Informe Código Dañino
CCN-CERT ID-07/21

WastedLocker ransomware

Abril 2021

Edita:

GOBIERNO DE ESPAÑA MINISTERIO DE DEFENSA

centro criptológico nacional

Fecha de Edición: marzo de 2021

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

Quedan rigurosamente prohibidas, sin la autorización escrita del Centro Criptológico Nacional, bajo las sanciones establecidas en las leyes, la reproducción parcial o total de este documento por cualquier medio o procedimiento, comprendidos la reprografía y el tratamiento informático, y la distribución de ejemplares del mismo mediante alquiler o préstamo públicos.

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. INFORMACIÓN DEL CÓDIGO DAÑINO

El presente documento recoge un análisis sobre el componente con firma:

Hash SHA-1
4FED7EAE00BFA21938E49F33B7C6794FD7D0750C

3. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

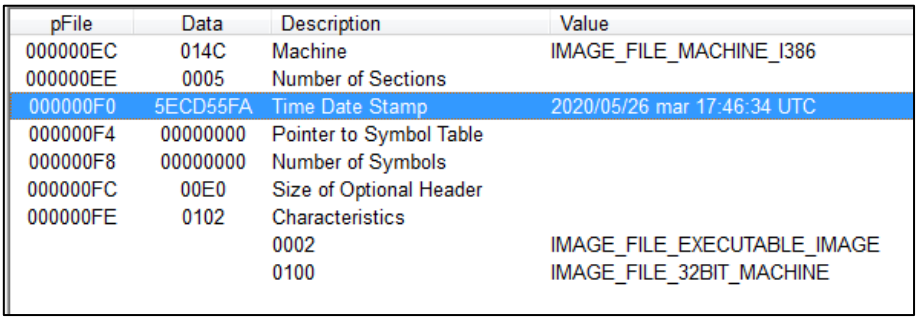
El código dañino examinado posee las siguientes características:

- Es compatible con sistemas Windows de 32 y 64 bits.
- Utiliza cifrado para dificultar su detección.
- Resuelve APIs de forma dinámica.
- Puede registrarse como servicio del sistema.
- Utiliza técnicas de escalación de privilegios (UAC Bypass).
- Realiza modificaciones en el registro de Windows.
- Cifra los ficheros de las unidades del sistema, utilizando algoritmos de cifrado simétrico (AES) y asimétrico (RSA).
- Crea un mensaje de rescate por cada fichero cifrado.
- No requiere de conexión a internet para funcionar.
- Se autodestruye al finalizar.

4. DETALLES GENERALES

La muestra analizada utiliza el formato PE EXE (Portable Executable), es decir, se corresponde con un ejecutable para sistemas operativos Windows, concretamente para 32 bits (por lo que puede funcionar también en sistemas de 64 bits), y compilado con "Microsoft Visual C/C++ 2015".

La fecha interna de creación del programa fue el 26 de mayo de 2020 a las 17:46:34 (UTC) horas, según se observa en la siguiente imagen. No obstante, hay que tener en cuenta que esta información puede ser fácilmente alterada.



pFile	Data	Description	Value
000000EC	014C	Machine	IMAGE_FILE_MACHINE_I386
000000EE	0005	Number of Sections	
000000F0	5ECD55FA	Time Date Stamp	2020/05/26 mar 17:46:34 UTC
000000F4	00000000	Pointer to Symbol Table	
000000F8	00000000	Number of Symbols	
000000FC	00E0	Size of Optional Header	
000000FE	0102	Characteristics	
		0002	IMAGE_FILE_EXECUTABLE_IMAGE
		0100	IMAGE_FILE_32BIT_MACHINE

Figura 1. Información del código dañino.

5. CARACTERÍSTICAS TÉCNICAS

5.1 CONFIGURACIÓN

El código dañino comienza descifrando el contenido de la sección ".bbs", donde mantiene su configuración, cadenas y distintos valores necesarios para su correcta ejecución.

```
nt_headers = (IMAGE_NT_HEADERS *)((char *)base + base->e_lfanew);
n_sections = nt_headers->FileHeader.NumberOfSections;
for ( sect = (IMAGE_SECTION_HEADER *)((char *)&nt_headers->OptionalHeader + nt_headers->FileHeader.SizeOfOptionalHeader);
      *(_DWORD *)sect->Name != 'ssb.';
      ++sect )
{
  if ( !--n_sections )
    return 11;
}
if ( !n_sections )
  return 11;
size = sect->SizeOfRawData;
key = size ^ nt_headers->FileHeader.TimeDateStamp ^ sect->PointerToRawData;
dst = (int *)HeapAlloc(hHeap, 0, size);
v7 = dst;
if ( !dst )
  return 8;
decrypt_section(dst, (int *)((char *)base + sect->VirtualAddress), size, key);
g_bss_rva = (char *)v7 - sect->VirtualAddress - (char *)base;
```

Figura 2. Descifrado sección ".bss".

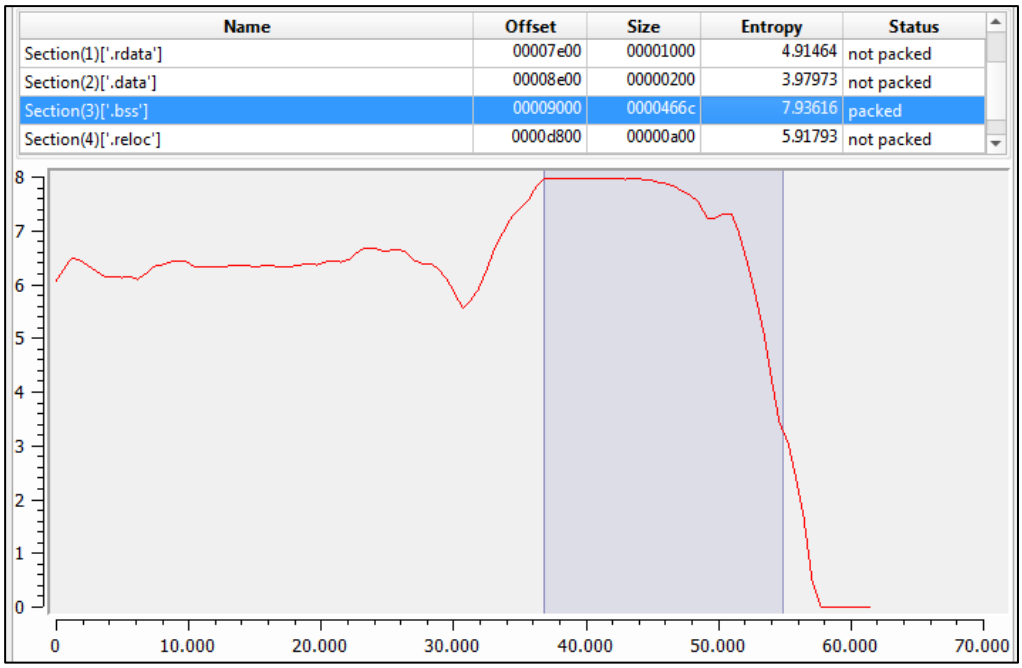


Figura 3. Entropía sección ".bss".

```
0037AB90 text "UTF-16LE", '%appdata%\',0
0037ABA6 aKernel32 db 'kernel32',0
0037ABAF aCreatethread db 'CreateThread',0
0037ABBC aLow:
0037ABBC text "UTF-16LE", 'Low\',0
0037ABC6 aWow64enablewow db 'Wow64EnableWow64FsRedirection',0
0037ABE4 aWindowsVista:
0037ABE4 text "UTF-16LE", 'Windows Vista',0
0037AC00 aExeDll:
0037AC00 text "UTF-16LE", '*.exe|*.dll',0
0037AC18 aBin:
0037AC18 text "UTF-16LE", ':bin',0
0037AC22 aMapviewoffile db 'MapViewOfFile',0
0037AC30 text "UTF-16LE", '\\?\',0
0037AC3A aReadprocessmem db 'ReadProcessMemory',0
0037AC4C aS:
0037AC4C text "UTF-16LE", '%S',0
0037AC52 aVirtualalloc db 'VirtualAlloc',0
0037AC5F aGetlasterror db 'GetLastError',0
0037AC6C aFloppy:
0037AC6C text "UTF-16LE", 'Floppy',0
```

Figura 4. Cadenas sección ".bss" descifradas.

En el bloque de configuración, el código dañino define el tipo de unidades a cifrar. En este caso, unidades de disco duro o flash (fixed), tarjetas de memoria (movable), unidades de red (remote) y carpetas compartidas (share).

```
002DD714 aMovableFixedRe:
002DD714 text "UTF-16LE", 'movable|fixed|remote|share'
002DD748 db  0Eh
```

Figura 5. Tipos de unidades a cifrar.

Asimismo, mantiene una lista blanca de extensiones, ficheros y directorios que deben ignorarse durante el cifrado de ficheros:

```
002F3438 ignored_files:                          ; DATA XREF: Stack[00000B60]:reta
002F3438 text "UTF-16LE", '*.bbawasted_info|*.bbawasted|*\NTLDR|*\BOOTMGR|*\GR'
002F3438 text "UTF-16LE", 'LDR|*.386|*.ps1|*.msu|*.ani|*.wpx|*.hlp|*.ocx|*.com'
002F3438 text "UTF-16LE", '|*.cpl|*.adv|*.cmd|*.lnk|*.drv|*.sys|*.icl|*.nls|*.'
002F3438 text "UTF-16LE", 'cab|*.bat|*.theme|*.bin|*.key|*.themepack|*.msi|*.i'
002F3438 text "UTF-16LE", 'cns|*.ics|*.idx|*.hta|*.scr|*.msstyles|*.diagcfg|*.'
002F3438 text "UTF-16LE", 'diagcab|*.nomedia|*.msc|*.cur|*.mod|*.shs|*.rtp|*.r'
002F3438 text "UTF-16LE", 'om|*.msp|*.ini|*.bak|*.dat|*.sdi|*.wim|*.dll|*.exe|'
002F3438 text "UTF-16LE", 'C:\ProgramData\*|C:\Windows\*|C:\Users\labs\AppData'
002F3438 text "UTF-16LE", '\Local\Temp*|C:\Users\labs\AppData\Roaming\*|C:\Rec'
002F3438 text "UTF-16LE", 'overy\*|C:\Program Files\*|C:\Program Files (x86)\*'
002F3438 text "UTF-16LE", '|*\bin\*|*\Boot\*|*\boot\*|*\dev\*|*\etc\*|*\lib\*|'
002F3438 text "UTF-16LE", '*\initdr\*|*\sbin\*|*\sys\*|*\vmlinuz\*|*\run\*|*\v'
002F3438 text "UTF-16LE", 'ar\*|*\Boot\*|*\System Volume Information\*|*\$RECY'
002F3438 text "UTF-16LE", 'CLE.BIN\*|*\WebCache\*|*\Caches\*|*\WindowsApps\*|*'
002F3438 text "UTF-16LE", '\AppData\*|*\ProgramData\*|*\Users\All Users\*',0,'▨'
```

Figura 6. Lista blanca.

*.bbawasted_info	*.themepack	C:\ProgramData*
*.bbawasted	*.msi	C:\Windows*
*\NTLDR	*.icns	C:\Users\labs\AppData\Local\Temp*
*\BOOTMGR	*.ics	C:\Users\labs\AppData\Roaming*
*\GRLDR	*.idx	C:\Recovery*
*.386	*.hta	C:\ProgramFiles*
*.ps1	*.scr	C:\ProgramFiles(x86)*
*.msu	*.msstyles	*\bin*
*.ani	*.diagcfg	*\Boot*
*.wpx	*.diagcab	*\boot*

*.hlp	*.nomedia	*\dev*
*.ocx	*.msc	*\etc*
*.com	*.cur	*\lib*
*.cpl	*.mod	*\initdr*
*.adv	*.shs	*\sbin*
*.cmd	*.rtp	*\sys*
*.lnk	*.rom	*\vmlinuz*
*.drv	*.msp	*\run*
*.sys	*.ini	*\var*
*.icl	*.bak	*\Boot*
*.nls	*.dat	*\SystemVolumeInformation*
*.cab	*.sdi	*\$RECYCLE.BIN*
*.bat	*.wim	*\WebCache*
*.theme	*.dll	*\Caches*
*.bin	*.exe	*\WindowsApps*
*.key		*\AppData*
		\ProgramData
		\Users\AllUsers

Por otro lado, contiene un bloque de 1028 bytes, que se corresponde con la clave pública RSA (4096 bits), que emplea durante el cifrado de los ficheros (Ver en ANEXOS).

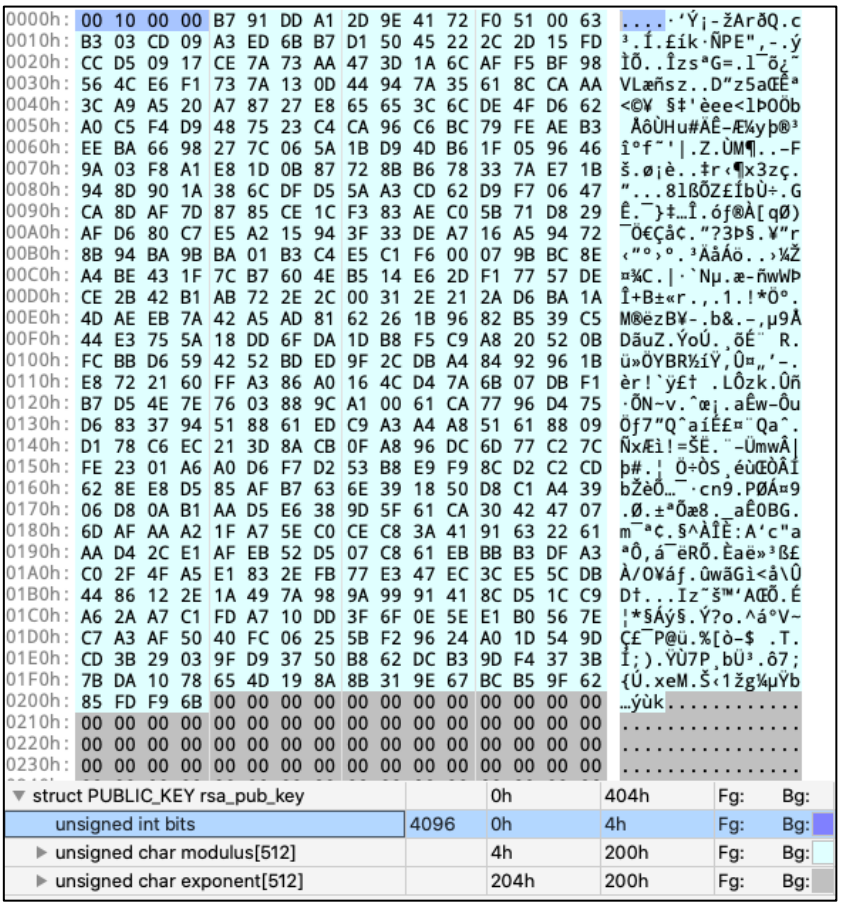


Figura 7. Clave pública RSA.

Entre los valores de la configuración, también se encuentra el nombre de la compañía objetivo, en este caso "BBA Aviation", a la cual se dirige el mensaje de rescate y en la que están basados los nombres de las extensiones que aplica sobre los ficheros cifrados.

Valor	Descripción
BBA Aviation	Nombre compañía
. bbawasted	Ext. fichero cifrado
. bbawasted_info	Ext. mensaje de rescate.

5.2 PARÁMETROS DE EJECUCIÓN

El código dañino procesa los parámetros pasados por la línea de comandos para modificar su forma de operar.

A continuación, se detallan los distintos parámetros soportados:

Parámetro	Descripción
-p <directorio>	Establece prioridad en el cifrado, cifrando primero el directorio especificado y luego el resto.
-f <directorio>	Cifra únicamente el directorio especificado.
-u user:password \\servidor	Cifra los ficheros ubicados en el recurso de red especificado, usando el usuario y contraseña proporcionado para su autenticación.
-r	Se instala como servicio, lo inicia y espera a que termine. Finalmente, elimina el servicio.
-s	Inicia el servicio creado, encargado de cifrar los ficheros del disco.

5.3 ESCALADA DE PRIVILEGIOS (UAC)

Si el código dañino detecta que la versión del sistema operativo es superior a Windows XP, y además el proceso está ejecutándose con un nivel de integridad no elevado (inferior a High Integrity), trata de escalar privilegios utilizando una técnica conocida para saltarse el UAC (User Account Control).

Para llevar a cabo la técnica, realiza los siguientes pasos:

1. Crea un directorio dentro de %appdata% con nombre aleatorio.
2. Elige un fichero EXE o DLL aleatorio dentro de %WINDIR%\system32 y lo copia en el directorio creado anteriormente, además de establecerle el atributo de oculto.
3. Escribe el contenido del código dañino en un ADS (Alternate Data Stream) con nombre ":bin" en el fichero EXE o DLL copiado en el paso anterior.

4. Crea un nuevo directorio temporal aleatorio (%temp%\<aleatorio>) y establece su punto de montaje apuntando a "C:\Windows " (incluyendo el espacio final), usando el API NtFsControlFile y la opción IO_REPARSE_TAG_MOUNT_POINT.
5. Crea un directorio "system32" dentro de la carpeta temporal (%temp%\<aleatorio>\system32) y copia en ella el ejecutable del sistema autoelevado "winsat.exe", junto con la DLL "winmm.dll" (importada por winsat.exe).
6. Parchea el punto de entrada de "winmm.dll" con un fragmento de código (shellcode) encargado de ejecutar el contenido guardado en el ADS ":bin" del paso 3.
7. Ejecuta winsat.exe con el API ShellExecuteW, que se creará alto nivel de integridad, y que, a su vez, cargará la versión parcheada de "winmm.dll", mediante una vulnerabilidad "DLL Hijacking".

Como consecuencia de este proceso, el código dañino se relanzará desde el ADS con privilegios elevados y evitando que aparezca la ventana de dialogo del UAC.

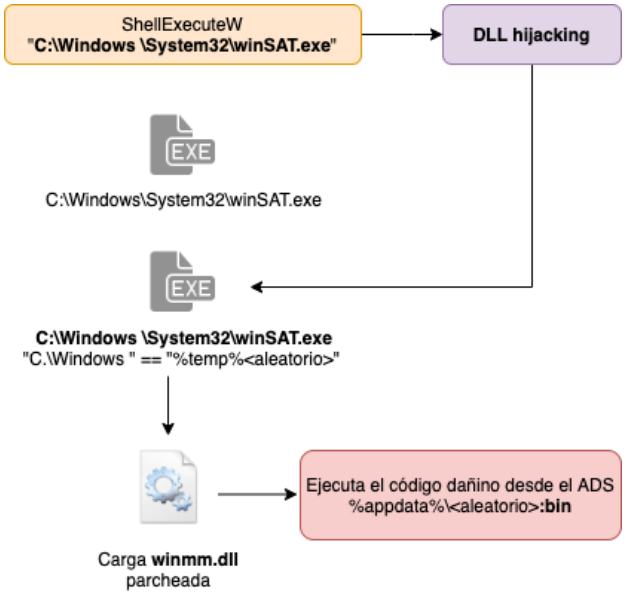


Figura 8. Escalación de privilegios vía DLL Hijacking.

5.4 ESQUEMA DE CIFRADO

Procesa las distintas unidades del sistema que coincidan con los tipos especificados en la configuración, enumerando sus ficheros y directorios, e ignorando aquellos que cumplan algún patrón de la lista blanca. Por otro lado, crea un hilo encargado de recorrer los ficheros encolados y cifrarlos.

A continuación, se detalla el proceso de cifrado aplicado en cada fichero:

- Crea el fichero "<nombre>.bbawasted_info" para la información de rescate, rellenándolo inicialmente con bytes aleatorios.
- Renombra el fichero a cifrar, añadiendo la extensión ".bbawasted".

- Mapea el fichero en memoria con permisos de lectura y escritura, usando los APIs CreateFileW, CreateFileMappingW y MapViewOfFile.
- Calcula el MD5 del contenido original del fichero.
- Genera una clave (32 bytes) y vector de inicialización (16 bytes) aleatorios, usando el API CryptGenRandom.
- Cifra el contenido del fichero con AES 256 en modo CBC.
- Des-mapea el fichero con el API UnmapViewOfFile, provocando que los cambios se apliquen en disco.
- Cifra la clave, vector de inicialización y el MD5 del contenido original del fichero, usando la clave pública RSA (4096 bits) que lleva embebida en la configuración, y codifica el resultado con base64.
- Finalmente, escribe el mensaje de rescate en el fichero "<fichero>.bbawasted_info", incluyendo el base64 de la operación anterior.

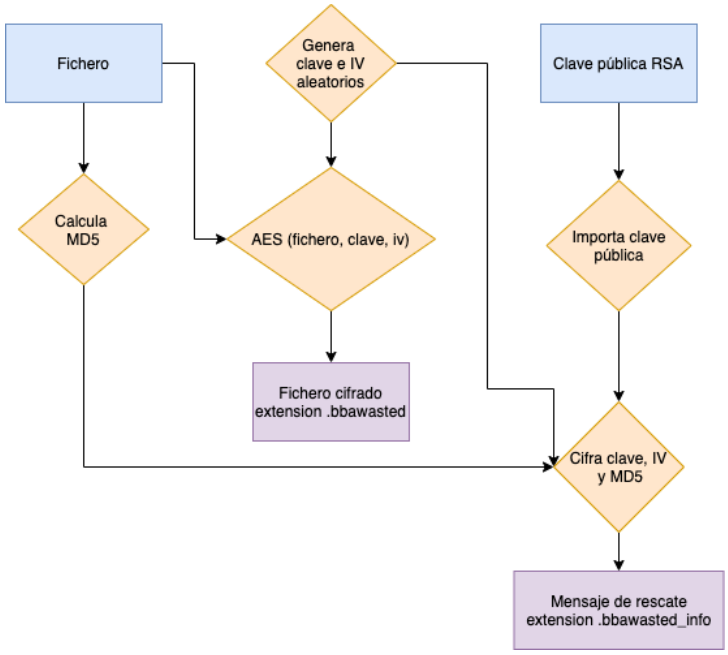
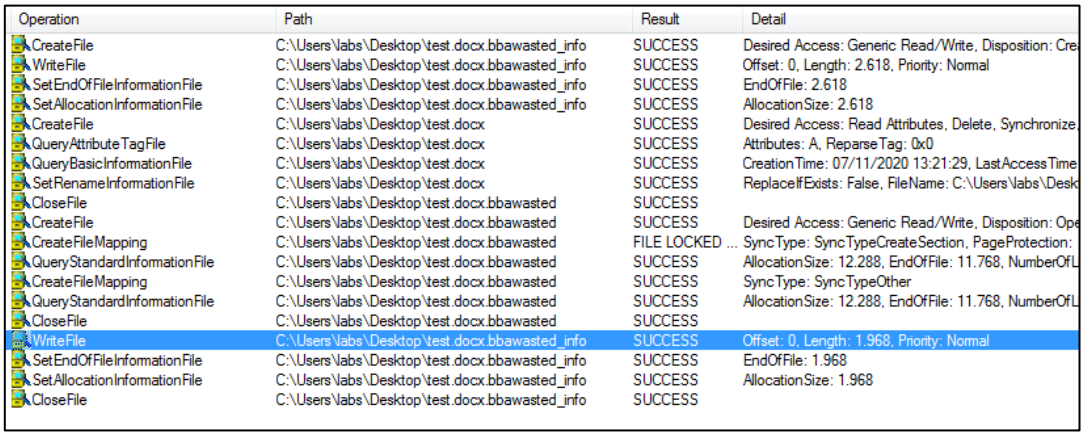


Figura 9. Flujo de cifrado.

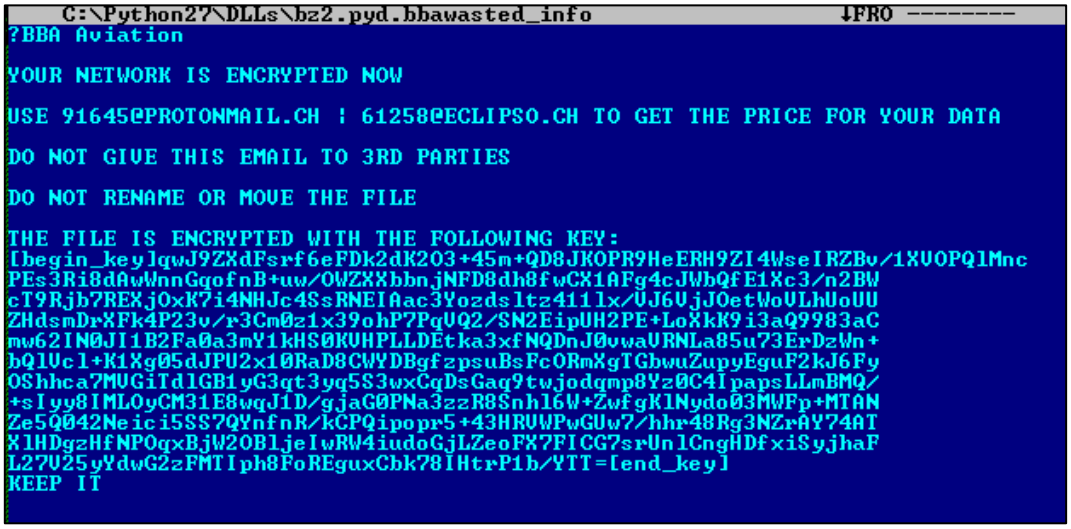
```
dwNumberOfBytesToMap = 0x4000000;
v3 = (WCHAR *)CreateFileW(lpFileName, 0xC0000000, 0, 0, 3u, 0x80u, 0);
lpFileNamea = v3;
if ( v3 == (WCHAR *)INVALID_HANDLE_VALUE )
  return GetLastError();
hFileMappingObject = CreateFileMappingW(v3, 0, PAGE_READWRITE, 0, 0, 0);
if ( hFileMappingObject )
{
  memset(a1, 0, 0x28u);
  file_size = GetFileSize(lpFileNamea, a1 + 7);
  v5 = a1[7] == 0;
  a1[6] = file_size;
  if ( v5 && file_size < 0x4000000 )
    dwNumberOfBytesToMap = file_size;
  v6 = MapViewOfFile(hFileMappingObject, 6u, 0, 0, dwNumberOfBytesToMap);// FILE_MAP_READ | FILE_MAP_WRITE
  if ( v6 )
  {
    a1[8] = dwNumberOfBytesToMap;
    a1[1] = (DWORD)hFileMappingObject;
    a1[2] = (DWORD)v6;
    a1[9] = 6;
  }
```

Figura 10. Mapeo de fichero en memoria.



Operation	Path	Result	Detail
CreateFile	C:\Users\labs\Desktop\test.docx.bbawasted_info	SUCCESS	Desired Access: Generic Read/Write, Disposition: Cre
WriteFile	C:\Users\labs\Desktop\test.docx.bbawasted_info	SUCCESS	Offset: 0, Length: 2.618, Priority: Normal
SetEndOfFileInformationFile	C:\Users\labs\Desktop\test.docx.bbawasted_info	SUCCESS	EndOfFile: 2.618
SetAllocationInformationFile	C:\Users\labs\Desktop\test.docx.bbawasted_info	SUCCESS	AllocationSize: 2.618
CreateFile	C:\Users\labs\Desktop\test.docx	SUCCESS	Desired Access: Read Attributes, Delete, Synchronize,
QueryAttributeTagFile	C:\Users\labs\Desktop\test.docx	SUCCESS	Attributes: A, ReparseTag: 0x0
QueryBasicInformationFile	C:\Users\labs\Desktop\test.docx	SUCCESS	CreationTime: 07/11/2020 13:21:29, LastAccessTime
SetRenameInformationFile	C:\Users\labs\Desktop\test.docx	SUCCESS	ReplaceIfExists: False, FileName: C:\Users\labs\Desk
CloseFile	C:\Users\labs\Desktop\test.docx.bbawasted	SUCCESS	
CreateFile	C:\Users\labs\Desktop\test.docx.bbawasted	SUCCESS	Desired Access: Generic Read/Write, Disposition: Ope
CreateFileMapping	C:\Users\labs\Desktop\test.docx.bbawasted	FILE LOCKED ...	SyncType: SyncTypeCreateSection, PageProtection:
QueryStandardInformationFile	C:\Users\labs\Desktop\test.docx.bbawasted	SUCCESS	AllocationSize: 12.288, EndOfFile: 11.768, NumberOfL
CreateFileMapping	C:\Users\labs\Desktop\test.docx.bbawasted	SUCCESS	SyncType: SyncTypeOther
QueryStandardInformationFile	C:\Users\labs\Desktop\test.docx.bbawasted	SUCCESS	AllocationSize: 12.288, EndOfFile: 11.768, NumberOfL
CloseFile	C:\Users\labs\Desktop\test.docx.bbawasted	SUCCESS	
WriteFile	C:\Users\labs\Desktop\test.docx.bbawasted_info	SUCCESS	Offset: 0, Length: 1.968, Priority: Normal
SetEndOfFileInformationFile	C:\Users\labs\Desktop\test.docx.bbawasted_info	SUCCESS	EndOfFile: 1.968
SetAllocationInformationFile	C:\Users\labs\Desktop\test.docx.bbawasted_info	SUCCESS	AllocationSize: 1.968
CloseFile	C:\Users\labs\Desktop\test.docx.bbawasted_info	SUCCESS	

Figura 11. Cifrado de fichero.



```
C:\Python27\DLLs\bz2.pyd.bbawasted_info                    ↓FRO --------
?BBA Aviation

YOUR NETWORK IS ENCRYPTED NOW

USE 91645@PROTONMAIL.CH : 61258@ECLIPSO.CH TO GET THE PRICE FOR YOUR DATA

DO NOT GIVE THIS EMAIL TO 3RD PARTIES

DO NOT RENAME OR MOVE THE FILE

THE FILE IS ENCRYPTED WITH THE FOLLOWING KEY:
[begin_keylqwJ9ZXdFsrf6eFDk2dK2O3+45m+QD8JKOPR9HeERH9ZI4WseIRZBv/1XUOPQlMnc
PEs3Ri8dAwWnnGqofnB+uw/OWZXXbbnjNFD8dh8fwCX1AFg4cJWbQfE1Xc3/n2BW
cT9Rjb7REXjOxK7i4NHJc4SsRNEIAac3Yozdsltz411x/UJ6UjJOetWoULhUoUU
ZHdsmDrXFk4P23v/r3Cm0z1x39ohP7PqUQ2/SN2EipUH2PE+LoXkK9i3aQ9983aC
mw62IN0JI1B2Fa0a3mY1kHS0KUHPLLDEtka3xfNQDnJ0vwaVRNLa85u73ErDzWn+
bQlUc1+K1Xg05dJPU2x10RaD8CWYDBgfzpsuBsFcORmXgTGbwuZupyEguF2kJ6Fy
OShhca7MUGiTd1GB1yG3qt3yq5S3wxCqDsGaq9twjodqmp8Yz0C4IpapsLLmBMQ/
+sIyy8IMLOyCM31E8wqJ1D/gjaG0PNa3zzR8Snh16W+ZwfgK1Nydo03MWFp+MTAN
Ze5Q042Neici5SS7QYnfnR/kCPQipopr5+43HRVWPwGUw7/hhr48Rg3NZrAY74AT
X1HDgzHfNPOqxBjW2OB1jeIwRW4iudoGjLZeoFX7FICG7srUn1CngHDfxiSyjhaF
L27U25yYdwG2zFMTIph8FoREguxCbk78IHtrP1b/YTT=[end_key]
KEEP IT
```

Figura 12. Mensaje de rescate con clave e IV cifrados.

Este esquema de cifrado, muy común en familias de ransomware, garantiza a los atacantes que los ficheros secuestrados solamente puedan ser descifrados usando la clave privada RSA que tienen en su posesión.

5.5 INSTALACIÓN COMO SERVICIO

En el caso de que el código dañino se ejecute con el parámetro "-r", procederá a instalarse como servicio del sistema para efectuar el proceso de cifrado.

A continuación, se detalla el flujo de instalación:

- Genera una cadena aleatoria a partir de las subclaves de registro de HKLM\SYSTEM\CurrentControlSet\Control, que utilizará como nombre del servicio y la copia del EXE.
- Se autocopia en %WINDIR%\system32\<aleatorio>.exe.
- Borra las shadow copies del sistema, ejecutando el comando:

```
vssadmin.exe Delete Shadows /All /Quiet
```

- Obtiene el control del fichero, ejecutando los comandos:

```
takeown.exe /F C:\Windows\system32\<aleatorio>.exe
icacls.exe C:\Windows\system32\<aleatorio>.exe /reset
```

- Registra el servicio, utilizando como línea de comandos "%WINDIR%\system32\<aleatorio>.exe -s" y usando como nombre de servicio: "<aleatorio>".
- Inicia el servicio.
- Espera a que termine de cifrar todos los ficheros.
- Elimina el servicio

```
hScMngr = OpenSCManagerW(0, 0, SC_MANAGER_CREATE_SERVICE);
hSCObject = hScMngr;
if ( !hScMngr )
  return GetLastError();
hSvc = CreateServiceW(
         hScMngr,
         lpServiceName,
         lpServiceName,
         SERVICE_ALL_ACCESS,
         SERVICE_WIN32_OWN_PROCESS,
         SERVICE_DEMAND_START,
         0,
         lpBinaryPathName,
         0,
         0,          lpBinaryPathName: LPCWSTR lpBinaryPathName; // [esp+3Ch] [ebp+Ch] ISARG BYREF
         0,          0x421F78:L"C:\\Windows\\system32\\Usbstor.exe -s"
         0,
         0);
lpServiceNamea = (WCHAR *)hSvc;
if ( hSvc )
{
  if ( StartServiceW(hSvc, 0, 0) )
```

Figura 13. Instalación del servicio.

5.6 CLAVES DE REGISTRO

El código dañino realiza cambios en la configuración del proxy del sistema, modificando valores de la siguiente clave de registro:

```
HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap
```

Valor	Modifcación
ProxyBypass	Borrado
IntranetName	Borrado
UNCAsIntranet	Establece valor: 0
AutoDetect	Establece valor: 1

Operation	Path	Result	Detail
RegDeleteValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\ProxyBypass	NAME NOT FOUND	
RegDeleteValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\IntranetName	NAME NOT FOUND	
RegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\UNCAsIntranet	SUCCESS	Type: REG_DWORD, Length: 4, Data: 0
RegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Internet Settings\ZoneMap\AutoDetect	SUCCESS	Type: REG_DWORD, Length: 4, Data: 1

Figura 14. Modificación valores de registro.

5.7 APIS RESUELTAS DINÁMICAMENTE

Durante la ejecución, resuelve dinámicamente algunas APIs necesarias para su correcto funcionamiento.

kernel32	dbghelp
CreateFileMappingA CreateThread ExitThread MapViewOfFile OpenProcess ReadProcessMemory VirtualAlloc CreateProcessW GetLastError Wow64EnableWow64FsRedirection	MiniDumpWriteDump

5.8 VOLCADO DE MEMORIA

El programa registra a su inicio un controlador de excepciones, para que en el caso de que se produzca algún error durante su ejecución, realice un volcado de la memoria del proceso usando el API MiniDumpWriteDump. El fichero con el volcado de memoria se escribirá en la ruta "%temp%\<nombre-proceso>.dmp".

```
if ( (ExceptionInfo->ExceptionRecord->ExceptionCode & 0xC0000000) == 0xC0000000 )
{
  v1 = WaitForSingleObject(hHandle, 0);
  if ( v1 )
  {
    SetEvent(hHandle);
    v1 = MakeMiniDump((int)ExceptionInfo);
  }
  ExitThread(v1);
}
return 1;
```

Figura 15. Generación de volcado de memoria (mini dump).

5.9 AUTOBORRADO

Cuando el código dañino termina el proceso de cifrado, se autodestruye ejecutando el siguiente comando:

```
cmd /c choice /t 10 /d y & attrib -h "<ruta-fichero>" & del "<ruta-fichero>"
```

```
memset(&StartupInfo.lpReserved, 0, 0x40u);
StartupInfo.cb = sizeof(_STARTUPINFOW);
if ( a2 )
  sub_1007356(0);
v3 = CreateProcessW(0, lpCommandLine, 0, 0, 0, 0x8000000u, 0, 0, &StartupInfo, &ProcessInformation);
if ( a2 )
  sub_1007356(1);
if ( !v3 )
  return GetLastError();
if ( lpExitCode )
{
  WaitForSingleObject(ProcessInformation.hProcess, 0xFFFFFFFF);
  GetExitCodeProcess(ProcessInformation.hProcess, lpExitCode);
```

```
lpCommandLine: LPWSTR lpCommandLine; // [esp+64h] [ebp+8h] ISARG
0x378A58:L"cmd /c choice /t 10 /d y & attrib -h \"C:\\Users\\labs\
```

Figura 16. Ejecución de autoborrado.

6. REGLAS DE DETECCIÓN

6.1 REGLAS YARA

```
import "pe"

rule WastedLocker_Ransomware
{
        meta:
                author      = "Centro Criptológico Nacional (CCN)"
                date        = "10/03/2021"
                description = "WastedLocker Ransomware"

        strings:
                $1 = {F7D123C88B45FC8345FC0489088ACBD3CA}
                $2 = {813E2E627373}
                $3 = {69C00D661900055FF36E3C}
                $4 = {351296BA5E}

         condition:
                uint16(0) == 0x5A4D and
                pe.machine == pe.MACHINE_I386 and
                pe.number_of_sections == 5 and
                all of them
}
```

7. ANEXOS

7.1 CLAVE PÚBLICA RSA (PEM)

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAt5HdoS2eQXLwUQBjswPN
CaPta7fRUEUiLC0V/czVCRfOenOqRz0abK/1v5hWTObxc3oTDUSUejVhjMqqPKml
IKeHJ+hlZTxs3k/WYqDF9NIldSPEypbGvHn+rrPuumaYJ3wGWhvZTbYfBZZGmgP4
oegdC4dyi7Z4M3rnG5SNkBo4bN/VWqPNYtn3BkfKja99h4XOHPODrsBbcdgpr9aA
x+WiFZQ/M96nFqWUcouUupu6AbPE5cH2AAebvI6kvkMffLdgTrUU5i3xd1fezitC
satyLiwAMS4hKta6Gk2u63pCpa2BYiYbloK1OcVE43VaGN1v2h249cmoIFIL/LvW
WUJSve2fLNukhJKWG+hyIWD/o4agFkzUemsH2/G31U5+dgOInKEAYcp3ltR11oM3
lFGIYe3Jo6SoUWGICdF4xuwhPYrLD6iW3G13wnz+IwGmoNb30lO46fmM0sLNYo7o
1YWvt2NuORhQ2MGkOQbYCrGq1eY4nV9hyjBCRwdtr6qiH6dewM7IOkGRYyJhqtQs
4a/rUtUHyGHru7Pfo8AvT6Xhgy77d+NH7DzlXNtEhhIuGkl6mJqZkUGM1RzJpiqn
wf2nEN0/bw5e4bBWfsejr1BA/AYlW/KWJKAdVJ3NOykDn9k3ULhi3LOd9Dc7e9oQ
eGVNGYqLMZ5nvLWfYoX9+WsCAwEAAQ==
-----END PUBLIC KEY-----
```

7.2 MENSAJE DE RESCATE

```
BBA Aviation

YOUR NETWORK IS ENCRYPTED NOW

USE 91645@PROTONMAIL.CH | 61258@ECLIPSO.CH TO GET THE PRICE FOR YOUR DATA

DO NOT GIVE THIS EMAIL TO 3RD PARTIES

DO NOT RENAME OR MOVE THE FILE

THE FILE IS ENCRYPTED WITH THE FOLLOWING KEY:
[begin_key]BASE64[end_key]
KEEP IT
```