Malware Analysis

Executive Summary

The malware sample.exe is a ransomware called cryptolocker. It functions by encrypting all the files present in the system drives and prompts the user to provide monetary compensation if he requires decryption. The malware attempts to contact a large number of command and control servers using a 'Domain Generation Algorithm' in the hopes of reaching out to an active server. Once this connection is established, the server generates a public-private key pair which the malware then uses to encrypt the user's files. While the malware is primarily distributed via the Gameover Zeus botnet, it is also found in spam emails as attachments. The information presented below is a concise representation of the analysis done on the malware.

Basic Static Analysis

The program is not packed as inferred from PEiD. In addition, we can observe that the virtual size and the raw data size of the .text section are roughly the same further confirming that the program is unpacked. The Malware was compiled on the 26th of November 2013 at 7:11 PM as observed from PEstudio and it is a windows GUI application. This particular sample has an MD5 hash of 8466505F0F110825420BEAC9CE551A29.

The program's resources (provided below) allow us to identify it's working in closer detail. A few window icons and images are found which establish the malware as a GUI program. Furthermore, on analysing the text resources, the program informs the user that his files are encrypted and that he must pay a particular amount to a certain mentioned virtual address in order to get them decrypted. This further confirms our previous statement that the malware is a ransomware called Cryptolocker. A few images can be found pertaining to payment options that the user can use to make the payment like Ukash, bitcoin etc. The program sections and contents are provided below.

On analysing the imports, the program appears to import a large number of http modules implying that the malware attempts to make http connections to a certain server to send or receive data. Another set of libraries imported are crypto libraries used in the encryption and decryption processes. We also observe that there are a few functions used to create, modify and delete registry keys which can potentially be harmful. A few suspicious imports are provided below.

There are also a large number of strings that are of concern. We provide a few important ones. "Microsoft enhanced RSA and AES cryptographic provider", "Microsoft enhanced cryptographic provider v1.0", "CryptStringtoBinary", "CryptDecodeObjectEx" and "CryptImportPublicKeyInfo" are all related to cryptographic schemes used by the Malware. We also see a number of file format strings like "*.p7c", "*.orf" and "*.pef". The malware encrypts all files with these formats on the user's drives. There are around 50 such file formats. A large number of HTTP related strings can also be seen like "WinHttpReceiveResponse", "WinHttpSendRequest", "WinHttpWriteData" and "WinHttpReadData". An IP address is also seen, "184.164.136.134" which is assumed to be the initial Command and Control Server that the malware tries to contact before running its DGA. A few other strings of importance are "Software\Microsoft\Windows\CurrentVersion\Run" and "Software\Cryptolocker". The former is a registry key that is updated to add the malware to a list of autorun programs to run on each startup while the latter is the name of the malware itself. We delve deeper into these findings in

the upcoming sections.

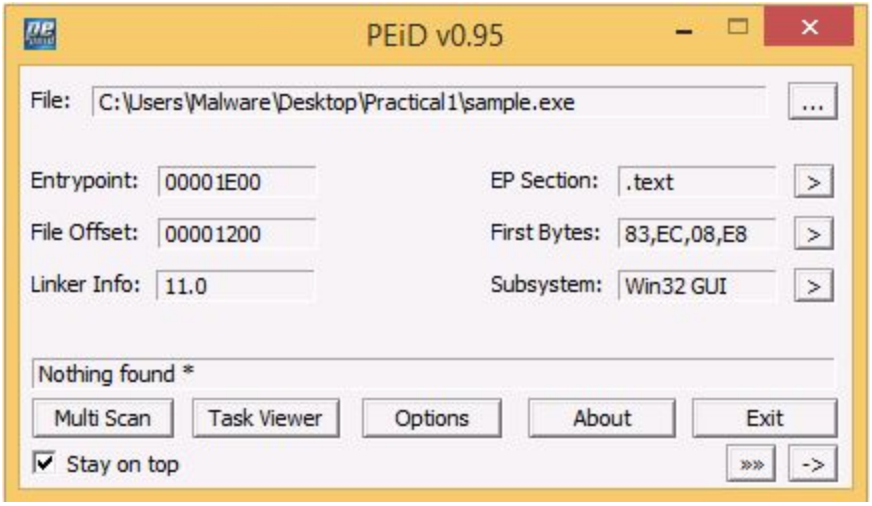

Fig.1 Unpacked program

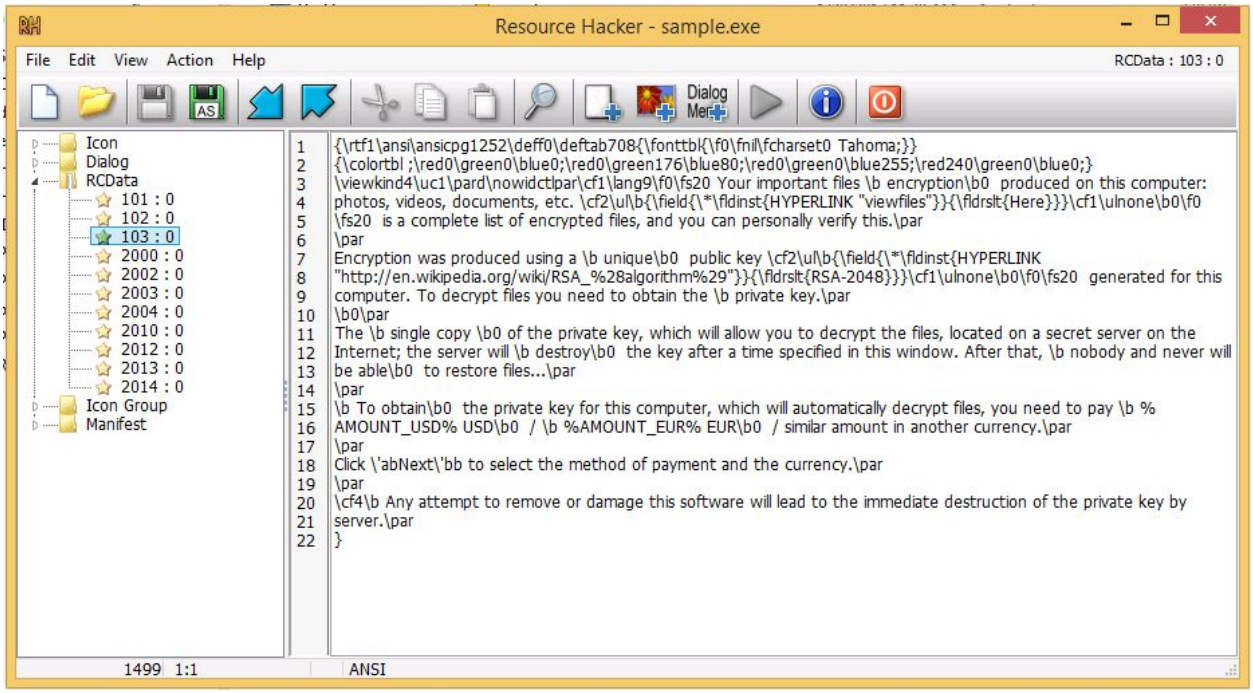

Fig.2 Text Resources

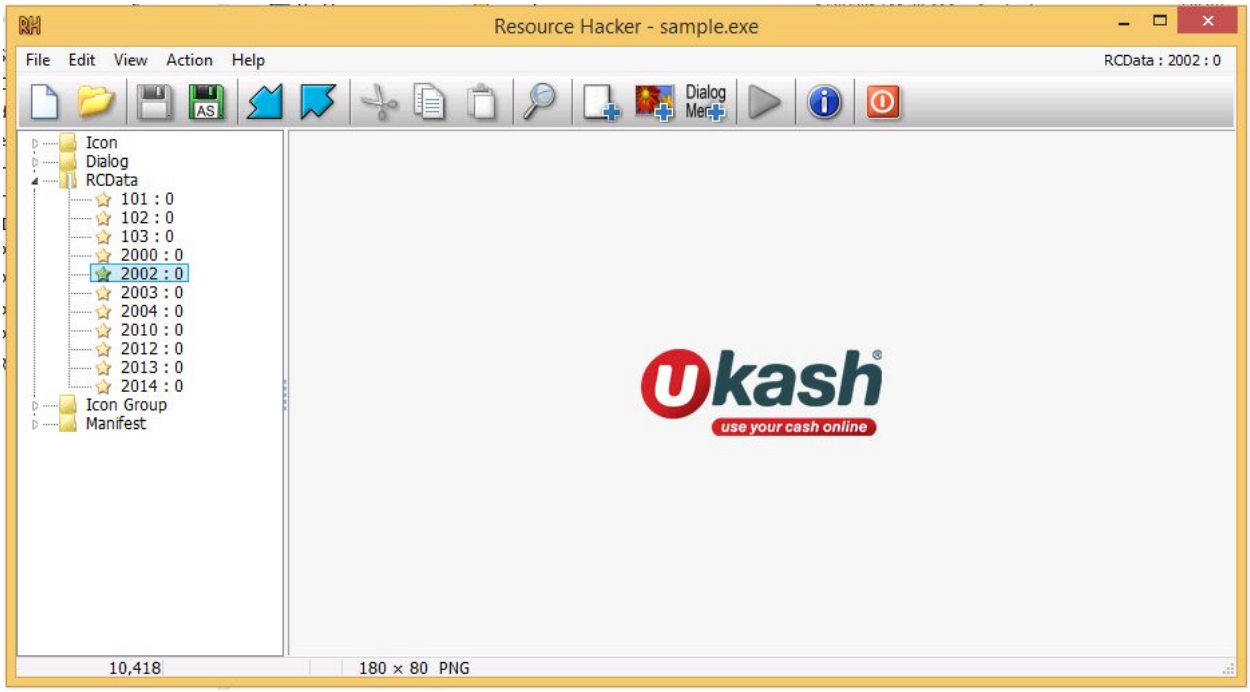

Fig.3 Payment option

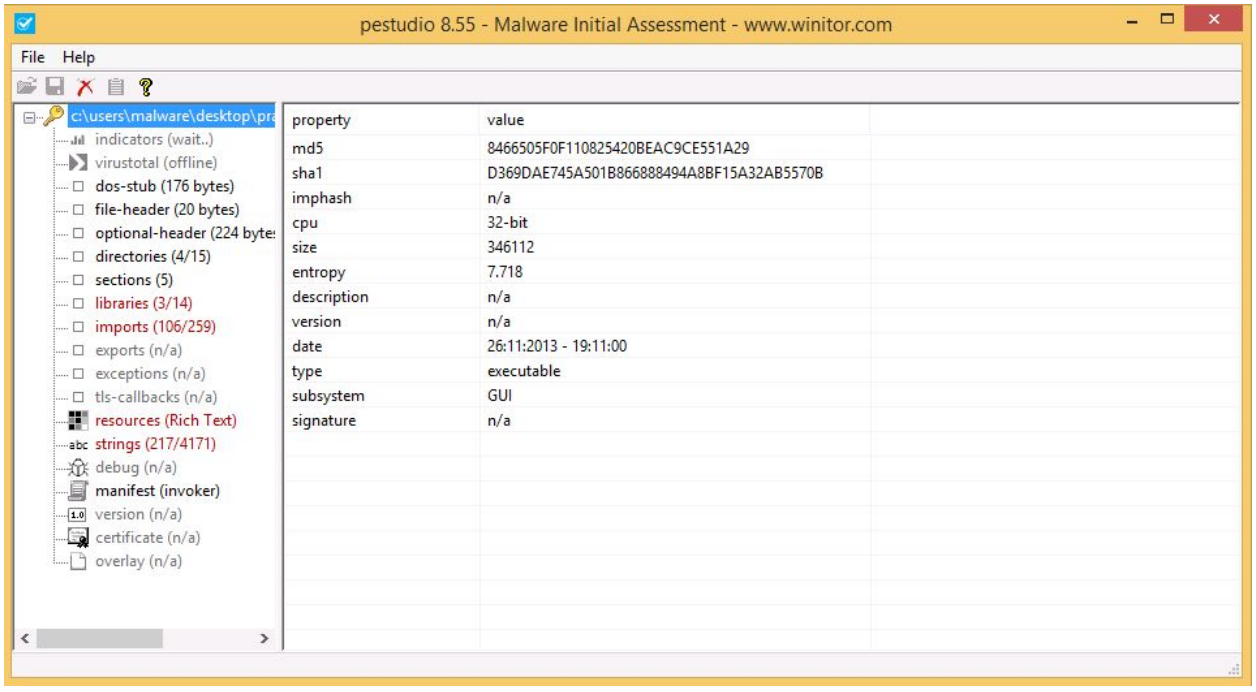

Fig.4 Malware information

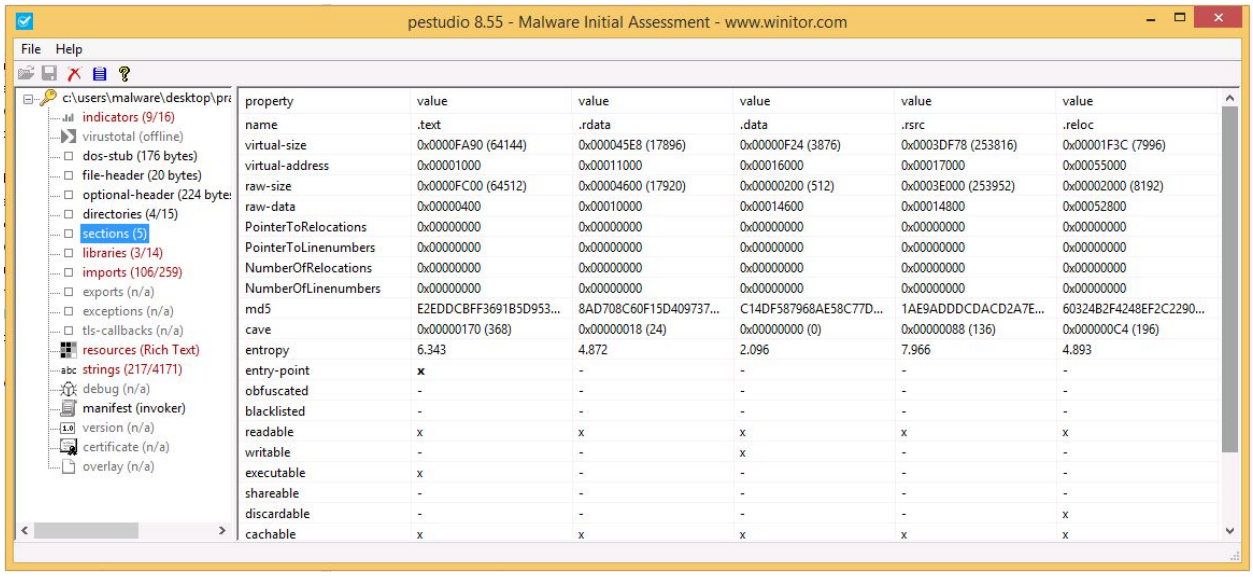

Fig.5 Program Sections

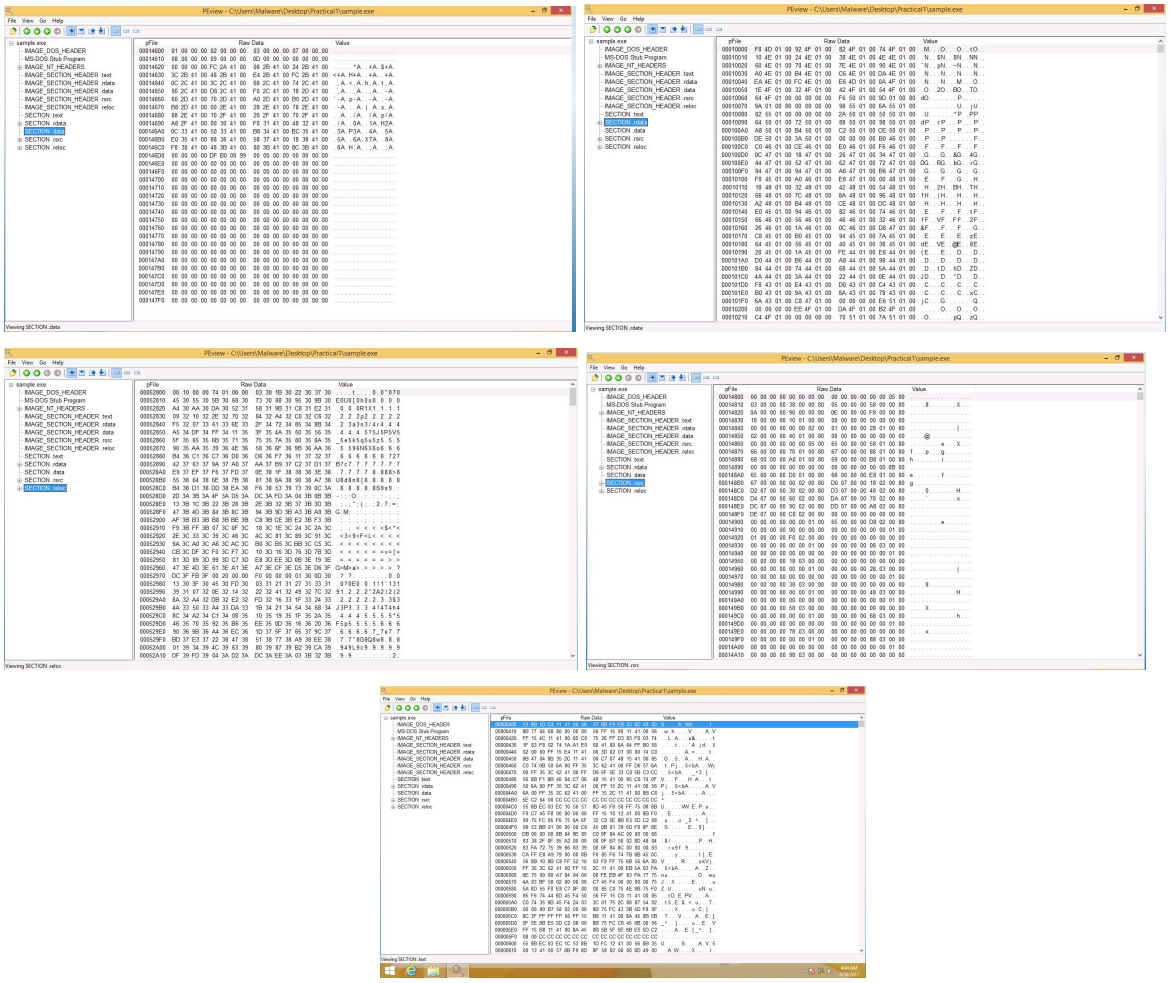

Fig.6 Section Contents

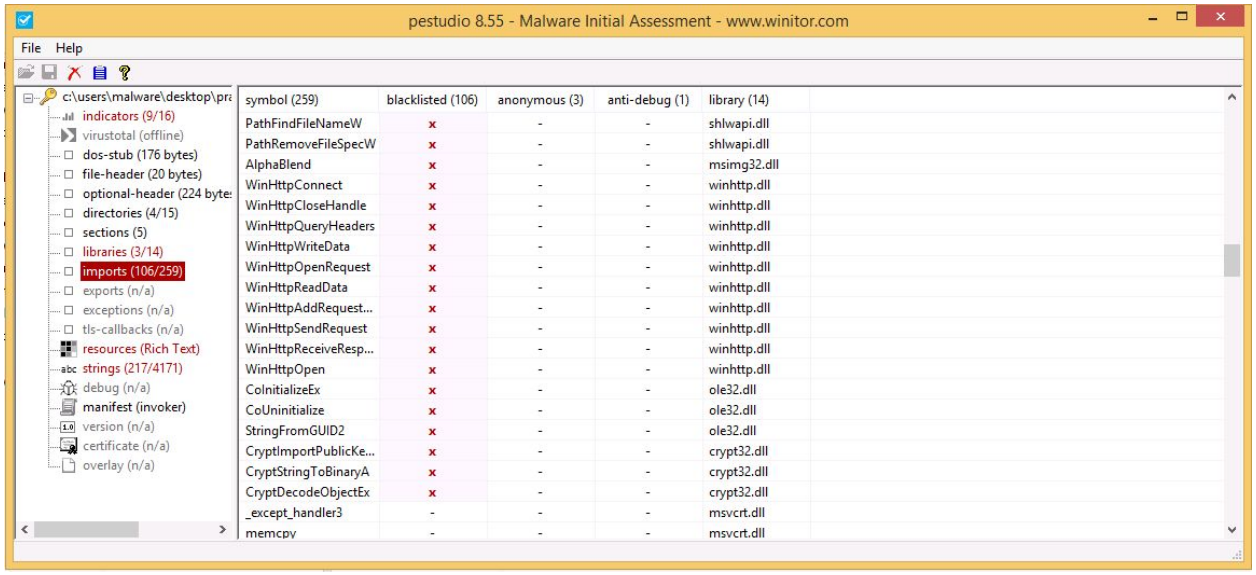

Fig.7 Http Imports

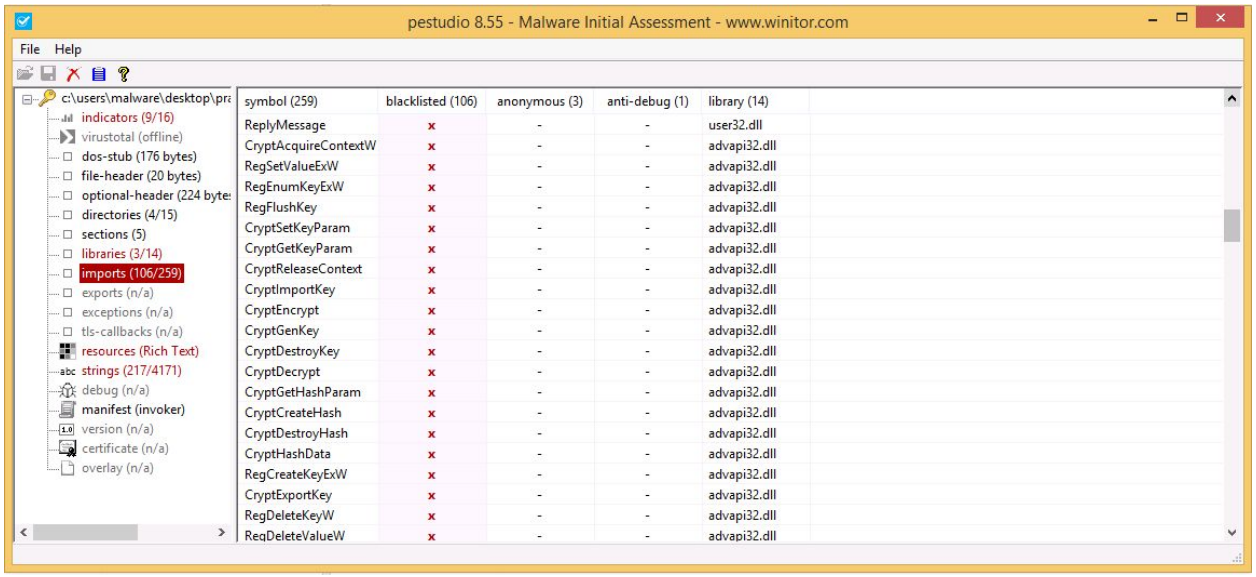

Fig.8 Crypto and Registry Imports

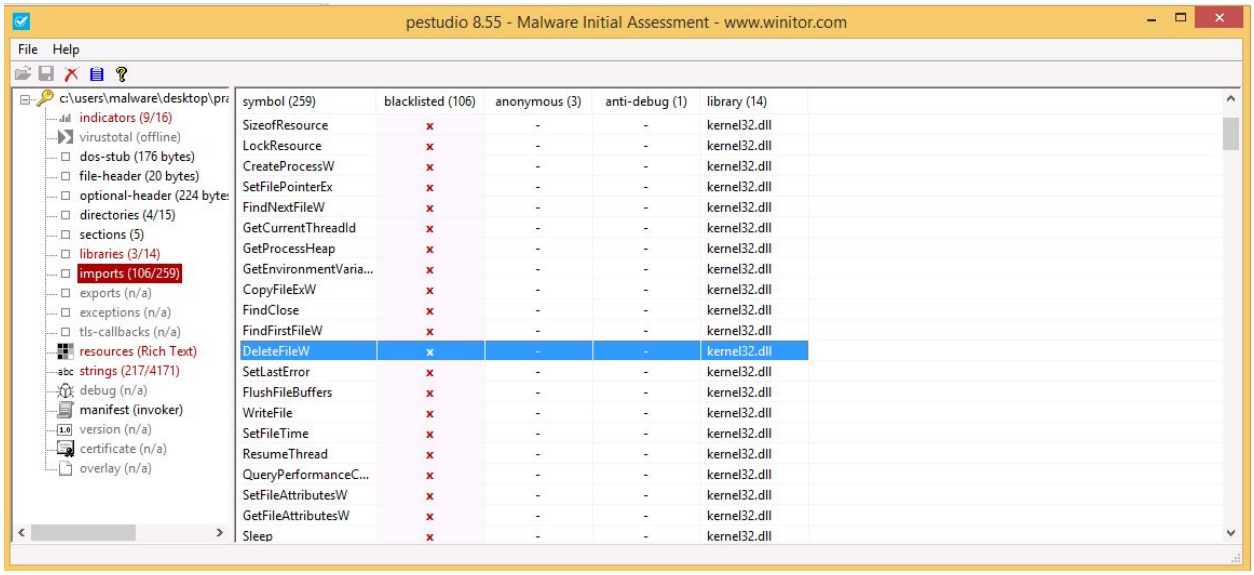

Fig.9 Delete Imports

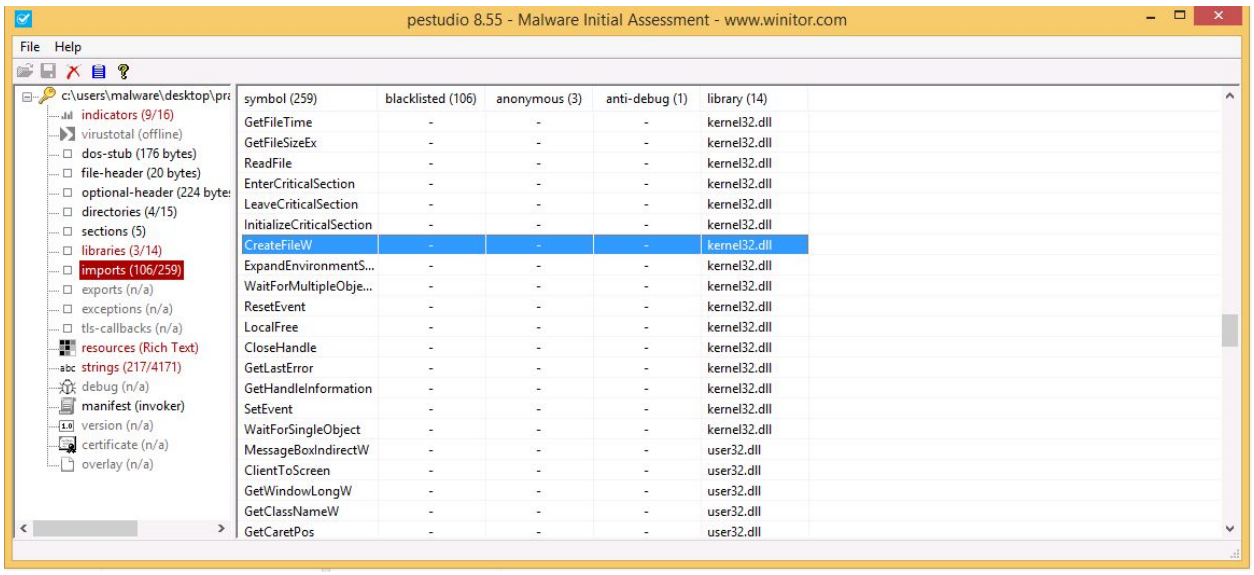

Fig.10 Create Imports

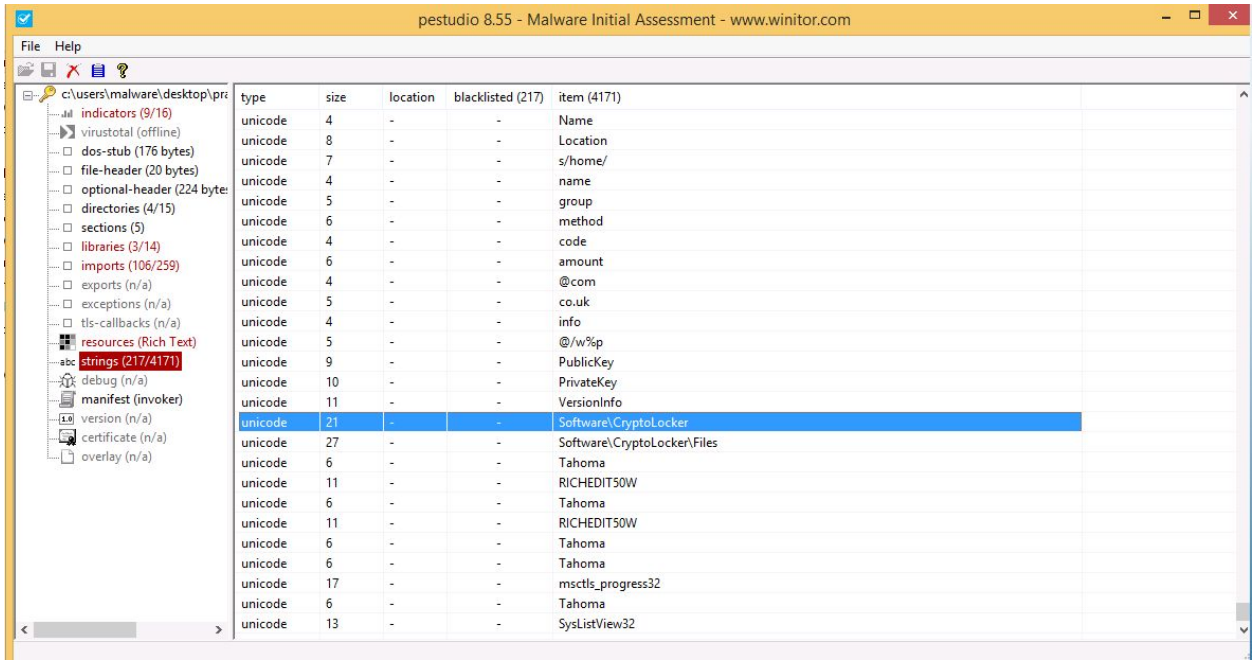

Fig.11 CryptoLocker String

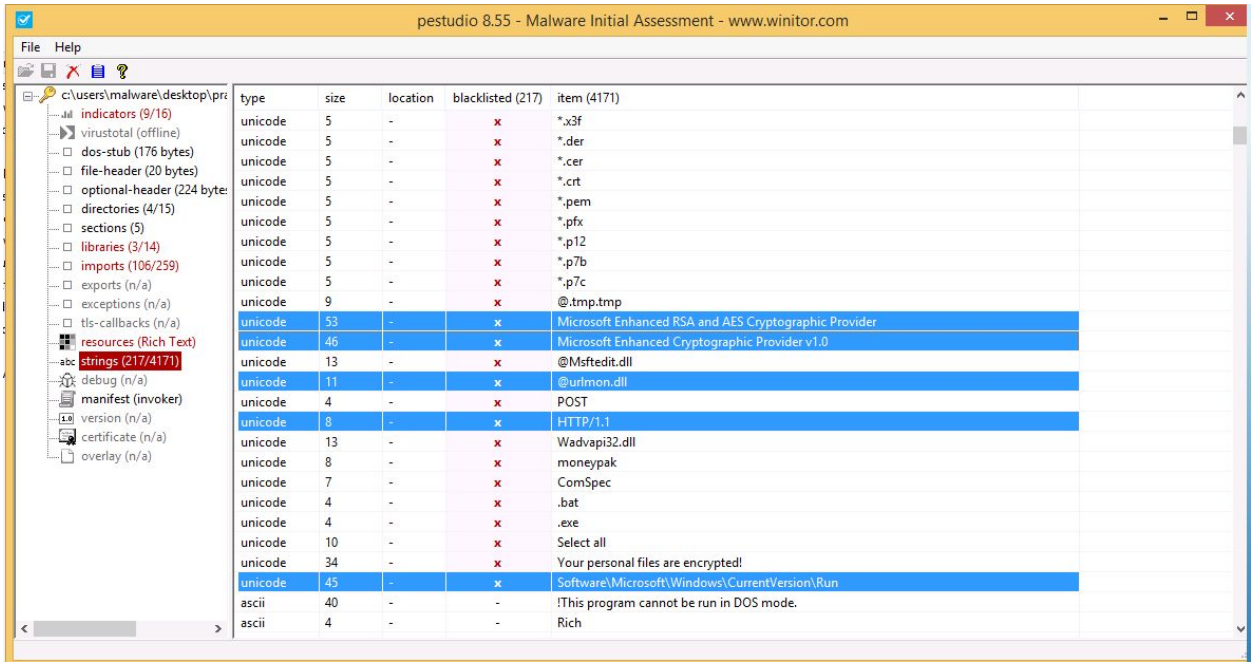

Fig.12 Important Strings

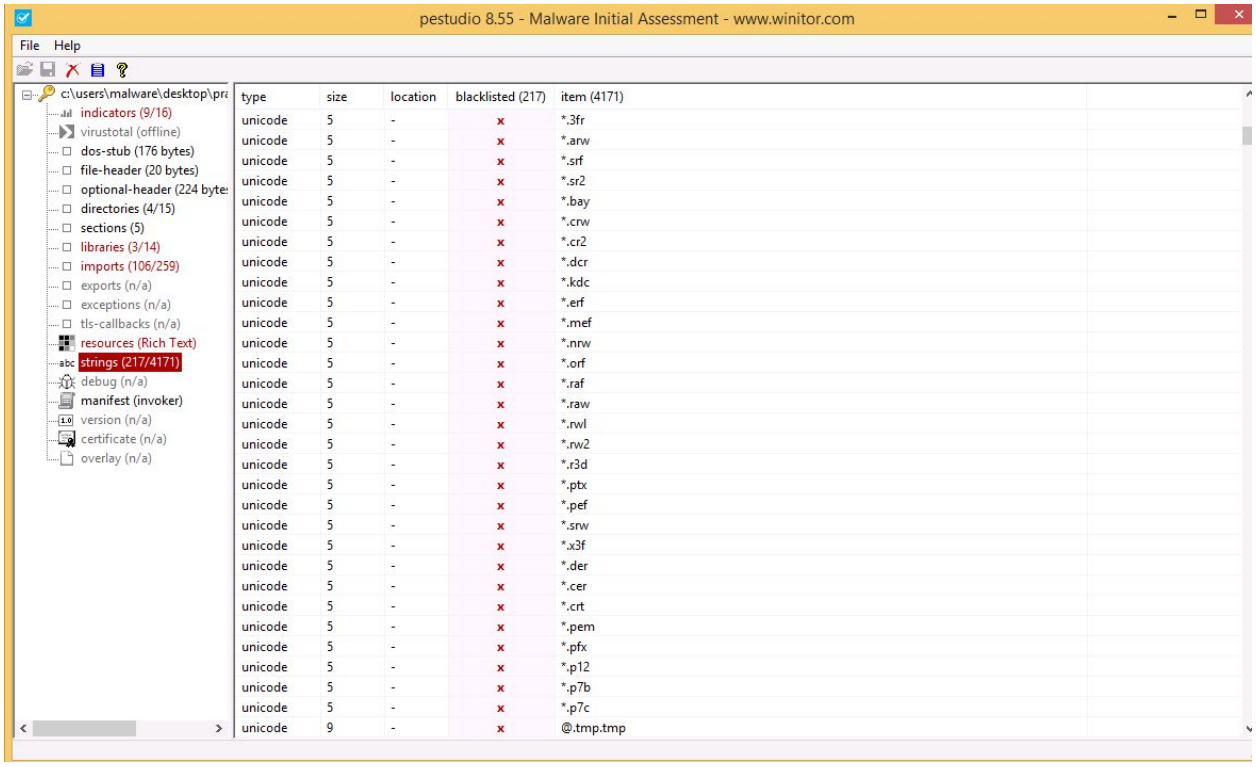

Fig.13 File types

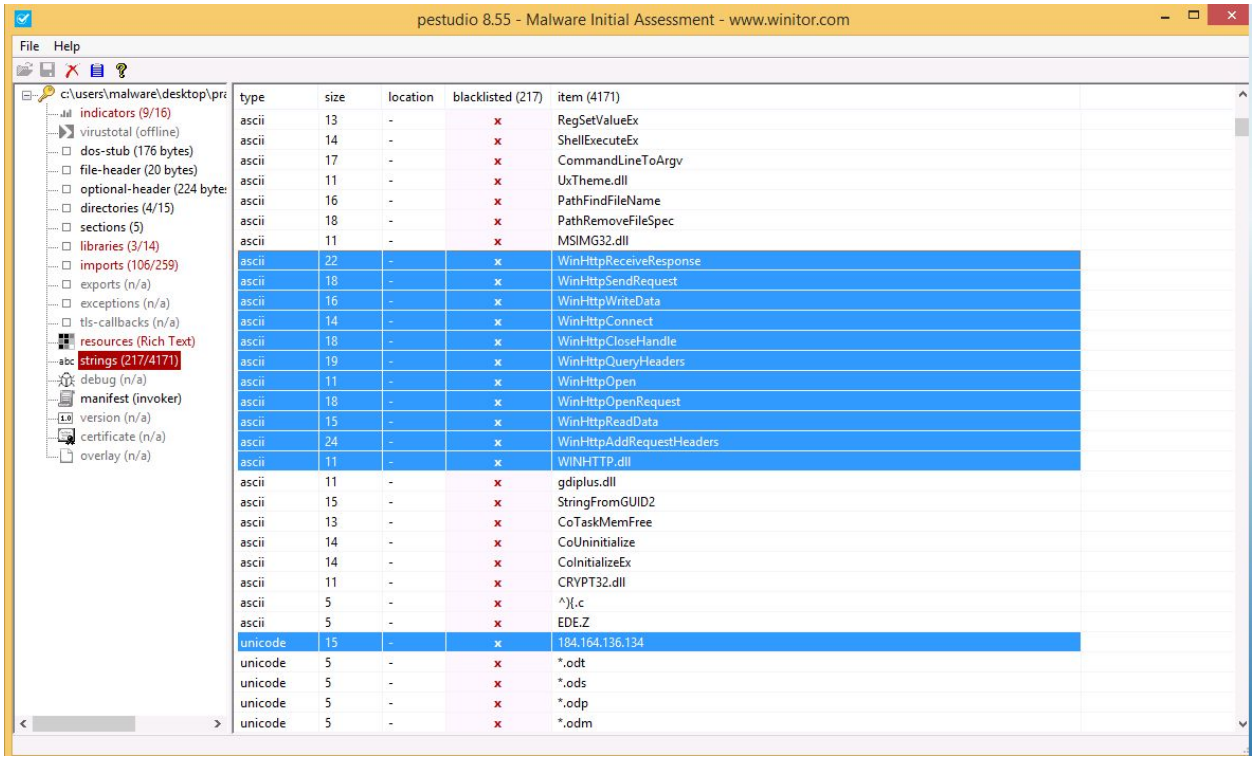

Fig.14 Http and IP

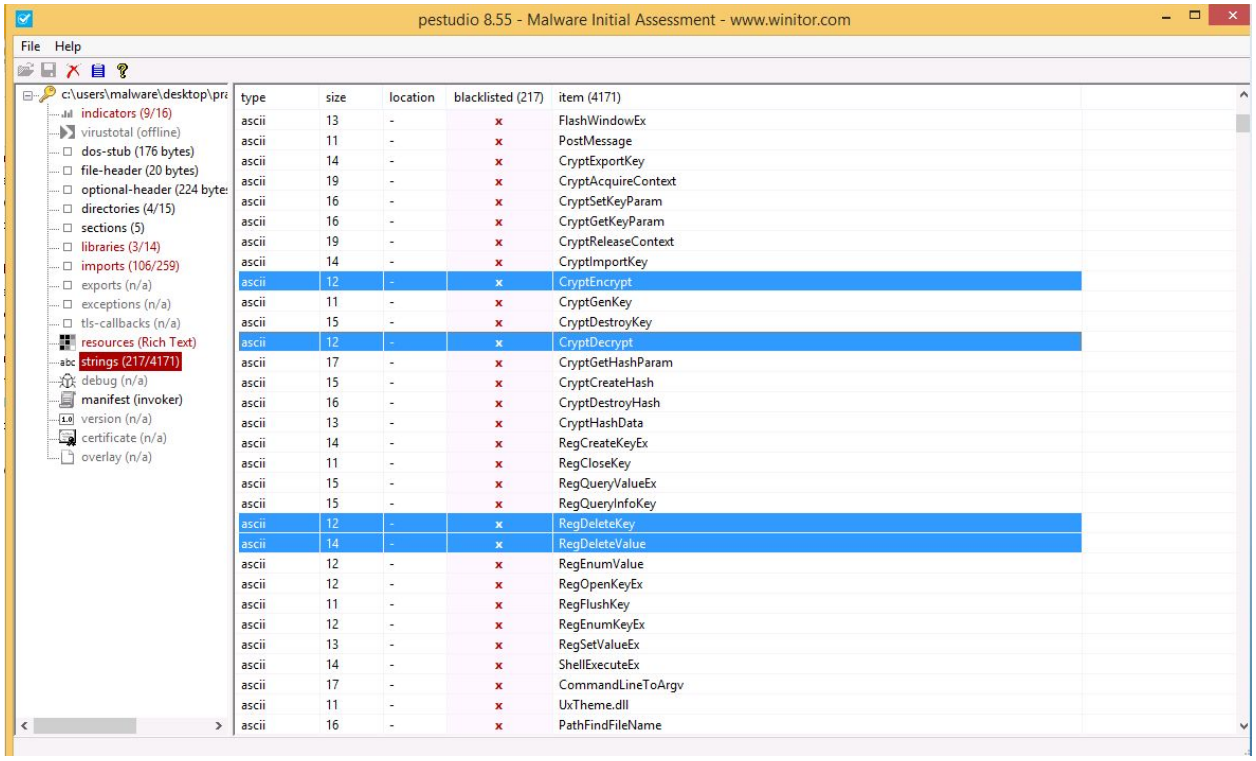

Fig.15 Crypto and Registry Strings

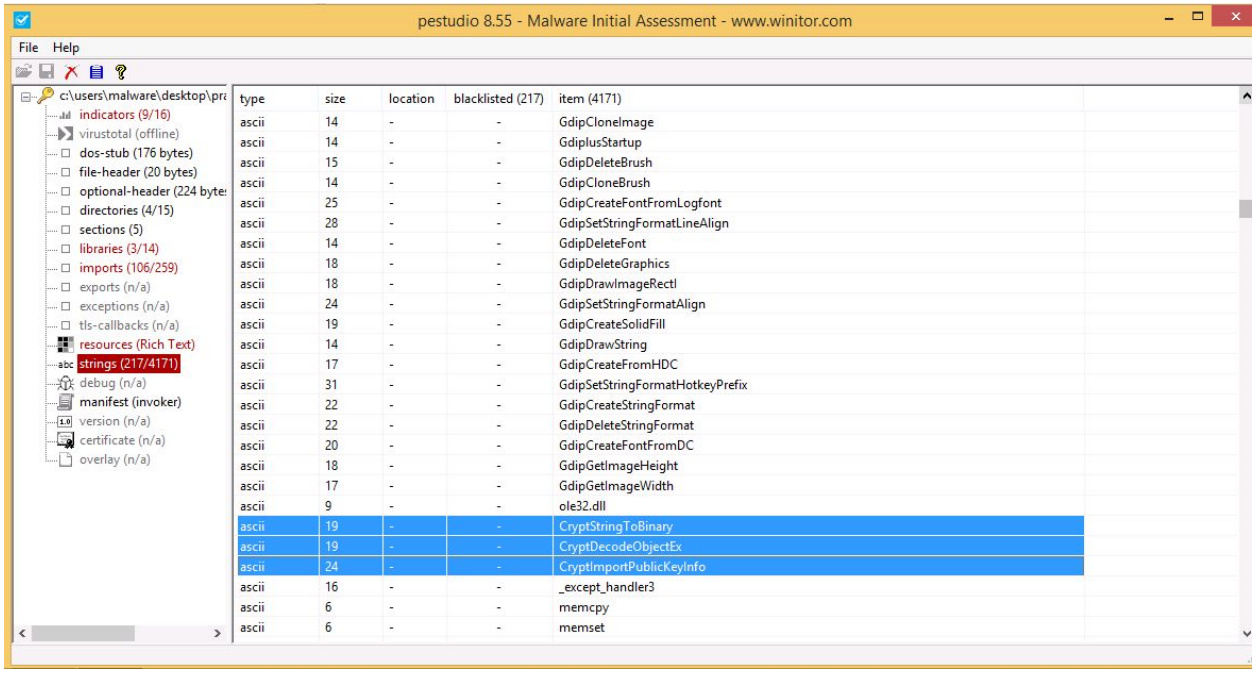

Fig.16 Crypto Strings

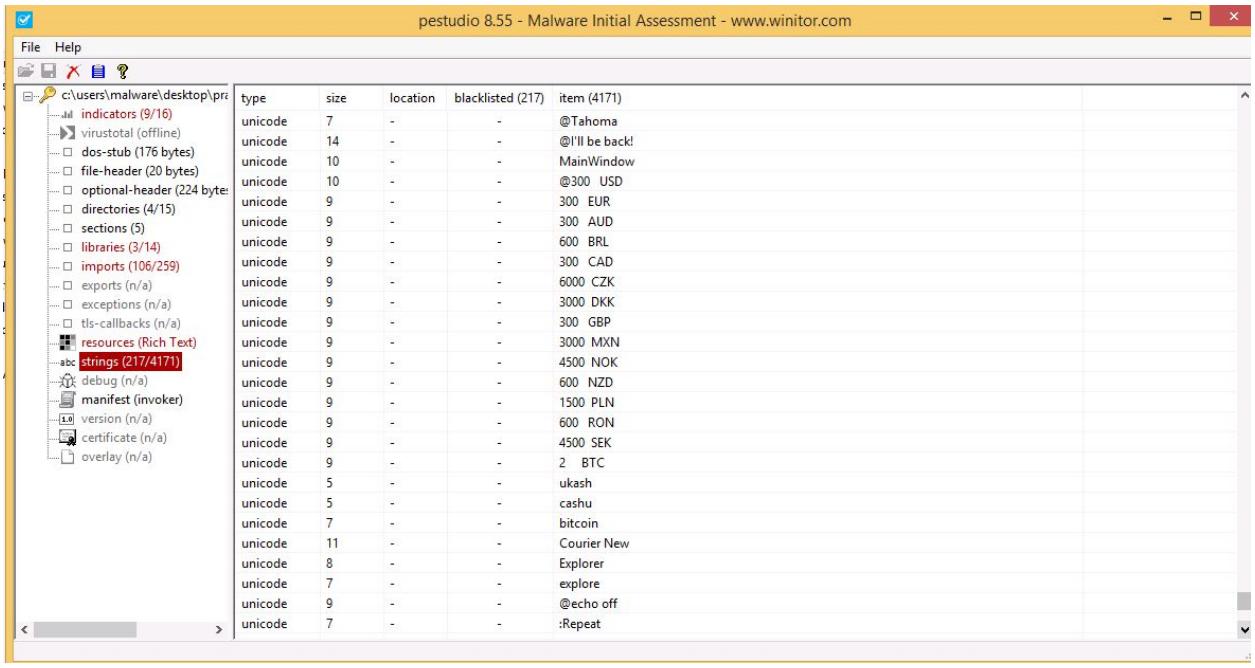

Fig.17 Currencies and amount

Basic Dynamic Analysis

Cryptolocker when it is first run, hides its presence from the user by creating a secondary executable and then deleting the initial executable file from the system. The secondary executable is hidden in the system and the typical user is not aware of its existence until it becomes active. The malware creates the secondary file in either \AppData or \LocalAppData as shown below. The malware initially keeps pinging a large number of command and control servers in the hopes of establishing a connection with an active one. It supposedly gets the list of urls to attempt connection to from its Domain Generation Algorithm (DGA). Once it successfully establishes an HTTP connection to one of these C2 servers, it then identifies itself to the server which generates a Public-Private key pair for use by it. The malware then becomes active and encrypts the user's files using the public key and holds the private key for a limited amount of time for ransom.

Cryptolocker also creates an autorun registry key shown below as "HKU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"CryptoLocker" : <filepath>" to ensure persistence of the malware across reboots. The malware also stores configuration data in "HKU\SOFTWARE\CryptoLocker" like the public key received from the C2 server.

It is also observed that the executable files use a random filename of 32 characters like {7EFA68C6-086B-43E1-A2D2-55A113531240}.

From the dynamic analysis, we see that the DGA produces a large number of domain names across seven possible Top-Level-Domains: net, biz, com, org, info, ru and co.uk. The domain names generated are 15 alphabetical characters long and are provided below. This is possibly due to the frequent shift of the C2 domains used by malware hosts to avoid detection or compromise. Further aspects of the malware are discussed in the following sections.

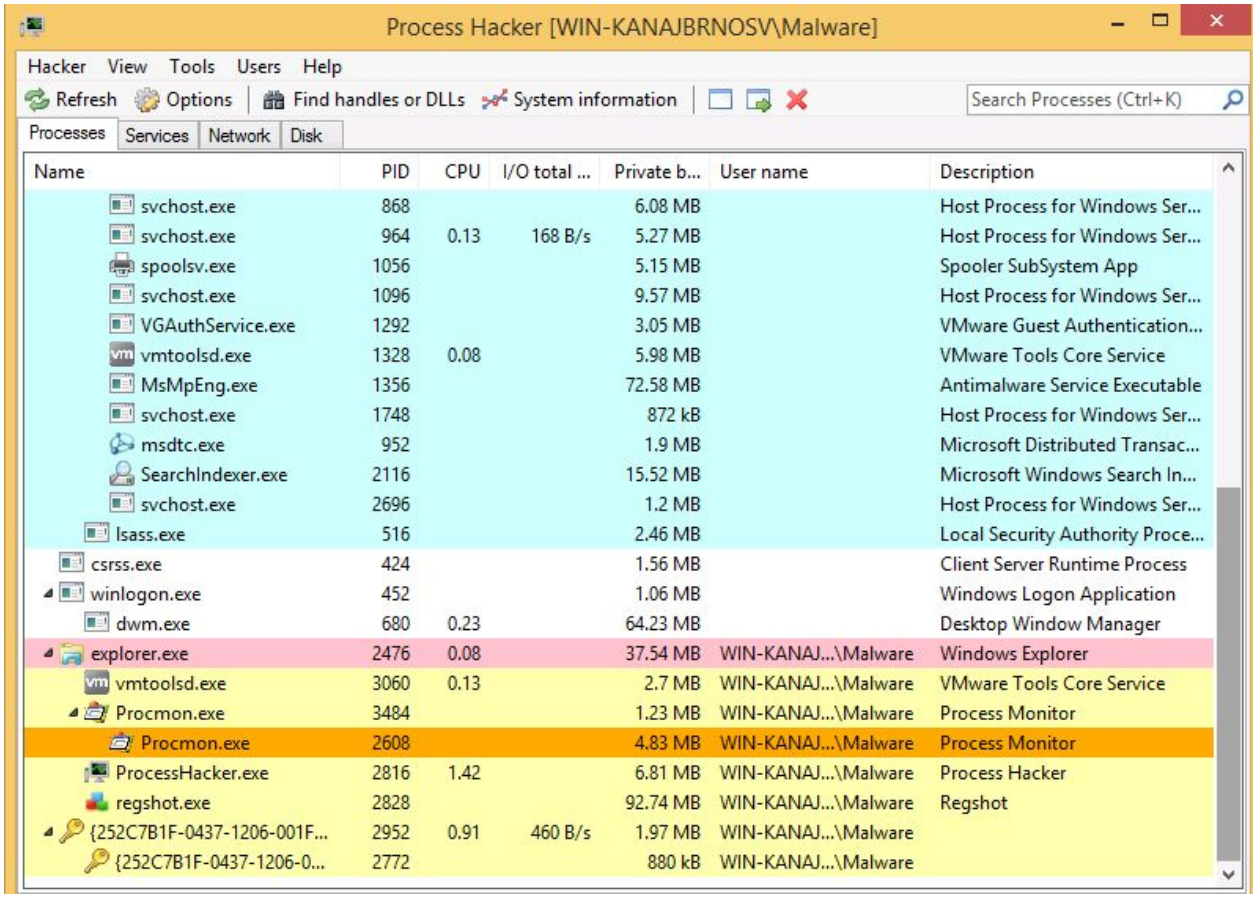

Fig.18.1 Subprocess running under main process

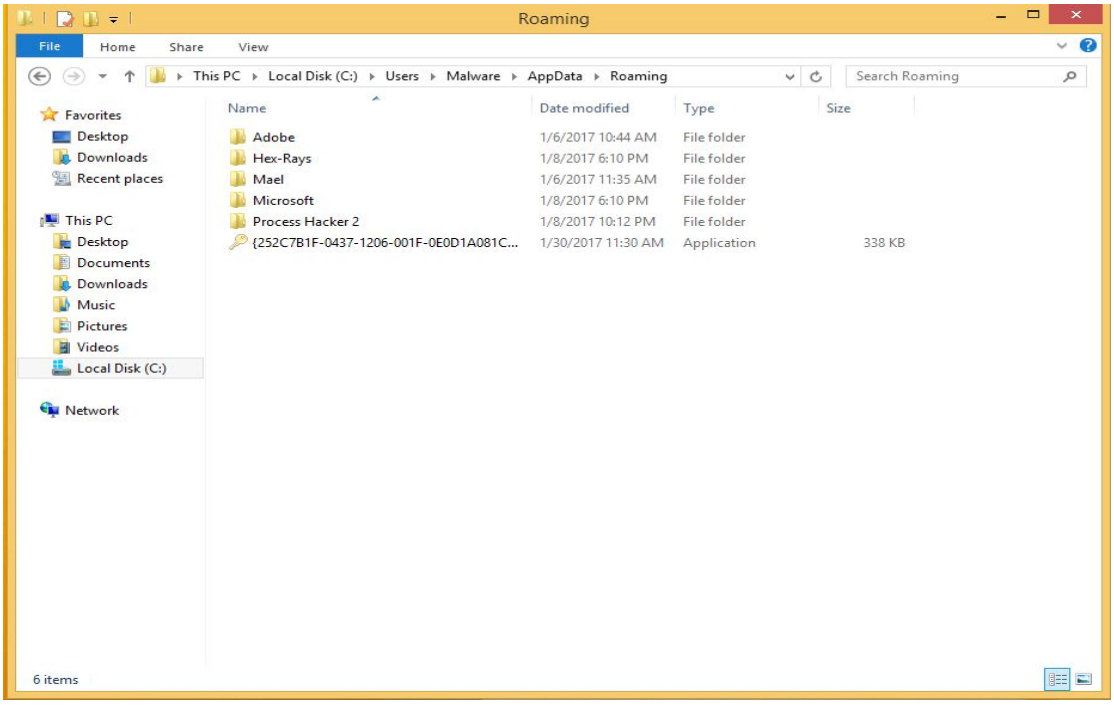

Fig.18.2 Hidden secondary executable

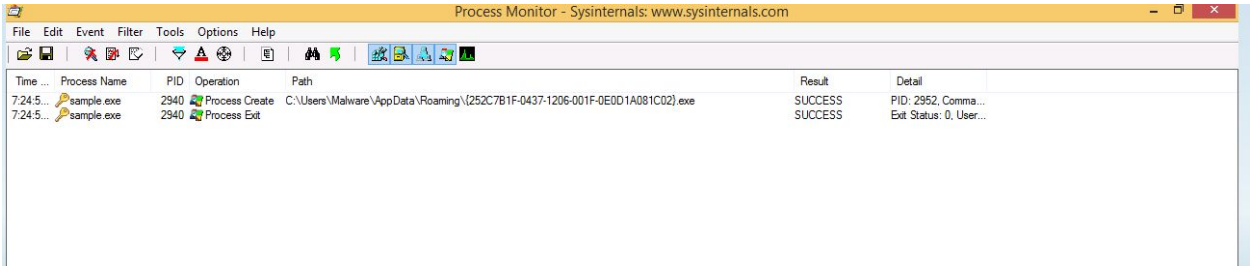

Fig.19 Creation of secondary executable and deletion of first executable

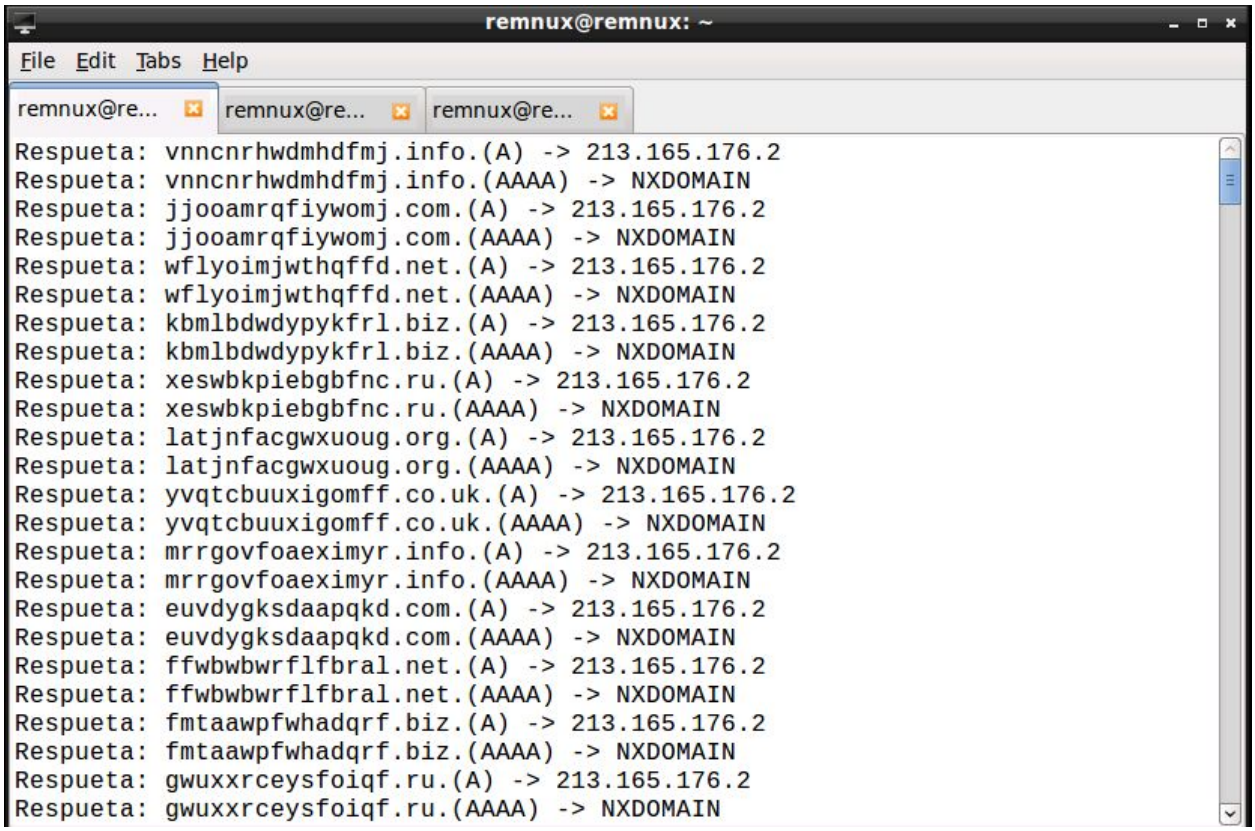

Fig.20 Attempting connection to urls generated by DGA

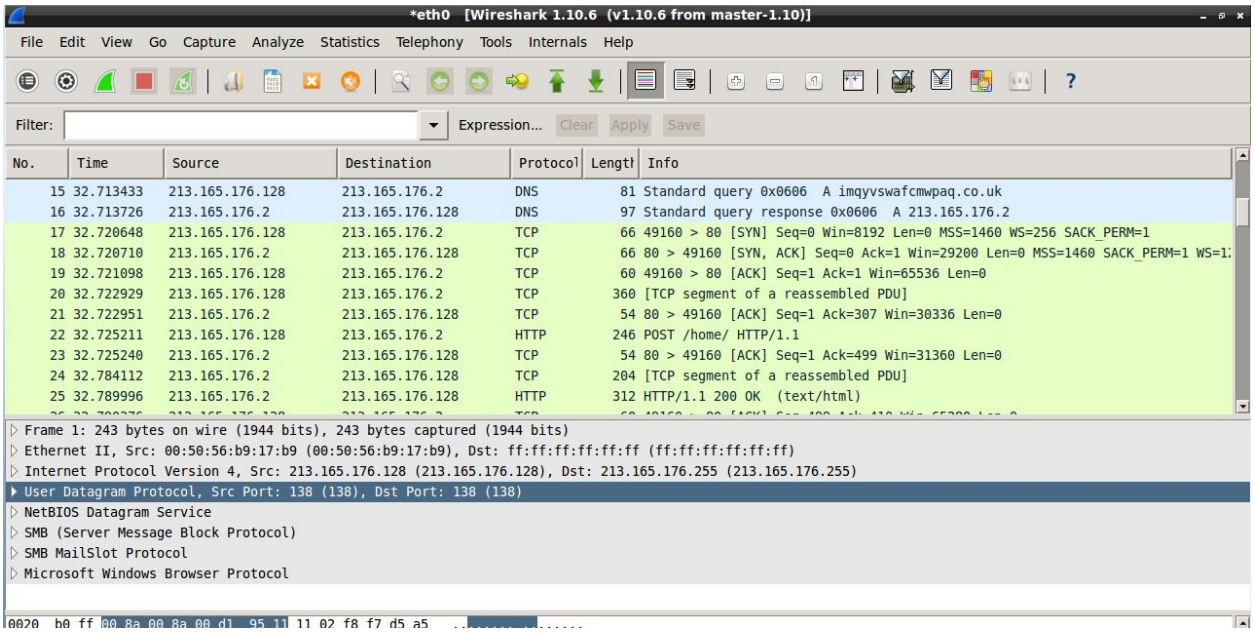

Fig.21 Attemption TCP and HTTP connection to the url.

```
Regshot 1.8.3-beta1V5
Comments:
Datetime:2017/2/20 12:21:40  ,  2017/2/20 12:36:49
Computer:WIN-KANAJBRNOSV , WIN-KANAJBRNOSV
Username:Malware , Malware

----------------------------------
Keys added:7
----------------------------------
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Modules\GlobalSettings\Sizer
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search\Preferences
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search\PrimaryProperties
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search\PrimaryProperties\UnindexedLocations
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\118\Shell\Inherit
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001_Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\118\Shell\Inherit

----------------------------------
```

Fig.22 Registry keys added

```
------------------------------------
Values added:31
------------------------------------
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Modules\GlobalSettings\Sizer\PageSpaceControlSizer: A0
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Coun
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search\PrimaryProperties\UnindexedLocations\SearchOnly
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search\Preferences\AutoWildCard: 0x00000001
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search\Preferences\EnableNaturalQuerySyntax: 0x0000000
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search\Preferences\WholeFileSystem: 0x00000000
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search\Preferences\SystemFolders: 0x00000001
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Explorer\Search\Preferences\ArchivedFiles: 0x00000000
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Run\CryptoLocker: "C:\Users\Malware\AppData\Roaming\{252C7B1F-0
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{7EFA68C6-086B-43E1-A2D2-55A113531240}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{F81E9010-6EA4-11CE-A7FF-00AA003CA9F6}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{1F2E5C40-9550-11CE-99D2-00AA006E086C}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{4A7DED0A-AD25-11D0-98A8-0800361B1103}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{596AB062-B4D2-4215-9F74-E9109B0A8153}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{ECCDF543-45CC-11CE-B9BF-0080C87CDBA6}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{85BBD920-42A0-1069-A2E4-08002B30309D}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{748F920F-FB24-4D09-B360-BAF6F199AD6D}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{0E32D9F9-10FB-4B90-8B24-826B07B084D0}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{69375D35-B310-40FD-A7DC-9548E1DBC3B5}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{35A5E985-12E6-46EE-B385-E887F3940FB0}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{9C07355E-C50A-45D2-B4A3-0A8235F8047F}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{BBB9389D-CB46-4855-ADA6-4570E7342B3C}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{C139C040-532D-451B-80CA-44D1F0839D2A}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows\CurrentVersion\Shell Extensions\Cached\{E2765AC3-564C-40F9-AC12-CD393FBAAB0F}
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store\C:\Users\Malwar
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Classes\Local Settings\MuiCache\18\52C64B7E\@zipfldr.dll,-10530: "Zip"
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Classes\Local Settings\MuiCache\18\52C64B7E\@explorerframe.dll,-14418: "Fax"
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\118\Shell\FolderType: "Generic"
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001_Classes\Local Settings\MuiCache\18\52C64B7E\@zipfldr.dll,-10530: "Zip"
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001_Classes\Local Settings\MuiCache\18\52C64B7E\@explorerframe.dll,-14418: "Fax"
HKU\S-1-5-21-2312279452-2117245222-1845265772-1001_Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags\118\Shell\FolderType: "Generic"
```

Fig.23 Registry values added

Fig.24 Registry Values modified

Further Static and Dynamic analysis

The Malware Cryptolocker, uses cryptographic libraries provided by Microsoft's CryptoAPI. It uses the "Microsoft Enhanced RSA and AES Cryptographic Provider" to create keys and to encrypt data. It is seen using IDA that the malware calls 'CryptAcquireContextW' which sets configuration data like flags and context for use by the encryption and decryption functions. The malware receives the public key to use in the encryption process from the C2 server that it establishes an HTTP connection with, which it stores in the "HKU\SOFTWARE\Cryptolocker" registry key. Once the environment is set, the program calls 'CryptEncrypt' (as shown below) to encrypt the data.

The same functionality works in reverse where the malware upon receiving payment from the user, obtains the private key from the C2 server which it stores in the same registry key. Afterwards, it makes a call to the 'CryptDecrypt' function to decrypt the user's files and return them to him.

The malware initially begins the entire encryption process by calculating the free space available for use in each drive as shown below. Once this is obtained, it makes a call to 'GetLogicalDrives' which obtains a list of all the drives on the system. Further, the 'GetDriveTypeW' function determines whether the drives are local or network based. These drives are then chosen for encryption and the files inside them that match the previously mentioned file formats are listed. The files are finally encrypted using the keys obtained from the C2 server and written back to disc. The encrypted files can only be recovered by using the private key held by the C2 server for ransom.

Furthermore, the malware appears to use command line arguments which it converts to an array of pointers to the arguments for use by the program as shown below. It is assumed that the command line arguments pertain to the GUI properties like language used, currency etc.

On investigating deeper into the malware using IDA, we see further evidence for our previous claims by seeing implementations for the different currencies and prices. Certain code is also seen that creates a new executable file using 'WriteFile' and deletes the existing file using 'DeleteFileW'. We assume this corresponds to the malware hiding its presence by deleting itself and creating a secondary executable for its functioning.

Finally, we also witness a lot of HTTP implementations as shown below. 'WinHttpOpenRequest' and 'WinHttpAddRequestHeaders' opens an HTTP request and modifies the package headers. Once this is done a 'WinHttpSendRequest' launches the request to the target domain. 'WinHttpWriteData' and 'WinHttpReceiveResponse' handle the response received from the domain. This adds further evidence to our initial statement that the malware receives the public-private key pair from the C2 server it connects to.

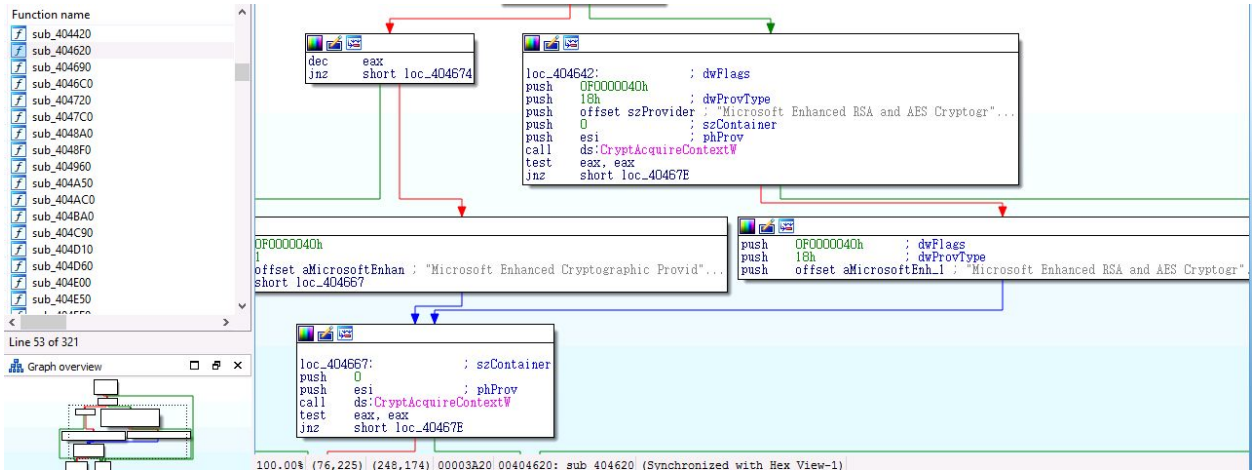

Fig.25 Setting environment for cryptographic libraries

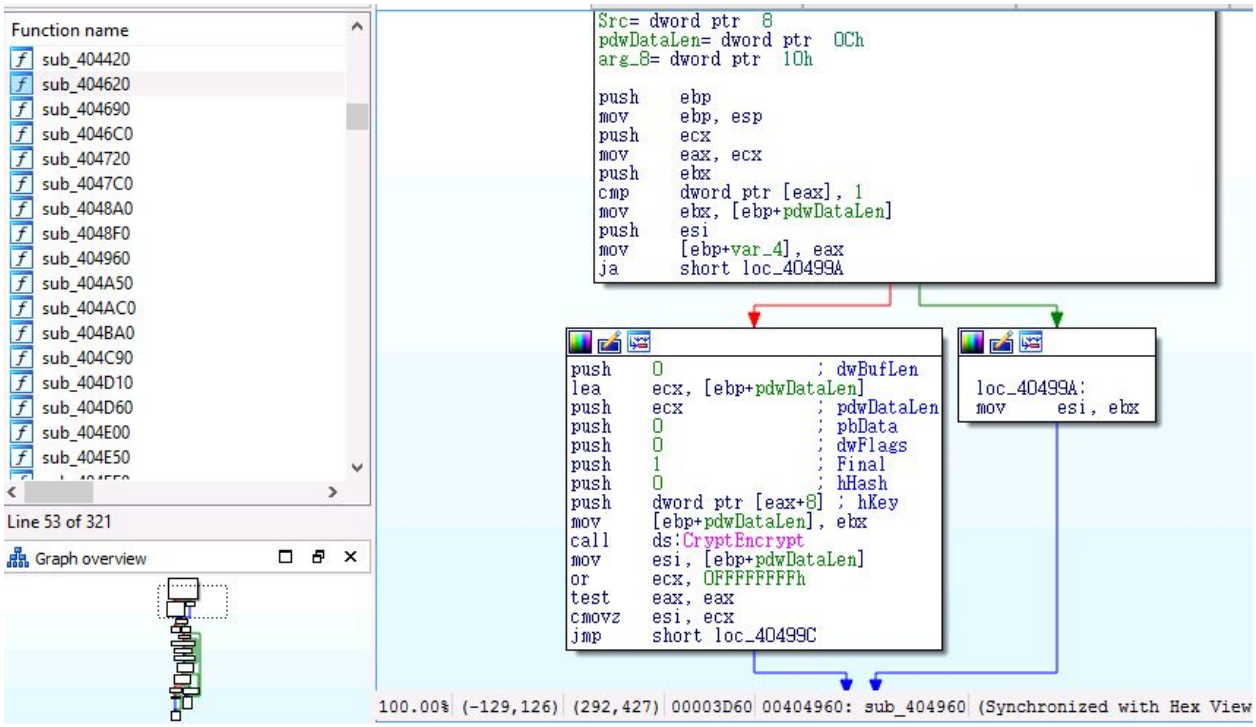

Fig.26 Encryption using CryptEncrypt

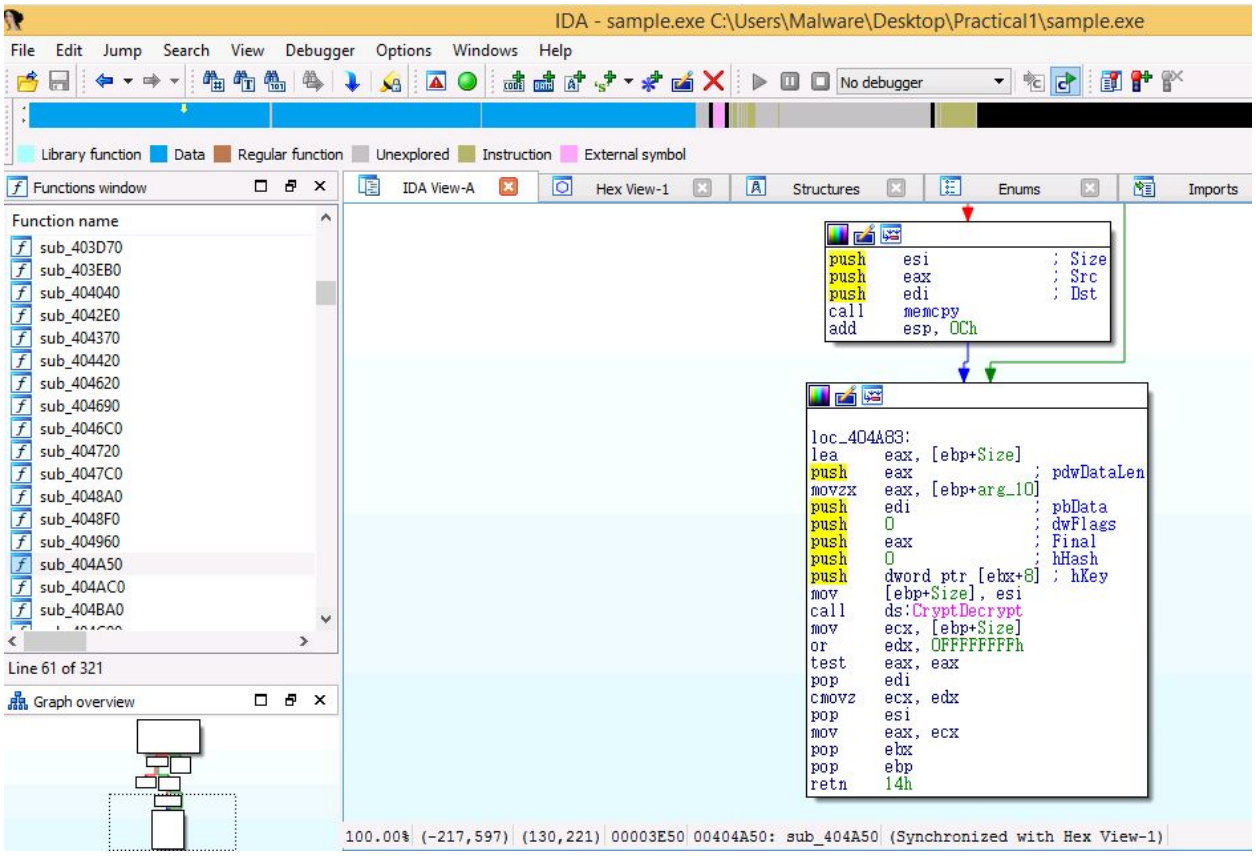

Fig.27 Decryption using CryptDecrypt

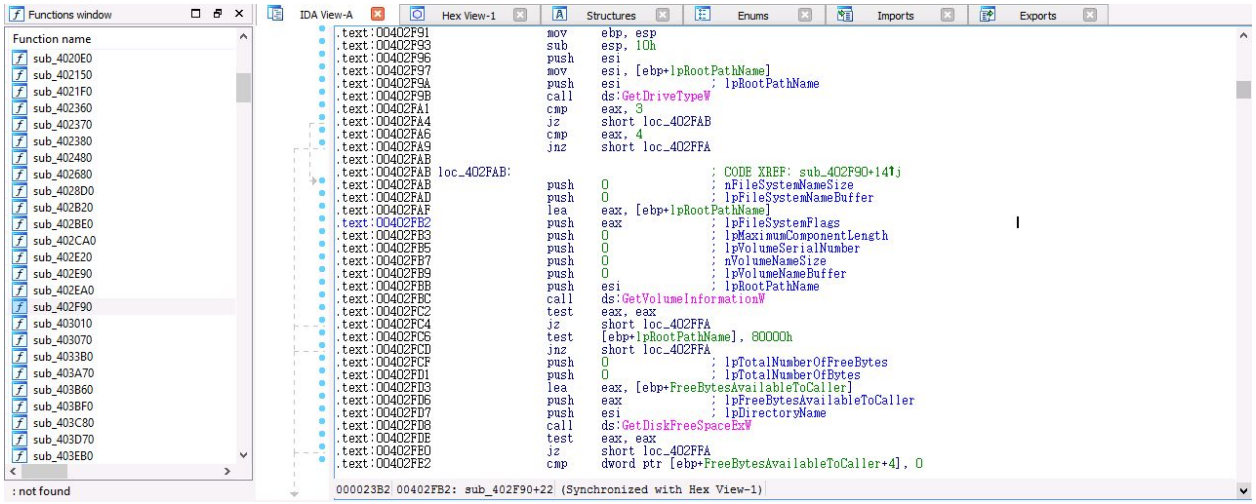

Fig.28.1 Acquiring Free Space

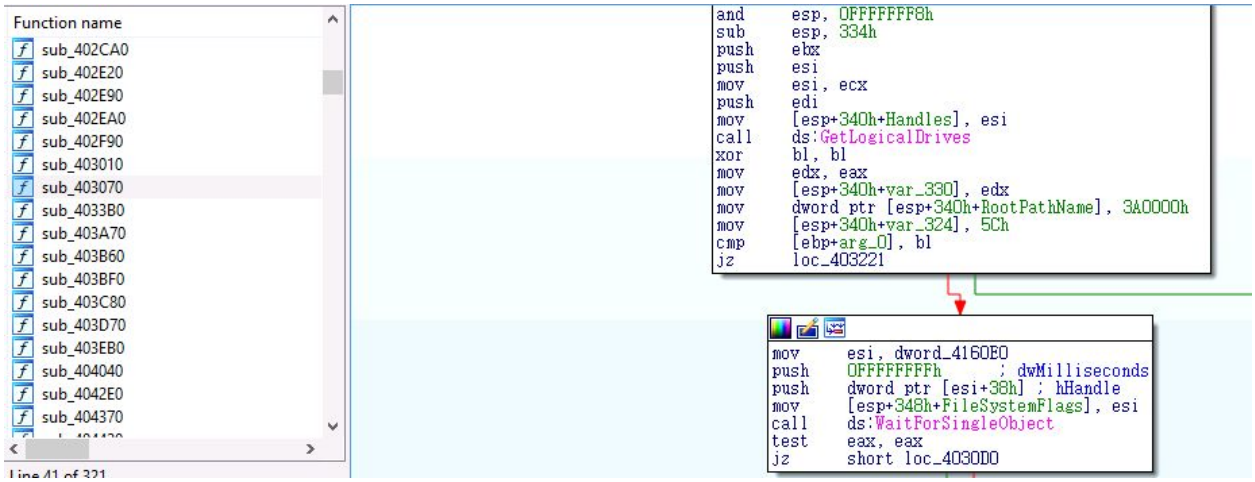

Fig.28.2 Calling GetLogicalDrives

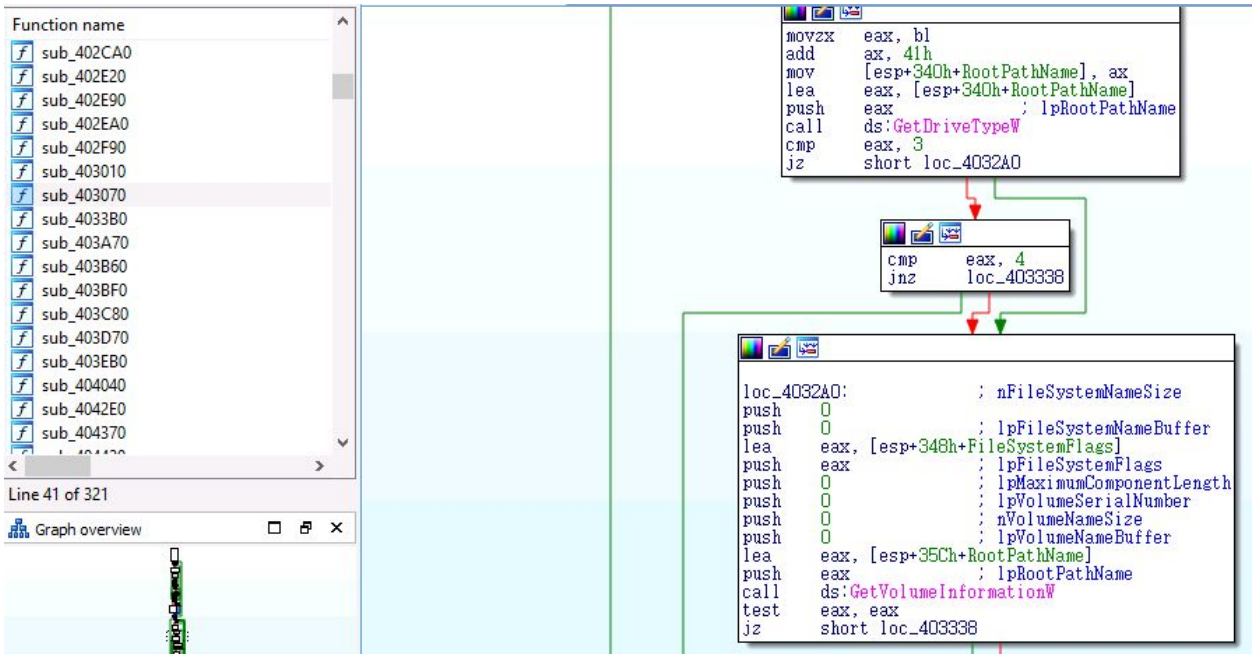

Fig.28.3 Calling GetDriveTypeW after

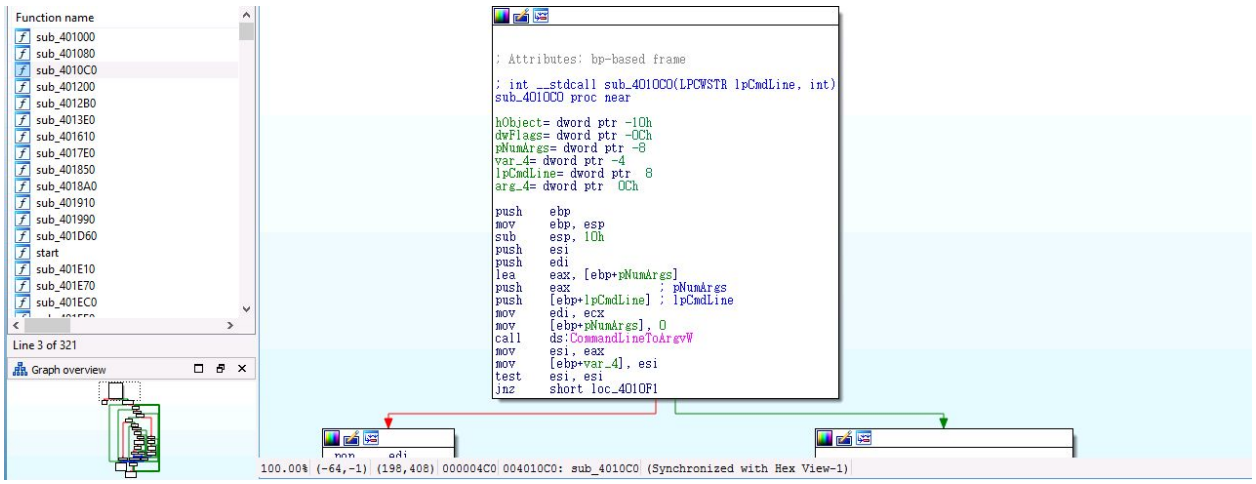

Fig.29 CommandLineToArgvW

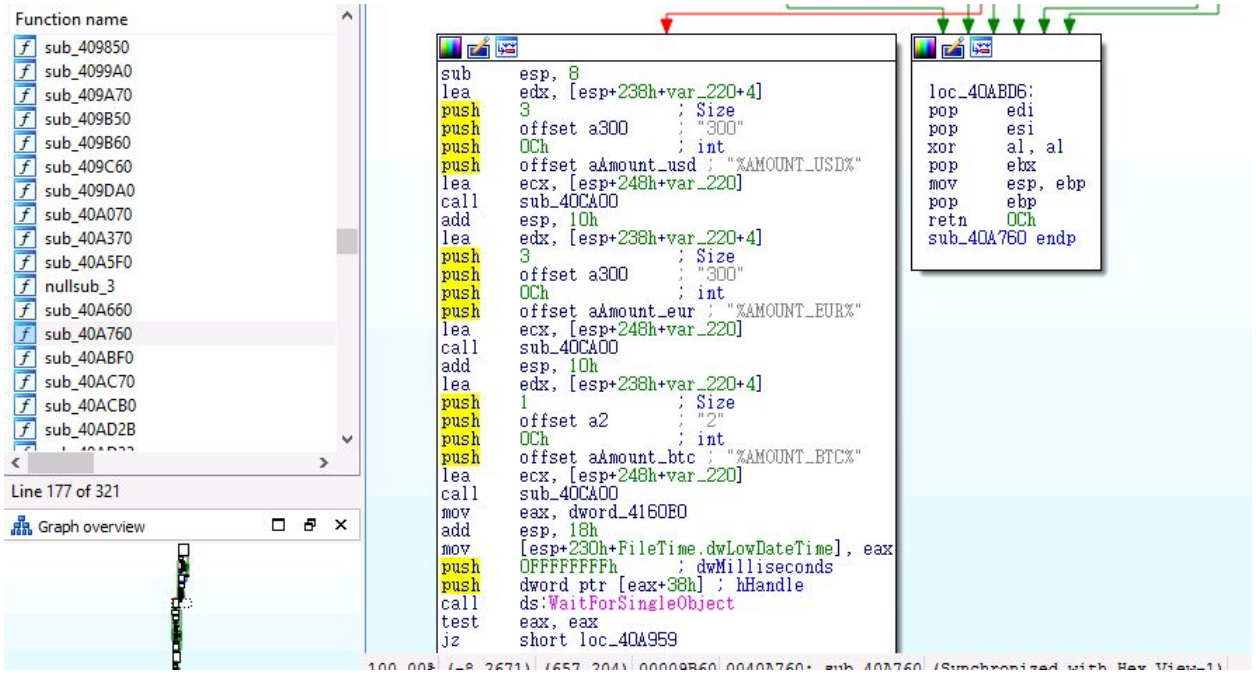

Fig.30 Currencies and prices

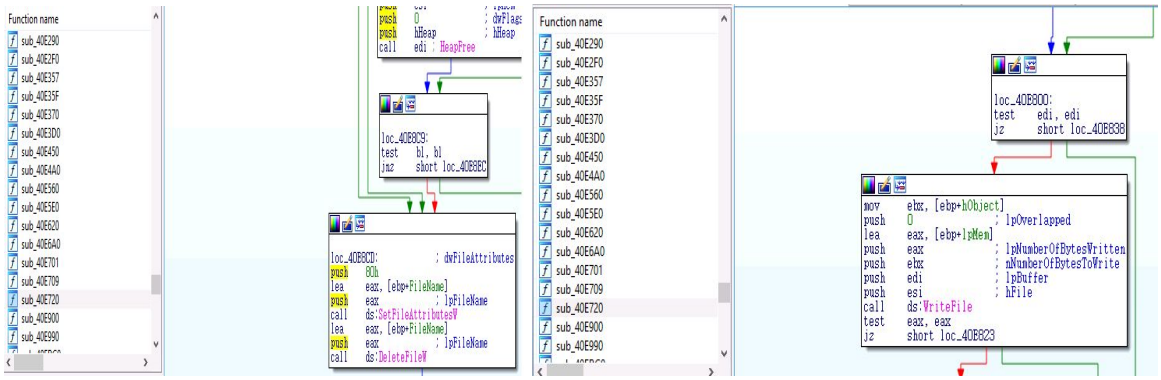

Fig.31 Creation and Deletion of executables

Fig.32 Sending Http Request

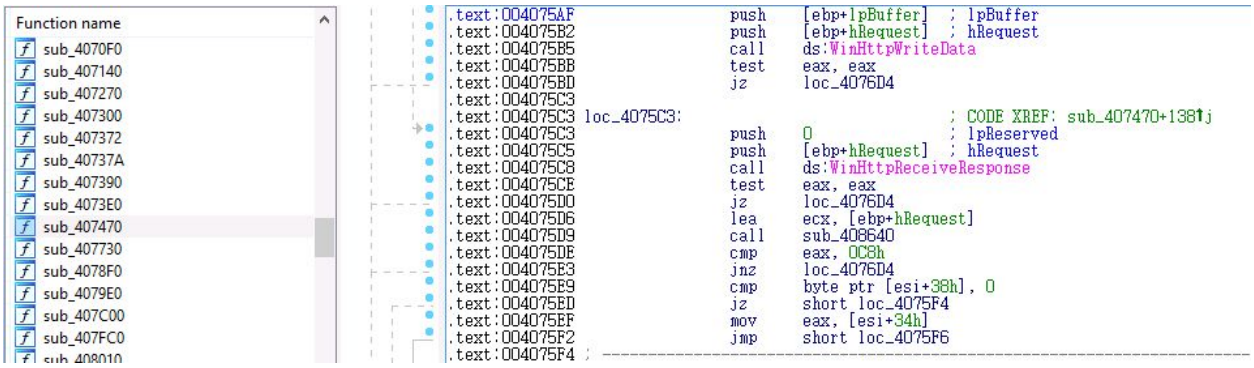

Fig.33 Receiving Http Response

Indicators of compromise

The malware exhibits a large number of network based indicators. The DGA requires the malware to continuously attempt connections to a large number of urls generated by it. These urls belong to one of seven TLDs : org, com, biz, info, co.uk, ru and net. They are also 15 alphabetical characters in size and mostly appear in random order. If one observes his system attempting a large number of connections to domains that match the aforementioned pattern, then it can serve as a strong indicator of compromise.

Further, a variety of host based indicators can be found by analysing the registry and the file system. Cryptolocker initially creates an autorun registry key to ensure persistence across system reboots while the malware attempts to contact an active C2 server. This registry key “HKU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run “CryptoLocker” : <Filepath>” can be used to identify the existence of the malware if the user locates it before the program establishes connection to an active server. Moreover, the registry key “HKU\SOFTWARE\CryptoLocker” stores configuration files pertaining to the malware which can again be used as an indicator.

Finally, the secondary executable created by the malware portrays a pattern in its naming. In particular, it consists of 32 random alphanumeric characters. If the user can locate hidden system files on his drives, then the identification of this executable serves as a strong indicator of potential compromise.

Conclusion

Cryptolocker is a dangerous ransomware that is hard to mitigate once activated, hence prevention is definitely a critical priority. It uses very potent encryption algorithms and has a very efficient distribution system owing to the GameOver Zeus botnet and the pervasiveness of spam Email. Periodic backup of system files can help by facilitating system formats followed by system restores.