Informe Código Dañino CCN-CERT ID-01/17

Ransom.VenusLocker

Febrero 2017

CENTRO CRIPTOLOGICO NACIONAL
2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC,
ou=sello electrónico, serialNumber=S2800155J,
o=CENTRO CRIPTOLOGICO NACIONAL, cn=CENTRO
CRIPTOLOGICO NACIONAL, c=ES
2017.02.01 15:49:26 +01'00'

LIMITACIÓN DE RESPONSABILIDAD

El presente documento se proporciona de acuerdo con los términos en él recogidos, rechazando expresamente cualquier tipo de garantía implícita que se pueda encontrar relacionada. En ningún caso, el Centro Criptológico Nacional puede ser considerado responsable del daño directo, indirecto, fortuito o extraordinario derivado de la utilización de la información y software que se indican incluso cuando se advierta de tal posibilidad.

AVISO LEGAL

ÍNDICE

1. SOBRE CCN-CERT

El CCN-CERT (www.ccn-cert.cni.es) es la Capacidad de Respuesta a Incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN (www.ccn.cni.es). Este servicio se creó en el año 2006 como el **CERT Gubernamental/Nacional** español y sus funciones quedan recogidas en la Ley 11/2002 reguladora del Centro Nacional de Inteligencia, el RD 421/2004 regulador del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad (ENS), modificado por el RD 951/2015 de 23 de octubre.

De acuerdo a todas ellas, es competencia del CCN-CERT la gestión de ciberincidentes que afecten a **sistemas clasificados**, del **Sector Público** y de **empresas y organizaciones de interés estratégico** para el país. Su misión, por tanto, es contribuir a la mejora de la ciberseguridad española, siendo el centro de alerta y respuesta nacional que coopere y ayude a responder de forma rápida y eficiente a los ciberataques y a afrontar de forma activa las ciberamenazas.

2. RESUMEN EJECUTIVO

El presente documento recoge el análisis del código dañino ***Ransom.VenusLocker*** que ha sido diseñado para comunicarse con un dominio de Internet, cifrar ciertos archivos y extorsionar a la víctima mostrando una notificación sobre el procedimiento de pago para rescatar los archivos cifrados.

El código dañino está compilado en .NET y no es compatible con Windows XP e inferiores aunque estos tengan la versión 3.5 de .NET instalada debido a ciertos fallos de programación.

3. INFORMACIÓN DE VERSIONES DEL CÓDIGO DAÑINO

Dicho binario solo tiene una versión.

3.1 EXTENSIONES A CIFRAR

El código dañino tiene dos tablas de archivos a considerar como objetivo para ser cifrados. La primera indica las extensiones de los archivos que serán cifrados en su totalidad, la segunda contiene las extensiones de los archivos que serán cifrados solamente los primeros 512 bytes. Esta característica no es normal en el tipo de código dañino *ransomware* donde el cifrado se suele producir siempre en la totalidad del archivo.

Esta es la tabla de archivos que cifra totalmente. Es interesante ver como la extensión "*xlsb*" no tiene el punto incluido por lo que no funcionaría.

.txt	**.doc**	**.xlsx**	**.potx**
.ini	**.dot**	**.xlsm**	**.potm**
.php	**.docx**	**.xltx**	**.ppam**
.html	**.docm**	**.xltm**	**.ppsx**
.css	**.dotx**	xlsb	**.ppsm**
.py	**.dotm**	**.xla**	**.sldx**
.c	**.rtf**	**.xlam**	**.sldm**
.cpp	**.wpd**	**.xll**	**.class**
.cc	**.docb**	**.xlw**	**.jar**
.h	**.wps**	**.ppt**	**.csv**
.cs	**.msg**	**.pot**	**.xml**
.log	**.xls**	**.pps**	**.dwg**
.pl	**.xlt**	**.pptx**	**.dxf**
.java	**.xlm**	**.pptm**	**.asp**

Archivos que solo cifra sus primeros 512 bytes:

.asf	.avs	.bdr	.war	.scx	.pot	.so	.bay
.pdf	.bik	.bib	.xpi	.sdt	.potx	.swd	.crw
.xls	.dir	.boc	.z02	.spr	.pptm	.tpu	.cr2
.docx	.divx	.crd	.z04	.sud	.psa	.tpx	.dcr
.xlsx	.dvx	.diz	.zap	.uax	.qdf	.tu	.kdc
.mp3	.evo	.dot	.zipx	.umx	.qel	.tur	.erf
.waw	.flv	.dotm	.zoo	.unr	.rgn	.vc	.mef
.jpg	.qtq	.dotx	.ipa	.uop	.rrt	.yab	.mrw
.jpeg	.tch	.dvi	.isu	.usa	.rsw	.aip	.nef
.txt	.rts	.dxe	.jar	.usx	.rte	.amxx	.nrw
.rtf	.rum	.mlx	.js	.ut2	.sdb	.ape	.orf
.doc	.rv	.err	.udf	.ut3	.sdc	.api	.raf
.rar	.scn	.euc	.adr	.utc	.sds	.mxp	.rwl
.zip	.srt	.faq	.ap	.utx	.sql	.oxt	.rw2
.psd	.stx	.fdr	.aro	.uvx	.stt	.qpx	.r3d
.tif	.svi	.fds	.asa	.uxx	.tcx	.qtr	.ptx
.wma	.swf	.gthr	.ascx	.vmf	.thmx	.xla	.pef
.gif	.trp	.idx	.ashx	.vtf	.txd	.xlam	.srw
.bmp	.vdo	.kwd	.asmx	.w3g	.txf	.xll	.x3f
.ppt	.wm	.lp2	.asp	.w3x	.upoi	.xlv	.der
.pptx	.wmd	.ltr	.indd	.wtd	.vmt	.xpt	.pem
.docm	.wmmp	.man	.asr	.wtf	.wks	.cfg	.pfx
.xlsm	.wmx	.mbox	.qbb	.ccd	.wmdb	.cwf	.p12
.pps	.wvx	.msg	.bml	.cd	.xl	.dbb	.p7b
.ppsx	.xvid	.nfo	.cer	.cso	.xlc	.slt	.p7c
.ppd	.3d	.now	.cms	.disk	.xlr	.bp2	.jfif
.eps	.3d4	.odm	.crt	.dmg	.xlsb	.bp3	.exif
.png	.3df8	.oft	.dap	.dvd	.xltx	.bpl	.docb
.ace	.pbs	.pwi	.htm	.fcd	.ltm	.clr	.xlt
.djvu	.adi	.rng	.moz	.flp	.xlwx	.dbx	.xltm
.tar	.ais	.rtx	.svr	.img	.mcd	.jc	.xlw
.cdr	.amu	.run	.url	.isz	.cap	.potm	.ppam
.max	.arr	.ssa	.wdgt	.mdf	.cc	.ppsm	.sldx
.wmv	.bmc	.text	.abk	.mds	.cod	.prc	.sldm

.avi	.bmf	.unx	.bic	.nrg	.cp	.prt	.class
.wav	.cag	.wbk	.big	.nri	.cpp	.shw	.db
.mp4	.cam	.wsh	.blp	.vcd	.cs	.std	.pdb
.pdd	.dng	.7z	.bsp	.vhd	.csi	.ver	.dat
.php	.ink	.arc	.cgf	.snp	.dcp	.wpl	.csv
.aac	.ini	.ari	.chk	.bkf	.dcu	.xlm	.xml
.ac3	.jif	.arj	.col	.ade	.dev	.yps	.spv
.amf	.jiff	.car	.cty	.adpb	.dob	.1cd	.grle
.amr	.jpc	.cbr	.dem	.dic	.dox	.bck	.sv5
.dwg	.jpf	.cbz	.elf	.cch	.dpk	.html	.game
.dxf	.jpw	.gz	.ff	.ctt	.dpl	.bak	.slot
.accdb	.mag	.gzig	.gam	.dal	.dpr	.odt	.aaf
.mod	.mic	.jgz	.grf	.ddc	.dsk	.pst	.aep
.tax2013	.mip	.pak	.h3m	.ddcx	.dsp	.log	.aepx
.tax2014	.msp	.pcv	.h4r	.dex	.eql	.mpg	.plb
.oga	.nav	.puz	.iwd	.dif	.ex	.mpeg	.prel
.ogg	.ncd	.rev	.ldb	.dii	.f90	.odb	.prproj
.pbf	.odc	.sdn	.lgp	.itdb	.fla	.wps	.eat
.ra	.odi	.sen	.lvl	.itl	.for	.xlk	.ppj
.raw	.opf	.sfs	.map	.kmz	.fpp	.mdb	.indl
.saf	.qif	.sfx	.md3	.lcd	.jav	.dxg	.indt
.val	.xwd	.sh	.mdl	.lcf	.java	.wpd	.indb
.wave	.abw	.shar	.nds	.mbx	.lbi	.wb2	.inx
.wow	.act	.shr	.pbp	.mdn	.owl	.dbf	.idml
.wpk	.adt	.sqx	.ppf	.odf	.pl	.ai	.pmd
.3g2	.aim	.tbz2	.pwf	.odp	.plc	.3fr	.xqx
.3gp	.ans	.tg	.pxp	.ods	.pli	.arw	.svg
.3gp2	.asc	.tlz	.sad	.pab	.pm	.srf	.as3
.3mm	.ase	.vsi	.sav	.pkb	.res	.sr2	.as
.amx	.bdp	.wad	.scm	.pkh	.rsrc		

3.2 EXTENSIÓN AÑADIDA A LOS ARCHIVOS CIFRADOS

El código dañino cambia la extensión de los archivos a alguna de las siguientes:

.Venusf
.Venusp

La extensión ".Venusf" se utiliza para los archivos cifrados completamente y ".Venusp" para los cifrados parcialmente.

3.3 ARCHIVOS DE RESCATE

Este binario modifica el fondo de pantalla del escritorio y crea un archivo de texto con información acerca del secuestro y el procedimiento a seguir para recuperarlo.

Un ejemplo del archivo de texto sería:

---------- Venus Locker ----------

Unfortunately, you are hacked.

1. What happened to my files?

Your personal files, including your photos, documents, videos and other important files on this computer, have been encrypted
with RSA-4096, a strong encryption algorithm. RSA algorithm generates a public key and a private key for your computer. The
public key was used to encrypt your files a moment ago. The private key is necessary for you to decrypt and recover your files.
Now, your private key is stored on our secret Internet server. And there is no doubt that no one can recover your files without
your private key.

For further information about RSA algorithm, please refer to https://en.wikipedia.org/wiki/RSA_(cryptosystem)

2. How to decrypt my files?

To decrypt and recover your files, you have to pay 100 US Dollars for the private key and decryption service. Please note that
you have ONLY 72 HOURS to complete your payment. If your peyment do not be completed within time limit, your private key will be
deleted automatically by our server. All your files will be permanently encrypted and nobody can recover them. Therefore, it is
advised that you'd better not waste your time, because there is no other way to recover your files except making a payment.

3. How to pay for my private key?

There are three steps to make a payment and recover your files:

1). For the security of transactions, all the payments must be completed via Bitcoin network. Thus, you need to exchange 100 US dollars
(or equivalent local currencies) to Bitcoins, and then send these Bitcoins (about 0.15 BTC) to the following address.

1Dj9YnMiciNgaKuyzKynygu7nB21tvV6QD

2). Send your personal ID to our official email: VenusLocker@mail2tor.com

Your personal ID is <identificador único de la víctima>

3). You will receive a decryptor and your private key to recover all your files within one working day.

4. What is Bitcoin?

Bitcoin is an innovative payment network and a new kind of money. It is based on an open-source cryptographic protocol that is independent of any central authority. Bitcoins can be transferred through a computer or a smartphone withour an intermediate financial institution.

5. How to make a payment with Bitcoin?

You can make a payment with Bitcoin based on Bitcoin Wallet or Based on Perfect Money. You can choose the way that is more convenient for you.

About Based on Bitcoin Wallet

1) Create a Bitcoin Wallet. We recommend Blockchain.info (https://blockchain.info/)

2) Buy necessary amount of Bitcoins. Our recommendations are as follows.
LocalBitcoins.com -- the fastest and easiest way to buy and sell Bitcoins.
CoinCafe.com -- the simplest and fastest way to buy, sell and use Bitcoins.
BTCDirect.eu -- the best for Europe.
CEX.IO -- Visa / MasterCard
CoinMama.com -- Visa / MasterCard
HowToBuyBitcoins.info -- discover quickly how to buy and sell Bitcoins in your local currency.

3) As mentioned above, send about 0.15 BTC (equivalent to 100 USD) to our Bitcoin receiving address.

4) As mentioned above, and then, send us your personal ID via email, you will receive your private key soon.

About Based on Perfect Money

1) Create a Perfect Money account. (https://perfectmoney.is)

2) Visit to PMBitcoin.com. (https://pmBitcoin.com/btc)
input our Bitcoin receiving address in the "Bitcoin Wallet" textbox.
input 100 in the "Amount" textbox, the amount of Bitcoin will be calculated automatically.
click "PAY" button, then you can complete you payment with your Perfect Money account and local debit card.

6. If you have any problem, please feel free to contact us via official email.

Best Regards
VenusLocker Team

La muestra se conecta a una URL para obtener la imagen que debe modificar el fondo del escritorio. En caso de no conseguir la imagen, se deja el fondo vacío.

http://i.imgur.com/Jk67LrS.jpg

También muestra una interfaz de usuario con información del secuestro, el tiempo que queda para poder recuperar los archivos y la dirección *Bitcoin* donde efectuar el pago.

Ilustración 1. Interfaz de usuario con información sobre el secuestro

4. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

El binario examinado posee las siguientes características:

- Carga el código dañino en el sistema.
- Crea un archivo en el sistema como marca de infección.
- Cifra todos los archivos que cumplan un patrón de extensión ya sea parcialmente o de forma completa.
- Cambia el nombre de los archivos por su equivalente en Base64 y cambia su extensión.
- Se comunica con el servidor de mando y control (C2) y realiza un envío de datos del sistema infectado y la clave de descifrado.
- Está compilado en .NET
- Muestra la interfaz acerca del secuestro de los archivos y los pasos a seguir para poder recuperarlos.

5. DETALLES GENERALES

La muestra analizada se corresponde con la siguiente firma MD5:

8675ffb697ad944748e0e24ac1a962ce

El binario tiene formato *Portable Executable* (PE), es decir, es un ejecutable para sistemas operativos Windows, concretamente para 32 bits.

En la muestra analizada se ha podido observar que la fecha interna de creación del programa data del 29 de julio de 2016.

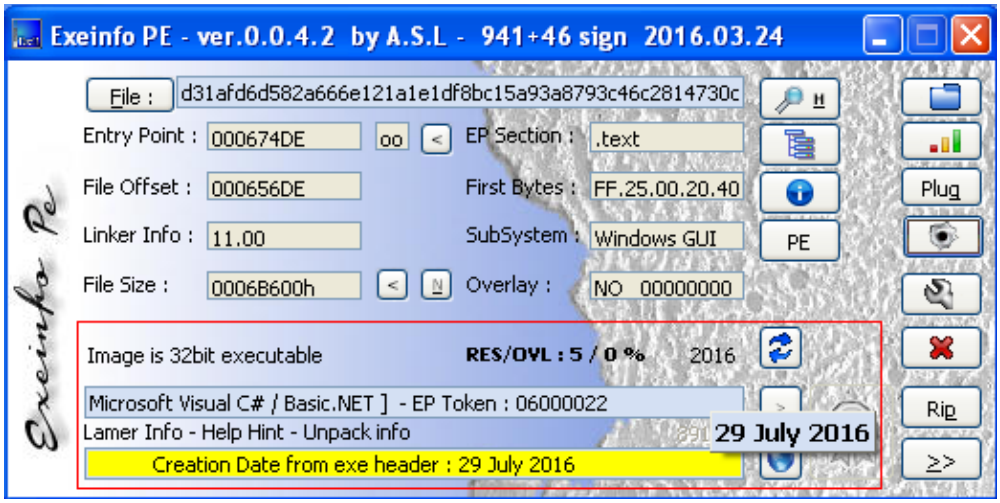

Ilustración 2. Detalles del binario

6. PROCEDIMIENTO DE INFECCIÓN

La infección en el equipo se produce al ejecutar el fichero que contiene el código y realiza las siguientes acciones:

- Comprueba que no exista un archivo en el sistema como marca de infección previa. En este caso, lo crea.
- Se comunica con su servidor C2 y envía información acerca del sistema comprometido.
- Cifra las unidades de disco duro del sistema comprometido y vuelve a comunicar con su servidor C2 para enviarle la clave usada en el proceso de cifrado. La clave se cifra previamente con el algoritmo RSA.
- Cambia el fondo de pantalla del escritorio.
- Crea el mensaje de texto con la información acerca del secuestro de los archivos y el procedimiento a seguir.
- Muestra la interfaz de usuario con la información del secuestro, procedimiento a seguir, dirección *Bitcoin* donde efectuar el pago, el tiempo que queda para poder pagar y el identificador único de la víctima.

7. CARACTERÍSTICAS TÉCNICAS

El código dañino está programado en .NET con lo que es trivial poder de-compilarlo y obtener su código fuente. No utiliza ningún tipo de ofuscación ni protección en los ensamblados de .NET. Tampoco va incluido en un *dropper* inicial, con lo que la ejecución es más rápida y se puede encontrar el código dañino en disco.

El principal problema del código dañino, y lo que hace que no sea compatible con sistemas operativos inferiores al Windows Vista, es que tiene todas sus rutas embebidas. Esto significa que asume a "C:" como directorio raíz y que existe la carpeta *Users*.

```
private string SingletonPath = "C:\\Users\\" + Environment.UserName + "\\U2FsdGVkX1DKeR.vluni";
```

La excepción no es manejada correctamente en la parte de la creación de archivos en esas rutas. Debido a esto, se muestra un mensaje de error de .NET posibilitando así cancelar la ejecución.

La primera acción que realiza el código dañino es comprobar la existencia de un archivo en la ruta base del usuario en ejecución.

```
C:\Users\<usuario>\U2FsdGVkX1DKeR.vluni
```

En caso de que el archivo exista, el código dañino se finaliza y, en caso contrario, se crea el archivo asignándole los atributos de sistema y oculto.

La siguiente acción realizada por el código dañino es obtener información del sistema comprometido y enviarla, mediante una comunicación POST, al servidor C2. Dicha comunicación utiliza el cifrado SSL con un certificado no autorizado por una Autoridad de Certificación (CA), por ello, el código dañino, a través de una instrucción concreta, acepta ese certificado dándolo por válido.

La información que obtiene del sistema comprometido es la siguiente:

- El nombre del equipo comprometido. En caso de no poder obtenerlo le asigna el valor *unknown* a la variable que almacena esta cadena.
- El nombre del usuario activo en el sistema. Del mismo modo que pasa con el nombre del equipo, en caso de no poder obtenerlo se usará el valor *unknown*.
- El lenguaje del sistema. Para calcular el lenguaje se conecta a la siguiente web:

  ```
  http://ip-api.com/csv?fields=country
  ```

  Esta web devuelve el país del sistema comprometido. En caso de no poder establecer conexión, el código dañino asume que el sistema está

en inglés. Si pudo obtener el país, crea la cadena de texto "S_<país>" y en caso de ser forzado "F_<USA>".

- El sistema operativo calculado por su versión mayor y menor. En este cálculo la cadena creada puede ser una de las siguientes:

9x	**Vista**
2000	**7**
XP	**8**
2003	**10**

En caso de no poder obtener la versión del sistema operativo usará la cadena *unknown*. Pese a que el código dañino admite poder enviar cadenas de sistemas operativos como Windows 98, 2000 o XP, por un fallo de programación, no funciona correctamente en esos sistemas operativos.

- El tiempo actual en UTC.
- Introduce la palabra *loads al final de recopilar todos los parámetros anteriores*.

Con la información nombre del equipo, nombre del usuario activo, idioma, sistema operativo y tiempo actual, calcula un HASH MD5 que es usado como identificador único de la víctima.

Tras crear el "*hash*", establece una comunicación con el servidor C2 y, mediante el método POST, envía todos los datos obtenidos y el "*hash*" MD5. Si se consigue enviar correctamente los datos, la función encargada de esto devuelve un valor "0".

Tras el envío se procede a comprobar el valor de retorno de la función: si es distinto de "0" se utiliza una clave AES embebida en el código dañino para cifrar los archivos; en caso contrario, procede a generar una clave aleatoria de 32 caracteres. Para la generación de la clave utiliza como "semilla" un objeto y método de .NET seguro.

```
public int VenusLockerRun()
{
  if (DateTime.Compare(DateTime.Now, Convert.ToDateTime("2016-08-15 00:00:00")) > 0)
    return -1;
  ServicePointManager.ServerCertificateValidationCallback += new RemoteCertificateValidationCallback(
  string UserId;
  int num = this.SendInfo(out UserId);
  this.PersonalID = UserId;
  string pwd = num != 0 ? "BGORMkj&v=u1X0O2hOybNdRvZb9SGGnm" : this.CreatePassword(32);
  this.Disk_Encryption(pwd);
  if (num == 0)
  {
    string AESKey = this.AESKeyEncryptWithRSA(pwd);
    this.SendKey(UserId, AESKey);
  }
  return 0;
}
```

Ilustración 3. Clave embebida AES en caso de no haber comunicación con el C2

Con la clave de cifrado procede a cifrar todos los discos del sistema, los recursos de red y extraíbles. Por cada archivo encontrado se comprueba que la extensión sea válida para ser cifrada y que los atributos del archivo no contengan los *flags* de sistema y oculto. Por cada directorio encontrado, se llama a la función de cifrado de forma recursiva.

Es importante tener en cuenta que el código dañino ignorará una serie de directorios a los que no accederá para cifrar archivos:

Program Files	**Microsoft Chart Controls**	**Windows NT**
Program Files (x86)	**Microsoft Games**	**Windows Media Player**
Windows	**Microsoft Office**	**Windows Mail**
Python27	**Microsoft.NET**	**NVIDIA Corporation**
Python34	**MicrosoftBAF**	**Adobe**
AliWangWang	**MSBuild**	**IObit**
Avira	**QQMailPlugin**	**AVAST Software**
Wamp	**Realtek**	**CCleaner**
Avira	**Skype**	**AVG**
360	**Reference Assemblies**	**Mozilla Firefox**
ATI	**Tencent**	**VirtualDJ**
Google	**USB Camera2**	**TeamViewer**
Intel	**WinRAR**	**ICQ**
Internet Explorer	**Windows Sidebar**	**Java**
Kaspersky Lab	**Windows Portable Devices**	**Yahoo!**
Microsoft Bing Pinyin	**Windows Photo Viewer**	

Dentro del listado se puede apreciar desde los directorios de instalación de programas de Windows a navegadores, determinados programas (como el WinRAR), programas antivirus, lenguajes de programación y programas de conversación online o de control remoto.

Por cada archivo que encuentra en un directorio permitido, y que tenga una de las extensiones admitidas, comprueba si hay que cifrarlo de forma parcial (los primeros 512 bytes) o completa. En ambos casos se procede a cifrar el archivo, o la parte que corresponda, mediante el algoritmo AES con la clave generada anteriormente o la embebida en el código dañino.

Tras el cifrado de cada archivo, el código dañino lo mueve a la misma ubicación pero cambiando el nombre a Base64 y una nueva extensión según se haya cifrado parcial o completamente. Las extensiones aparecen indicadas en el apartado 3.2 EXTENSIÓN AÑADIDA A LOS ARCHIVOS CIFRADOS del presente informe.

Una vez terminado el proceso de cifrado, si ha utilizado una clave AES aleatoria, se procederá a cifrar dicha clave con una clave pública RSA que lleva embebida el código dañino y envía el resultado al servidor C2, mediante el método POST. Es importante darse cuenta del uso de la misma clave AES para todos los archivos del sistema y que siempre se usa la misma clave pública RSA. Debido a esto, cualquier víctima que obtenga un descifrador válido podría descifrar a todos los afectados por el mismo código.

Posteriormente, se procede a cambiar el fondo de pantalla del escritorio por una imagen que el código dañino se descarga. Si no se puede descargar dicha imagen, se deja un fondo de pantalla vacío.

Tras ello, crea el mensaje de texto con la información del secuestro y los pasos a seguir para poder pagar, ponerse en contacto con los autores del código dañino y obtener un descifrador.

Por último, muestra la interfaz de usuario con la información acerca del secuestro de los archivos, la dirección *Bitcoin* donde enviar el pago, una dirección de correo de contacto y el identificador personal de la víctima.

8. CIFRADO Y OFUSCACIÓN

El código dañino utiliza los algoritmos AES y RSA para el proceso de cifrado de los archivos.

```
public string AESKeyEncryptWithRSA(string pwd)
{
  RSACryptoServiceProvider cryptoServiceProvider = new RSACryptoServiceProvider(2048);
  cryptoServiceProvider.FromXmlString("<RSAKeyValue><Modulus>laQ/Ndkbwswz2zvMG4MaRAt7/t/1krRdDHURawFCEUiqMbkmwY9MKxmq1T
  byte[] bytes = Encoding.UTF8.GetBytes(pwd);
  return Convert.ToBase64String(cryptoServiceProvider.Encrypt(bytes, true));
}
```

Ilustración 4. Cifrado de la clave AES con RSA

El algoritmo AES es utilizado para cifrar los archivos mientras que el algoritmo RSA se usa para cifrar la clave anteriormente usada con el algoritmo AES. Esto se realiza porque el algoritmo AES es simétrico, es decir, la misma clave que se usó para cifrar sirve para descifrar, mientras que el algoritmo RSA es asimétrico ya que usa una clave pública para cifrar y otra privada para descifrar.

```
using (MemoryStream memoryStream = new MemoryStream())
{
  using (RijndaelManaged rijndaelManaged = new RijndaelManaged())
  {
    rijndaelManaged.KeySize = 256;
    rijndaelManaged.BlockSize = 128;
    Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(passwordBytes, salt, 1000);
    rijndaelManaged.Key = rfc2898DeriveBytes.GetBytes(rijndaelManaged.KeySize / 8);
    rijndaelManaged.IV = rfc2898DeriveBytes.GetBytes(rijndaelManaged.BlockSize / 8);
    rijndaelManaged.Mode = CipherMode.CBC;
    if (!isPadding)
      rijndaelManaged.Padding = PaddingMode.None;
    using (CryptoStream cryptoStream = new CryptoStream((Stream) memoryStream, rijndaelManaged.CreateEncryptor(),
    {
      cryptoStream.Write(bytesToBeEncrypted, 0, bytesToBeEncrypted.Length);
      cryptoStream.Close();
    }
    return memoryStream.ToArray();
```

Ilustración 5. Cifrado AES de un archivo

9. PERSISTENCIA EN EL SISTEMA

El código dañino no tiene ningún sistema de persistencia en el sistema.

10. CONEXIONES DE RED

El código dañino establece conexión con su servidor C2 en dos ocasiones mediante el método POST. Al principio de su ejecución, y tras recabar información sobre el sistema, la envía a la siguiente dirección:

https://158.255.5.153/create.php

La segunda conexión se produce tras cifrar la clave AES con la clave RSA y procede a enviar la clave AES cifrada a la siguiente dirección:

https://158.255.5.153/keysave.php

Toda la comunicación se hace mediante SSL tal y como se puede apreciar en la siguiente captura:

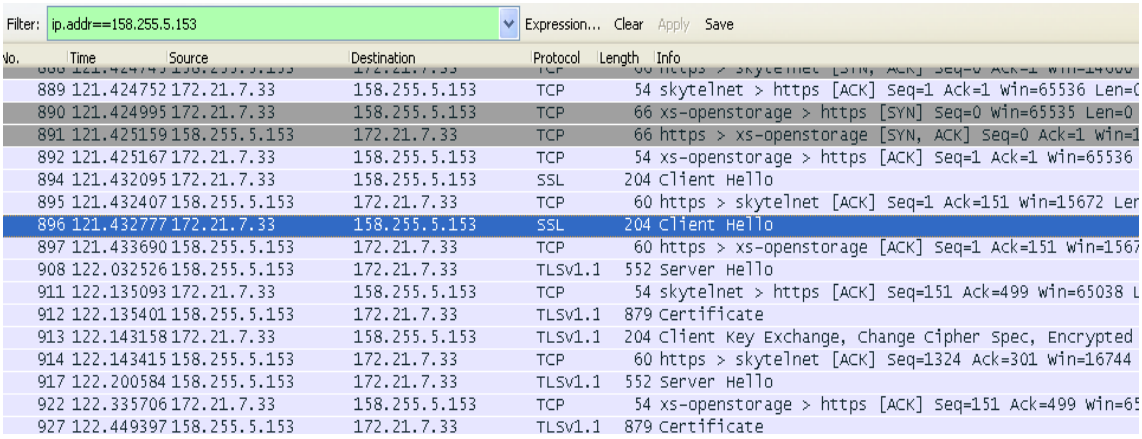

Ilustración 6. Comunicación con el servidor C2

11. ARCHIVOS RELACIONADOS

Los archivos relacionados con el código dañino son los siguientes:

<C:\Users\<usuario_activo>\>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
U2FsdGVkX1DKeR.vluni	<varía>	0 bytes	da39a3ee5e6b4b0d3255bfef95601890afd80709
bg.jpg	<varía>	<varía>	<varía>
<C:\Users\<usuario_activo>\Desktop\>			
Nombre	**Fecha Creación**	**Tamaño *bytes***	**Hash SHA1**
ReadMe.txt	<varía>	3845 bytes	<varía>

12. DETECCIÓN

Para detectar si un equipo se encuentra, o ha estado infectado, se ejecutará alguna de las herramientas de Mandiant como el "Mandiant IOC Finder" o el colector obtenido por RedLine© con los indicadores de compromiso generados para su detección. También se podrán usar Herramientas del Sistema como el Gestor de Tareas.

12.1 HERRAMIENTAS DEL SISTEMA

Cuando el código dañino esté cifrando, se puede observar con el Gestor de Tareas como la aplicación consume un alto nivel de la CPU. En caso de que el código dañino haya finalizado su proceso de cifrado, se detectará la aparición de archivos con las extensiones indicadas en el apartado de 3.2 EXTENSIÓN AÑADIDA A LOS ARCHIVOS CIFRADOS del presente informe que no podrán ser usados con ningún programa. Por ejemplo:

U2FsdGVkX1DKeR.vluni

Se podrá observar también un cambio del fondo de escritorio a una pantalla limpia o una nota de secuestro. También aparecerá una interfaz de usuario que informará acerca del proceso. Estos hechos son claros indicadores de la presencia del código dañino en el sistema.

12.2 MANDIANT

Se ha generado un nuevo archivo indicador de compromiso. El nombre del indicador generado es con GUID "b01adf7f-9917-40ab-9d33-852dbb44662e".

Se utilizará el indicador con alguna de las herramientas de las que dispone Mandiant como "Mandiant_ioc_finder" o para la confección de un recolector de evidencias mediante "Mandiant RedLine!".

Se recomienda consultar la guía de seguridad CCN-STIC-423 Indicadores de Compromiso (IOC), donde se recoge qué es un indicador de compromiso, cómo crearlo y cómo identificar equipos comprometidos.

13. DESINFECCIÓN

El código dañino no se borra del sistema por lo que el primer paso es eliminar el binario que se haya descargado. Posteriormente, se procederá a cambiar el fondo de escritorio a algo adecuado según la decisión del usuario afectado y se procederá a borrar el archivo de texto del secuestro de los archivos.

También se borrará el archivo de marca de infección:

C:\Users\<usuario>\U2FsdGVkX1DKeR.vluni

En el caso de que no se haya podido establecer conexión con el servidor C2, la clave AES embebida en el código y usada en todos los archivos es la siguiente:

BGORMkj&v=u1X0O2hOybNdRvZb9SGGnm

Con esta clave sería trivial hacer un programa que, haciendo uso de AES, descifre los archivos del sistema sabiendo las listas de cuales son cifrados enteros y cuales parcialmente o, incluso, por la detección de la extensión del archivo cifrado.

En el caso de que la clave haya sido generada aleatoriamente, **no existe forma conocida en el momento actual para recuperar los archivos cifrados por el código dañino, debiéndose recurrir a usar copias de seguridad previas de dichos archivos para obtener la información.**

El código dañino no borra los *Shadow Volumes* con lo que, si están activos, se pueden usar herramientas como ShadowExplorer para poder acceder a versiones previas de los archivos.

14. REFERENCIAS

- http://www.shadowexplorer.com/

15. INFORMACIÓN DEL ATACANTE

15.1 158.255.5.153

La información de WHOIS de la dirección IP "158.255.5.153" es la siguiente:

IP Location	Russian Federation Moscow Mir Telematiki Ltd
ASN	AS49335 NCONNECT-AS , RU (registered May 20, 2009)
Whois Server	whois.ripe.net
IP Address	158.255.5.153

```
% Abuse contact for '158.255.4.0 - 158.255.5.255' is ' abuse@hostkey.ru '

inetnum:        158.255.4.0 - 158.255.5.255
netname:        RU-Breakleft
org:            ORG-MTL21-RIPE
descr:          Breakleft Networks
country:        RU
admin-c:        ANSH13-RIPE
tech-c:         ANSH13-RIPE
status:         assigned PA
mnt-by:         MTLM-MNT
descr:          abuse-mailbox:  complaints@breakleft.net
created:        2014-03-14T12:08:08Z
last-modified:  2014-03-14T12:08:08Z
source:         RIPE

organisation:   ORG-MTL21-RIPE
org-name:       Mir Telematiki Ltd
org-type:       LIR
address:        Barabannii pereulok 4/4
address:        107023
address:        Moscow
address:        RUSSIAN FEDERATION
phone:          +7 495 369 9796
fax-no:         +7 495 369 9796
e-mail:          ulp@hostkey.ru
mnt-ref:        MTLM-MNT
mnt-ref:        RIPE-NCC-HM-MNT
mnt-by:         RIPE-NCC-HM-MNT
abuse-mailbox:   abuse@hostkey.ru
```

Ilustración 7. Información WHOIS de la IP 158.255.5.153

15.1.1 GEOLOCALIZACIÓN

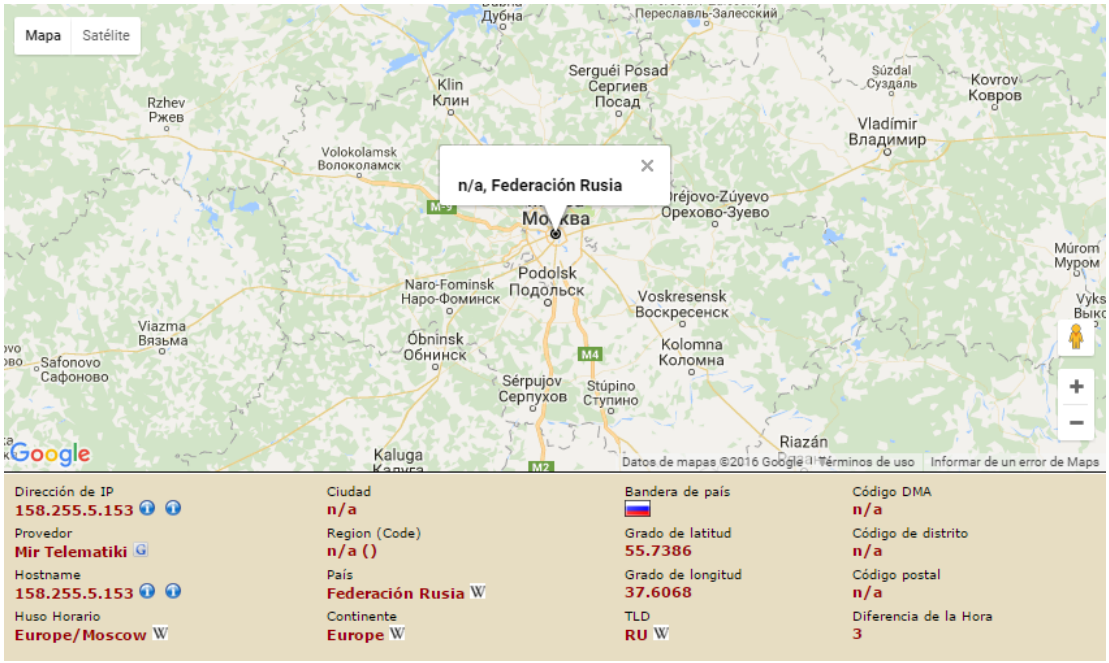

Ilustración 8. Geolocalización de la IP 158.255.5.153

16. REGLAS DE DETECCIÓN

16.1 INDICADOR DE COMPROMISO – IOC

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" id="b01adf7f-9917-40ab-9d33-
852dbb44662e" last-modified="2016-10-06T21:20:25"
xmlns="http://schemas.mandiant.com/2010/ioc">
  <short_description>Ransom_VenusLocker</short_description>
  <description>Indicador de compromiso para detectar
Ransom.VenusLocker</description>
  <authored_by>CCN-CERT</authored_by>
  <authored_date>2016-08-17T20:53:52</authored_date>
  <links />
  <definition>
    <Indicator operator="OR" id="b8658ca5-bd76-4a13-8376-399d257209d3">
      <IndicatorItem id="01efcfa8-7ca1-4193-84e4-b73ef8bafe69" condition="is">
        <Context document="RouteEntryItem" search="RouteEntryItem/Destination"
type="mir" />
        <Content type="IP">158.255.5.153</Content>
      </IndicatorItem>
      <IndicatorItem id="de6bb11a-3c6f-4da3-8be8-3f4b10a56819" condition="is">
        <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
        <Content type="md5">8675ffb697ad944748e0e24ac1a962ce</Content>
      </IndicatorItem>
```

```
      <IndicatorItem id="b3fafd81-a1e0-4c21-a9a6-3640ca9615d0" condition="is">
        <Context document="FileItem" search="FileItem/FileName" type="mir" />
        <Content type="string">U2FsdGVkX1DKeR</Content>
      </IndicatorItem>
      <Indicator operator="AND" id="34c17778-2f6b-427d-9f55-298eefaa3cce">
        <IndicatorItem id="9d06f873-2556-4605-bd42-16bee5e68b43" condition="is">
          <Context document="FileItem" search="FileItem/FileExtension" type="mir" />
          <Content type="string">vluni</Content>
        </IndicatorItem>
      </Indicator>
    </Indicator>
  </definition>
</ioc>
```

16.2 YARA

```
rule Ransom_VenusLocker
{
    /*
      Regla para detectar el codigo Ransom.VenusLocker
    */
    meta:
        description = "Regla para detectar el codigo Ransom.VenusLocker"
        author      = "CCN-CERT"
        version     = "1.0"
    strings:
        $a = { 42 00 47 00 4F 00 52 00 4D 00 6B 00 6A 00 26 00 }
        $b = { 76 00 3D 00 75 00 31 00 58 00 30 00 4F 00 32 00 }
        $c = { 68 00 4F 00 79 00 62 00 4E 00 64 00 52 00 76 00 }
        $d = { 5A 00 62 00 39 00 53 00 47 00 47 00 6E 00 6D }
        $e = { 31 00 44 00 6A 00 39 00 59 00 6E 00 4D 00 69 00 }
        $f = { 63 00 69 00 4E 00 67 00 61 00 4B 00 75 00 79 00 }
        $g = { 7A 00 4B 00 79 00 6E 00 79 00 67 00 75 00 37 00 }
        $h = { 6E 00 42 00 32 00 31 00 74 00 76 00 56 00 36 00 51 00 44 }
        $i = { 2E 00 56 00 65 00 6E 00 75 00 73 00 ?? 00 00 }
    condition:
        all of them
}
```