,0
.rdata:0040A902                 align 8
.rdata:0040A908 aBeginPublicKey db '-----BEGIN PUBLIC KEY-----',0Ah ; DATA XREF: .data:Src↓o
.rdata:0040A908                 db 'MIIBITANBgkqhkiG9w0BAQEFAAOCAQ4AMIIBCQKCAQBpljud8ZE16vvjtgezhDxx',0Ah
.rdata:0040A908                 db 'HFNB7y+zyv5p0O0Vr04c5+3f3BG+pIJSvxi0/YP6j/rbEFKVJdHYX+89BfNopRVa',0Ah
.rdata:0040A908                 db 'Sks9daf21YPn0n0/f1srlnpOF68PLPGH/rQ1IdZQ+G05+FVrDnE1SubHr12G5Q17',0Ah
.rdata:0040A908                 db 'JTkrwpsNOkq3FM1JMP6tjX8y7k9YPbHtv9HvZ+oxbpdD/MAeVVWaVj61PRaZ8T8Z',0Ah
.rdata:0040A908                 db 'GZTvUNiZJGF5InYt41MjRVFBo6zzlVKcR8pehJzu2YEMejsufNFMQ1gw4irzC2HB',0Ah
.rdata:0040A908                 db '5+BS9MOVM9NFZWEMKCY+jn9nGykdH7DoVgn7JIBV7M9fkBbhF4gc0BRotDGUPVbf',0Ah
.rdata:0040A908                 db 'AgMBAAE=',0Ah
.rdata:0040A908                 db '-----END PUBLIC KEY-----',0
```

مقدار کلید عمومی باج‌افزار جهت رمزگذاری فایل‌ها در زیر قابل مشاهده می‌باشد :

MIIBITANBgkqhkiG9w0BAQEFAAOCAQ4AMIIBCQKCAQBpljud8ZE16vvjtgezhDxxHFNB7y+zy
v5p0O0Vr04c5+3f3BG+pIJSvxi0/YP6j/rbEFKVJdHYX+89BfNopRVaSks9daf2IYPn0n0/f1srln
pOF68PLPGH/rQlIdZQ+G05+FVrDnE1SubHr12G5QI7JTkrwpsNOkq3fM1JMP6tjX8y7k9YPb
Htv9HvZ+oxbpdD/MAeVVWaVj6lPRaZ8T8ZGZTvUNiZJGF5InYt41MjRVFBo6zzlVKcR8pehJzu
2YEMejsufNFMQ1gw4irzC2HB5+BS9MOVM9NFZWEMKCY+jn9nGykdH7DoVgn7JIBV7M9fk
BbhF4gc0BRotDGVPVbfAgMBAAE=

قطعه کد زیر مربوط به تابع ایجاد فایل مربوط به پیغام باج‌خواهی می‌باشد :



قطعه کدهای زیر مربوط به فرایند رمزگذاری فایل‌ها می‌باشد که در این فرایند توابع CryptStringToBinaryA()، CryptImportPublicKeyInfo()، CryptEncrypt() و ... جهت انجام رمزگذاری فایل‌ها فراخوانی می‌شوند :

Sample_5b28bc5ec036f72574b3ceb6.c

```
3370    DWORD pdwDataLen; // [esp+24h] [ebp-1008h]
3371    BYTE pbBinary; // [esp+28h] [ebp-1004h]
3372    char Dst; // [esp+828h] [ebp-804h]
3373
3374    v2 = Src;
3375    v7 = (DWORD *)a2;
3376    v3 = strlen(::Src);
3377    memcpy(&Dst, ::Src, v3);
3378    pcbBinary = 2048;
3379    phProv = 0;
3380    phKey = 0;
3381    result = CryptStringToBinaryA(&Dst, 0, 0, &pbBinary, &pcbBinary, 0, 0);
3382    if ( result )
3383    {
3384      if ( !CryptDecodeObjectEx(1u, (LPCSTR)8, &pbBinary, pcbBinary, 0x8000u, 0, &pvStructInfo, &pcbStructInfo) )
3385        goto LABEL_13;
3386      if ( !CryptAcquireContextW(&phProv, 0, L"Microsoft Enhanced Cryptographic Provider v1.0", 1u, 0xF0000000) )
3387        goto LABEL_13;
3388      if ( !CryptImportPublicKeyInfo(phProv, 1u, pvStructInfo, &phKey) )
3389        goto LABEL_13;
3390      Size = *((_DWORD *)Src + 4);
3391      pdwDataLen = Size;
3392      if ( !CryptEncrypt(phKey, 0, 1, 0, 0, &pdwDataLen, Size) )
3393        goto LABEL_13;
3394      v5 = operator new[](pdwDataLen);
3395      memset(v5, 0, pdwDataLen);
3396      if ( *((_DWORD *)Src + 5) >= 0x10u )
3397        v2 = *(void **)Src;
3398      memcpy(v5, v2, Size);
3399      if ( CryptEncrypt(phKey, 0, 1, 0, (BYTE *)v5, &Size, pdwDataLen) )
3400      {
3401        *v7 = pdwDataLen;
3402        result = (BOOL)v5;
3403      }
3404      else
3405      {
3406  LABEL_13:
3407        result = 0;
3408      }
3409    }
3410    return result;
3411  }
```

قطعه کد زیر مربوط به تابع IsProcessorFeaturePresent() می‌باشد که باج‌افزار با استفاده از آن بررسی می‌نماید که ویژگی‌های پردازنده موردنظر باج‌افزار با سیستم قربانی یکسان است یا خیر.

Sample_5b28bc5ec036f72574b3ceb6.c

```
7582    int v29; // [esp+2D4h] [ebp-54h]
7583    int v30; // [esp+2DCh] [ebp-4Ch]
7584    struct _EXCEPTION_POINTERS ExceptionInfo; // [esp+320h] [ebp-8h]
7585    int savedregs; // [esp+328h] [ebp+0h]
7586    int retaddr; // [esp+32Ch] [ebp+4h]
7587
7588    if ( IsProcessorFeaturePresent(0x17u) )
7589      __fastfail(a4);
7590    dword_40F808 = 0;
7591    v4 = memset(&Dst, 0, 0x2CCu);
7592    v21 = v4;
7593    v20 = v5;
7594    v19 = v6;
7595    v18 = a1;
7596    v17 = a3;
7597    v16 = a2;
7598    v27 = __SS__;
7599    v24 = __CS__;
7600    v15 = __DS__;
7601    v14 = __ES__;
7602    v13 = __FS__;
7603    v12 = __GS__;
7604    v7 = __readeflags();
7605    v25 = v7;
7606    v23 = retaddr;
7607    v26 = &retaddr;
7608    Dst = 65537;
7609    v22 = savedregs;
7610    memset(&v28, 0, 0x50u);
7611    v28 = 1073741845;
7612    v29 = 1;
7613    v30 = retaddr;
7614    v8 = IsDebuggerPresent();
7615    ExceptionInfo.ExceptionRecord = (PEXCEPTION_RECORD)&v28;
7616    v9 = v8 == 1;
7617    ExceptionInfo.ContextRecord = (PCONTEXT)&Dst;
7618    SetUnhandledExceptionFilter(0);
7619    result = UnhandledExceptionFilter(&ExceptionInfo);
7620    if ( !result )
7621    {
7622      result = -(v9 != 0);
7623      dword_40F808 &= result;
7624    }
7625    return result;
7626  }
```

قطعه کد زیر مربوط به دستور حذف فایل اجرایی باج‌افزار پس از پایان فرایند رمزگذاری فایل‌ها می‌باشد :

Sample_5b28bc5ec036f72574b3ceb6.c

```
3413    //----- (00403D31) --------------------------------------------------------
3414    BOOL sub_403D31()
3415    {
3416      BOOL result; // eax
3417      WCHAR String1; // [esp+8h] [ebp-414h]
3418      WCHAR Filename; // [esp+210h] [ebp-20Ch]
3419
3420      result = 0;
3421      if ( GetModuleFileNameW(0, &Filename, 0x104u) )
3422      {
3423        if ( GetShortPathNameW(&Filename, &Filename, 0x104u) )
3424        {
3425          lstrcpyW(&String1, L"/c del ");
3426          lstrcatW(&String1, &Filename);
3427          lstrcatW(&String1, L" >> NUL");
3428          if ( GetEnvironmentVariableW(L"ComSpec", &Filename, 0x104u) )
3429          {
3430            if ( (signed int)ShellExecuteW(0, 0, &Filename, &String1, 0, 0) > 32 )
3431              result = 1;
3432          }
3433        }
3434      }
3435      return result;
3436    }
3437
```

همانطور که اشاره نمودیم باج‌افزار PainLocker فایل‌های موجود در برخی از دایرکتوری‌ها را رمزگذاری نمی‌کند، قطعه کد زیر مربوط به این فرایند می‌باشد :

Sample_5b28bc5ec036f72574b3ceb6.c

```
1366    int v23; // [esp+94h] [ebp-4h]
1367
1368    v8 = 0;
1369    v9 = 7;
1370    v7 = 0;
1371    v0 = wcslen(L"Windows");
1372    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"Windows", v0);
1373    v11 = 0;
1374    v12 = 7;
1375    v10 = 0;
1376    v1 = wcslen(L"Program files");
1377    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"Program files", v1);
1378    v14 = 0;
1379    v15 = 7;
1380    v13 = 0;
1381    v2 = wcslen(L"Program files (x86)");
1382    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"Program files (x86)", v2);
1383    v17 = 0;
1384    v18 = 7;
1385    v16 = 0;
1386    v3 = wcslen(L"System volume information");
1387    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"System volume information", v3);
1388    v21 = 7;
1389    v19 = 0;
1390    v20 = 0;
1391    v4 = wcslen(L"$Recycle.bin");
1392    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"$Recycle.bin", v4);
1393    LOBYTE(v6) = 0;
1394    v23 = 4;
1395    dword_40F740 = 0;
1396    dword_40F744 = 0;
1397    dword_40F748 = 0;
1398    sub_402AC8(&dword_40F740, 0, (int)&v7, (int)&v22, v6);
1399    v23 = -1;
1400    sub_407D0B((int)&v7, 24, 5, (void (__thiscall *)(int))&loc_403564);
1401    return atexit((void (__cdecl *)())sub_409C51);
1402  }
1403  // 40F740: using guessed type int dword_40F740;
1404  // 40F744: using guessed type int dword_40F744;
1405  // 40F748: using guessed type int dword_40F748;
1406
```

همانطور که اشاره نمودیم این باج‌افزار ساختار برخی فایل‌ها با پسوندهای مشخص را به طور کامل تغییر می‌دهد، لیست برخی از این فایل‌ها در قطعه کد زیر قابل مشاهده می‌باشد :

Sample_5b28bc5ec036f72574b3ceb6.c

```
1490    v5 = wcslen(L".dacpac");
1491    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L".dacpac", v5);
1492    v33 = 0;
1493    v34 = 7;
1494    v32 = 0;
1495    v6 = wcslen(L".db3");
1496    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L".db3", v6);
1497    v36 = 0;
1498    v37 = 7;
1499    v35 = 0;
1500    v7 = wcslen(L".dtxs");
1501    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L".dtxs", v7);
1502    v39 = 0;
1503    v40 = 7;
1504    v38 = 0;
1505    v8 = wcslen(L".mdt");
1506    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L".mdt", v8);
1507    v42 = 0;
1508    v43 = 7;
1509    v41 = 0;
1510    v9 = wcslen(L".sdf");
1511    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L".sdf", v9);
1512    v45 = 0;
1513    v46 = 7;
1514    v44 = 0;
1515    v10 = wcslen(L".MDF");
1516    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L".MDF", v10);
1517    v49 = 7;
1518    v48 = 0;
1519    v47 = 0;
1520    v11 = wcslen(L".DBF");
1521    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L".DBF", v11);
1522    LOBYTE(v13) = 0;
1523    v51 = 11;
1524    dword_40F758 = 0;
1525    dword_40F75C = 0;
1526    dword_40F760 = 0;
1527    sub_402AC8(&dword_40F758, 0, (int)&v14, (int)&v50, v13);
1528    v51 = -1;
1529    sub_407D0B((int)&v14, 24, 12, (void (__thiscall *)(int))&loc_403564);
1530    return atexit((void (__cdecl *)())sub_409C6A);
1531  }
```

قطعه کد زیر مربوط به تابع GetSystemTimeAsFileTime() می‌باشد که باج‌افزار با استفاده از آن تاریخ و زمان سیستم قربانی را بازیابی می‌کند که به نظر می‌رسد باج‌افزار تنها کاربران خاصی در نقاط مختلف دنیا را مورد هدف خود قرار می‌دهد :

Sample_5b28bc5ec036f72574b3ceb6.c

```
7852  //----- (00408264) --------------------------------------------------------
7853  unsigned int sub_408264()
7854  {
7855    LARGE_INTEGER PerformanceCount; // [esp+0h] [ebp-14h]
7856    struct _FILETIME SystemTimeAsFileTime; // [esp+8h] [ebp-Ch]
7857    DWORD v3; // [esp+10h] [ebp-4h]
7858
7859    SystemTimeAsFileTime.dwLowDateTime = 0;
7860    SystemTimeAsFileTime.dwHighDateTime = 0;
7861    GetSystemTimeAsFileTime(&SystemTimeAsFileTime);
7862    v3 = SystemTimeAsFileTime.dwLowDateTime ^ SystemTimeAsFileTime.dwHighDateTime;
7863    v3 ^= GetCurrentThreadId();
7864    v3 ^= GetCurrentProcessId();
7865    QueryPerformanceCounter(&PerformanceCount);
7866    return (unsigned int)&v3 ^ v3 ^ PerformanceCount.LowPart ^ PerformanceCount.HighPart;
7867  }
```

قطعه کد زیر مربوط به تابع GetLogicalDrives() می‌باشد که با استفاده از این تابع درایوهای سیستم قربانی شناسایی می‌شوند :

```
Sample_5b28bc5ec036f72574b3ceb6.c
6291      v12 = &v13;
6292      sub_40282E(&v16, (int *)&v11);
6293      v5 = GetLogicalDrives();
6294      v12 = (int *)v5;
6295      LOBYTE(v2) = 65;
6296      v6 = 0;
6297      do
6298      {
6299        if ( (1 << v6) & v5 )
6300        {
6301          v20 = 7;
6302          v19 = 0;
6303          Dst = 0;
6304          LOBYTE(v21) = 2;
6305          sub_404E67(&Dst, 1u, (unsigned __int16)(char)v2);
6306          v7 = wcslen(L":\\");
6307          sub_404EF1(&Dst, L":\\", v7);
6308          sub_406A41((unsigned int *)&v13, v2, (unsigned int)&Dst);
6309          LOBYTE(v21) = 1;
6310          sub_404B1A(&Dst, 1, 0);
6311          v5 = (DWORD)v12;
6312        }
6313        LOBYTE(v2) = v2 + 1;
6314        ++v6;
6315      }
6316      while ( (char)v2 <= 90 );
6317      v8 = v13;
6318      if ( (v14 - v13) / 24 )
6319      {
6320        v6 = 0;
6321        do
6322        {
6323          v9 = (wchar_t *)(v6 + v8);
6324          if ( *(_DWORD *)(v6 + v8 + 20) >= 8u )
6325            v9 = *(wchar_t **)v9;
6326          sub_406609(v9, a1);
6327          ++v1;
6328          v8 = v13;
6329          v6 += 24;
6330        }
6331        while ( v1 < (v14 - v13) / 24 );
6332      }
6333      sub_404AA7(&v16);
6334      return sub_404B9D((int)&v13, 24, v6);
```

همانطور که پیشتر نیز اشاره شد باج‌افزار PainLocker پس از رمزگذاری فایل‌ها، به انتهای آن‌ها عبارت [Pain@cock.lu].pain. را اضافه می‌کند، که در قطعه کد زیر این موضوع به خوبی قابل مشاهده است :

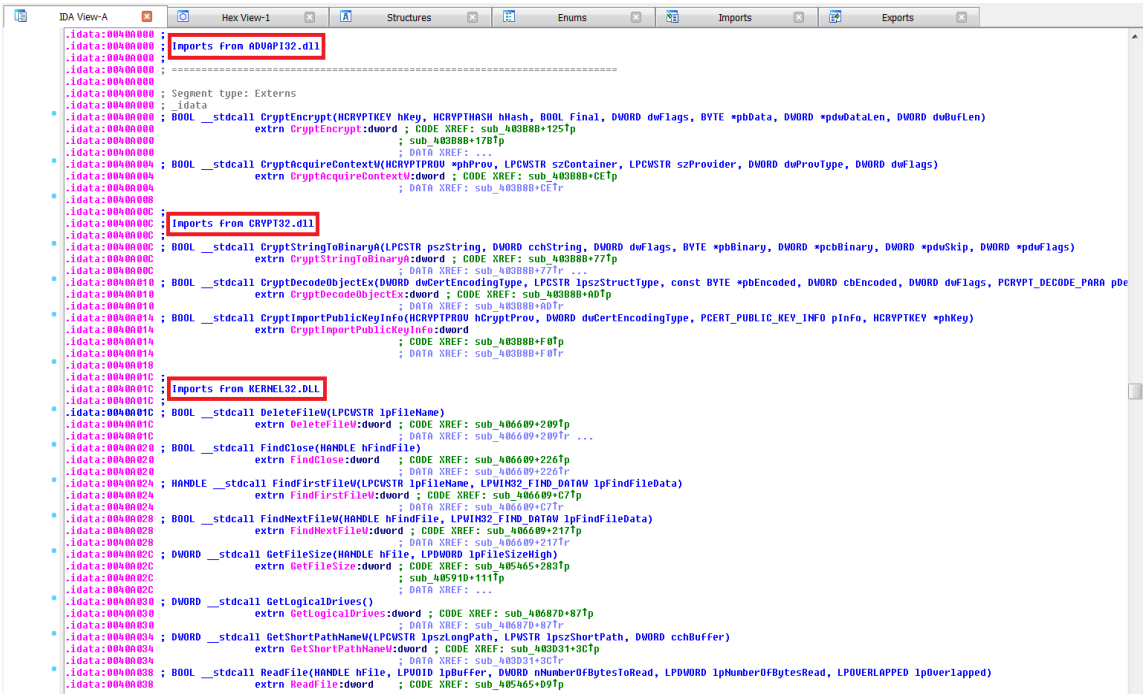
```
IDA View-A | Hex View-1 | Structures | Enums
.text:00401AD3 ; ---------------------------------------------------------------------------
.text:00401AD3                 push    esi
.text:00401AD4                 mov     esi, offset a_Pain@cock_lu_ ; ".[pain@cock.lu].pain"
.text:00401AD9                 push    esi
.text:00401ADA                 call    ds:wcslen
.text:00401AE0                 pop     ecx
.text:00401AE1                 push    eax
.text:00401AE2                 push    esi
.text:00401AE3                 mov     ecx, offset Dst
.text:00401AE8                 call    sub_4051AF
.text:00401AED                 push    offset sub_409C5B
.text:00401AF2                 call    _atexit
.text:00401AF7                 pop     ecx
.text:00401AF8                 pop     esi
.text:00401AF9                 retn
```

همچنین اشاره نمودیم که باج‌افزار PainLocker پس از اجرا از ادامه‌ی فعالیت برخی فرایندها جلوگیری کرده و همچنین مانع اجرای مجدد آن‌ها می‌شود. لیست برخی از این فرایندها در تصویر زیر قابل مشاهده می‌باشد :

Sample_5b28bc5ec036f72574b3ceb6.c

```
1583    v13 = 0;
1584    v14 = 7;
1585    v12 = 0;
1586    v0 = wcslen(L"sqlserv.exe");
1587    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"sqlserv.exe", v0);
1588    v16 = 0;
1589    v17 = 7;
1590    v15 = 0;
1591    v1 = wcslen(L"oracle.exe");
1592    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"oracle.exe", v1);
1593    v19 = 0;
1594    v20 = 7;
1595    v18 = 0;
1596    v2 = wcslen(L"ntdbsmgr.exe");
1597    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"ntdbsmgr.exe", v2);
1598    v22 = 0;
1599    v23 = 7;
1600    v21 = 0;
1601    v3 = wcslen(L"sqlservr.exe");
1602    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"sqlservr.exe", v3);
1603    v25 = 0;
1604    v26 = 7;
1605    v24 = 0;
1606    v4 = wcslen(L"sqlwriter.exe");
1607    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"sqlwriter.exe", v4);
1608    v28 = 0;
1609    v29 = 7;
1610    v27 = 0;
1611    v5 = wcslen(L"MsDtsSrvr.exe");
1612    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"MsDtsSrvr.exe", v5);
1613    v31 = 0;
1614    v32 = 7;
1615    v30 = 0;
1616    v6 = wcslen(L"msmdsrv.exe");
1617    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(L"msmdsrv.exe", v6);
1618    v34 = 0;
1619    v35 = 7;
1620    v33 = 0;
1621    v7 = wcslen(L"ReportingServecesService.exe");
1622    std::basic_string<wchar_t,std::char_traits<wchar_t>,std::allocator<wchar_t>>::assign(
1623      L"ReportingServecesService.exe",
1624      v7);
```

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.



KERNEL۳۲.DLL	CRYPT۳۲.dll	MPR.dll	SHELL۳۲.dll	ADVAPI۳۲.dll	msvcrt.dll
VirtualProtect LoadLibraryA ExitProcess GetProcAddress	CryptDecodeObjectEx	WNetCloseEnum	ShellExecuteW	CryptEncrypt	exit

بر اساس بررسی‌های صورت گرفته، این باج‌افزار پس از اجرا فرایندهای زیر را ایجاد می‌کند :

PainLocker.exe

- vssadmin.exe vssadmin delete shadows /all /quiet
- cmd.exe /c timeout ١ && del "C:\d٩ea٣٧b٤٣b٥٩٧٣٤٠d١٤٣٠f٣b٩٧cb٧٢d٨٤ef٨٨aeb٥٥٩acf٨٦٨dfd٠a٧٣de٤f٤٥f٨.exe" >> NUL
  - timeout.exe timeout ١

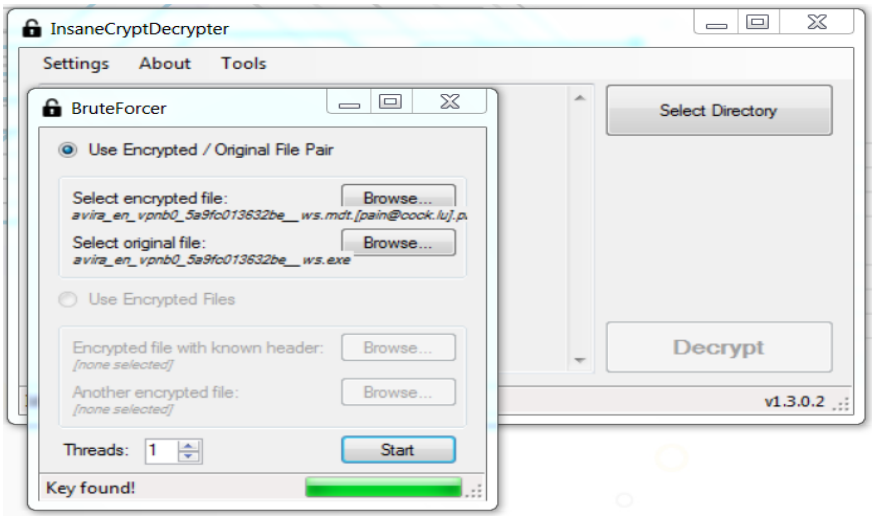
با اجرای فرایند vssadmin.exe و انجام دستور Delete Shadows /All /Quiet نسخه‌های shadowcopy حذف می‌شوند.

با اجرای فرایند cmd.exe و انجام دستور c timeout ١ && del فایل اجرایی باج‌افزار حذف خواهد شد.

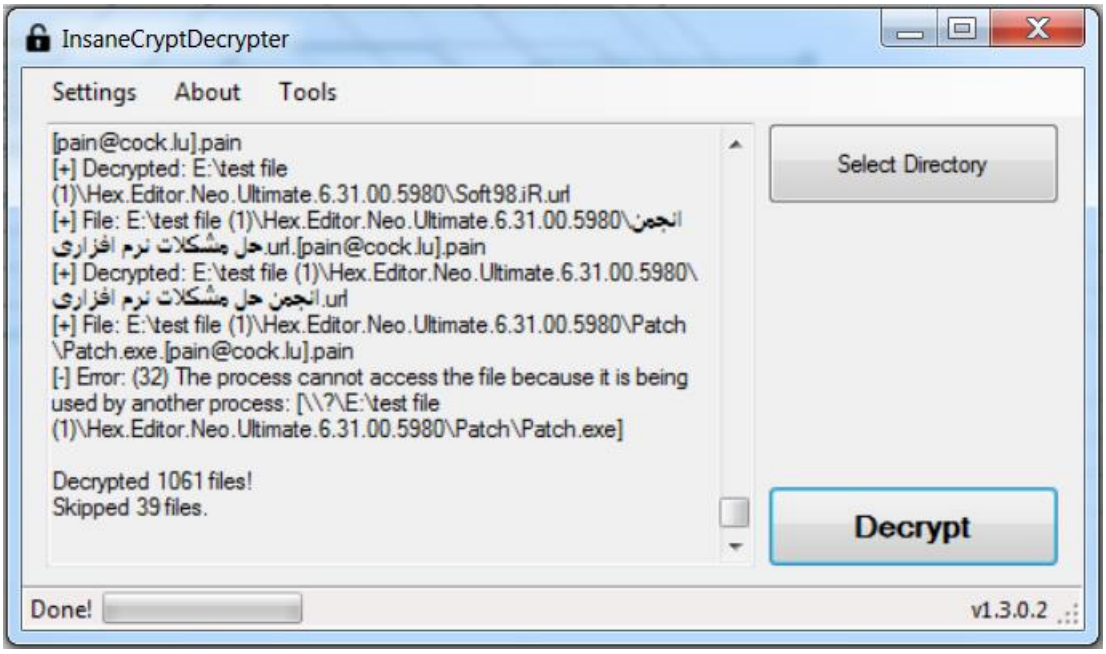
فرایند رمزگشایی :

طبق بررسی‌های صورت گرفته فایل‌های رمزگذاری شده توسط این نسخه از خانواده‌ی Everbe، قابل رمزگشایی می‌باشند و قربانیان می‌توانند با استفاده از ابزار رمزگشایی منتشر شده، به راحتی فایل‌های خود را رمزگشایی نمایند. در زیر توضیح مختصری درباره‌ی نحوه‌ی رمزگشایی فایل‌ها ارائه شده است.

جهت رمزگشایی فایل‌ها با استفاده از ابزار رمزگشایی ارائه شده، قربانیان نیاز به کلید رمزگشایی دارند که این کلید را می‌توانند از مقایسه‌ی یک نمونه فایل سالم با نمونه‌ی رمزگذاری شده‌ی آن بدست بیاورند. در زیر تصویر مربوط به این فرایند قابل مشاهده است :



پس از بدست آوردن کلید رمزگشایی، قربانیان به راحتی می‌توانند دایرکتوری‌های مورد نظر خود را انتخاب نمایند و با کلیک بر روی دکمه‌ی Decrypt فرایند رمزگشایی فایل‌ها آغاز خواهد شد، تصویر زیر مربوط به این فرایند می‌باشد :



همانطور که مشاهده می‌گردد فایل‌ها با موفقیت رمزگشایی شده‌اند اما به دلیل آسیب‌های وارده به سیستم قربانیان توسط باج‌افزار، آن‌ها باید ویندوز سیستم خود را تعویض نمایند.

تحلیل ترافیک شبکه :

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

خروجی سامانه VirusTotal :

در حال حاضر تعداد ٥٤ مورد از ٦٧ آنتی ویروس و آنتی بـدافزار موجـود در سـامانه VirusTotal قـادر بـه شناسایی این باج افزار بوده وآن را حذف یا غیرفعال می‌کنند.

Ad-Aware	Generic.Ransom.Everbe.90A281DE	AegisLab	Troj.Ransom.W32.Gen!c
AhnLab-V3	Malware/Win32.Generic.C2541232	ALYac	Trojan.Ransom.PainLocker
Antiy-AVL	Trojan[Ransom]/Win32.AGeneric	Arcabit	Generic.Ransom.Everbe.90A281DE
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
Avira	TR/DelFile.qfokv	AVware	Trojan.Win32.Generic!BT
Baidu	Win32.Trojan.WisdomEyes.16070401....	BitDefender	Generic.Ransom.Everbe.90A281DE
Bkav	W32.eHeur.Malware14	CAT-QuickHeal	Trojan.IGENERIC
Comodo	.UnclassifiedMalware	CrowdStrike Falcon	malicious_confidence_90% (W)
Cybereason	malicious.b1726f	Cylance	Unsafe
Cyren	W32/Trojan.FQQT-5450	Emsisoft	Generic.Ransom.Everbe.90A281DE (B)
Endgame	malicious (moderate confidence)	eScan	Generic.Ransom.Everbe.90A281DE
ESET-NOD32	a variant of Win32/Filecoder.NQU	F-Secure	Generic.Ransom.Everbe.90A281DE
Fortinet	W32/Gen.IYH!tr	GData	Generic.Ransom.Everbe.90A281DE
Ikarus	Trojan-Ransom.FileCoder	Jiangmin	Trojan.Gen.wa
K7AntiVirus	Riskware (0040eff71)	K7GW	Riskware (0040eff71)
Kaspersky	Trojan-Ransom.Win32.Gen.iyh	Malwarebytes	Trojan.Agent
MAX	malware (ai score=97)	McAfee	Artemis!B047A4AB1726
McAfee-GW-Edition	BehavesLike.Win32.Dropper.nc	Microsoft	Ransom:Win32/Genasom
NANO-Antivirus	Trojan.Win32.DelFile.fcuux2	Palo Alto Networks	generic.ml
Panda	Trj/RnkBend.A	Qihoo-360	Win32/Trojan.IM.ff2
Sophos AV	Mal/Behav-031	Sophos ML	heuristic
Symantec	Ransom.CryptXXX	TACHYON	Ransom/W32.PainLocker.64000
Tencent	Win32.Trojan.Gen.Pbyl	TheHacker	Posible_Worm32
TrendMicro	Ransom_PAIN.THECOAH	TrendMicro-HouseCall	Ransom_PAIN.THECOAH
VIPRE	Trojan.Win32.Generic!BT	ViRobot	Trojan.Win32.Z.Ransom.31232
Webroot	W32.Ransom.Gen	Yandex	Trojan.Gen!YKWQhJ53cTQ
Zillya	Trojan.Filecoder.Win32.7773	ZoneAlarm	Trojan-Ransom.Win32.Gen.iyh

خروجی سامانه ویروس‌کاو مرکز ماهر :

در حال حاضر تعداد ۷ 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.

نام فایل: Honest_Sample_5b28bc5ec036f72574b3ceb61.bin.b047a4ab1726fe7414484493c60d5870

پرینت

حجم فایل: ۳۱ کیلوبایت

تاریخ اسکن: ۲۳ آبان ۱۳۹۷ - ۳:۳۰

MD5: b047a4ab1726fe7414484493c60d5870

SHA1: e9a544bac430261008af9fbf80ab6fba6385c2cb

SHA256: d9ea37b43b597340d1430f3b97cb72d84ef88aeb559acf868dfd0a73de4f45f8

وضعیت:

نتایج اسکن:

آنتی ویروس	نتیجه اسکن	
avast	ii	Dangerous
clamav	✓	Clean
fsecure	✓	Clean
پادویش	✓	Clean
kaspersky	ii	Dangerous
comodo	ii	Dangerous
drweb	✓	Clean
bitdefender	ii	Dangerous
eset	ii	Dangerous a variant of Win32/Filecoder.NQU trojan
sophos	ii	Dangerous Mal/Behav-031
symantec	ii	Dangerous Ransom.CryptXXX