GRANDOREIRO

BCSC-MALWARE-GRANDOREIRO

TLP:WHITE

www.basquecybersecurity.eus

Abril 2021

TABLA DE CONTENIDO

Cláusula de exención de responsabilidad

Cláusula de prohibición de venta

SOBRE EL BCSC

El Centro Vasco de Ciberseguridad (Basque Cybersecurity Centre, BCSC) es la entidad designada por el Gobierno Vasco para elevar el nivel de madurez de la ciberseguridad en Euskadi.

Es una iniciativa transversal que se enmarca en la Agencia Vasca de Desarrollo Empresarial (SPRI), sociedad dependiente del Departamento de Desarrollo Económico, Sostenibilidad y Medio Ambiente del Gobierno Vasco. Así mismo, involucra a otros tres Departamentos del Gobierno Vasco: el de Seguridad, el de Gobernanza Pública y Autogobierno, y el de Educación, y a cuatro agentes de la Red Vasca de Ciencia, Tecnología e Innovación: Tecnalia, Vicomtech, Ikerlan y BCAM.

El BCSC es la entidad de referencia para el desarrollo de la ciberseguridad y de la confianza digital de ciudadanos, empresas e instituciones públicas en Euskadi, especialmente para los sectores estratégicos de la economía de la región.

La misión del BCSC es por tanto promover y desarrollar la ciberseguridad en la sociedad vasca, dinamizar la actividad empresarial de Euskadi y posibilitar la creación de un sector profesional que sea referente. En este contexto se impulsa la ejecución de proyectos de colaboración entre actores complementarios en los ámbitos de innovación tecnológica, investigación y transferencia tecnológica a la industria de fabricación avanzada y otros sectores.

Así mismo, ofrece diferentes servicios en su rol como Equipo de Repuesta a Incidentes (en adelante CERT, por sus siglas en inglés “Computer Emergency Response Team”) y trabaja en el ámbito de la Comunidad Autónoma del País Vasco para aumentar la capacidad de detección y alerta temprana de nuevas amenazas, la respuesta y análisis de incidentes de seguridad de la información, y el diseño de medidas preventivas para atender a las necesidades de la sociedad vasca. Con el fin de alcanzar estos objetivos forma parte de diferentes iniciativas orientadas a la gestión de incidentes de ciberseguridad:

1. RESUMEN EJECUTIVO

Grandoreiro es un malware de tipo troyano bancario cuyas primeras apariciones se detectaron en 2017 y apuntaba a Brasil y Perú. Entrado 2019, se expandió y añadió a México y España a su lista de países objetivo.

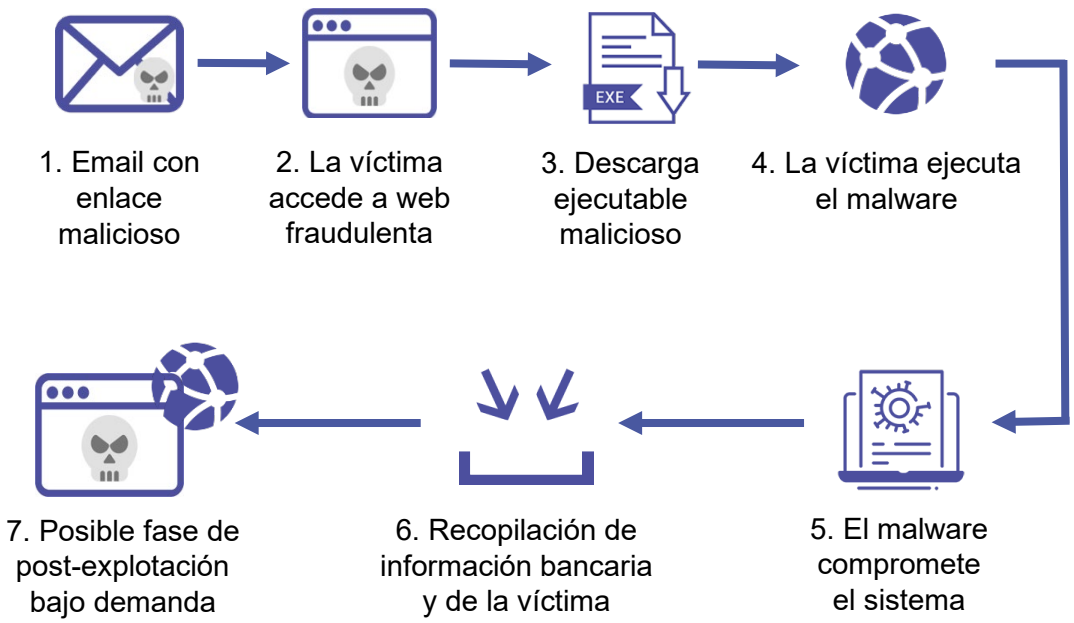
Está hecho en Delphi y se distribuye a través de campañas de correos maliciosos tratando de simular actualizaciones de Java o Flash. Recientemente, también se han aprovechado del miedo por el COVID-19.

Su principal característica, la cual lo diferencia del resto de troyanos bancarios, es la forma de realizar el *binary padding*, que consiste en aumentar el tamaño del binario para que supere el límite de tamaño que los mecanismos tienen establecido a la hora de analizar archivos maliciosos y así no sea detectado.

Su principal función es la de obtener credenciales bancarias, incluyendo direcciones de wallets de BTC. También actúa como puerta trasera, permitiendo realizar acciones sobre el equipo infectado a través de sus servidores de C&C.

2. ANÁLISIS TÉCNICO

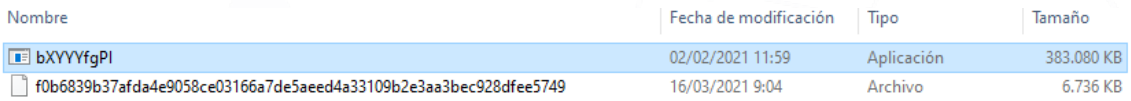
Flujo de infección



Grandoleiro se distribuye a través de campañas de spam con enlaces a sitios web fraudulentos desde donde se descargan ficheros maliciosos. Al ejecutarse, el malware comienza a recopilar información sobre la víctima y el equipo infectado, como usuarios y contraseñas de acceso a entidades financieras y otro tipo de credenciales. También actúa como puerta trasera, permitiendo realizar acciones sobre el equipo infectado a través de sus servidores de C&C.

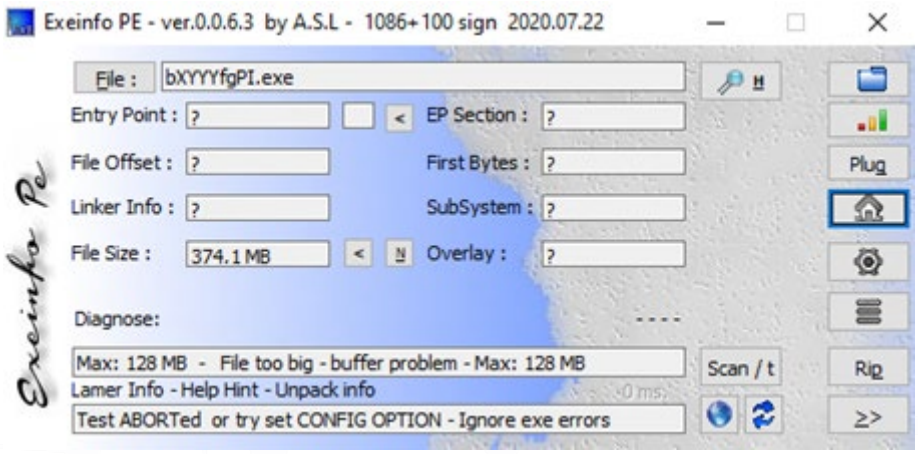
El análisis se realizará sobre una muestra cuya firma MD5 es: **951DE5701EE5F747589997BC50B5820B.**

Se trata de un fichero ZIP sin contraseña de **6.736 KB**. Al extraer el contenido del fichero, se obtiene un binario con nombre "bXYYYfgPI", el cual tiene un tamaño de **383.080 KB**, tal y como se observa en la siguiente imagen:



Nombre	Fecha de modificación	Tipo	Tamaño
bXYYYfgPI	02/02/2021 11:59	Aplicación	383.080 KB
f0b6839b37afda4e9058ce03166a7de5aeed4a33109b2e3aa3bec928dfee5749	16/03/2021 9:04	Archivo	6.736 KB

En este punto se introduce la primera técnica empleada por el *malware*, el *binary padding*. Esta técnica consiste en rellenar el binario con información irrelevante con el fin de que, por su gran tamaño, no sea analizado dificultando así su detección.



La prueba se puede observar en la imagen anterior donde *Exeinfo PE* no es capaz de analizar el binario para obtener sus datos debido a su tamaño. De igual forma ocurría con VirusTotal que hasta 2019 únicamente permitía subir archivos de como máximo 256 MB, y ocurre con distintas herramientas de detección de *malware*.

Analizando las secciones del binario y el tamaño de éstas, se observa que la sección de recursos, **.rsrc**, contiene prácticamente el 97% del binario. A continuación, se muestran algunos de los elementos contenidos en dicha sección.

type (8)	name	file-offset (185)	signature (9)	non-standard	size (376987461 ...	file-ratio (96.10%)	md5	entropy	language (6)
rcdata	ACGL	0x1760C3B8	PNG	-	3263	0.00 %	3617368B179E75D6B3E0EB3A024C30A2	7.886	English-Un...
rcdata	ACSHDA	0x1760D078	PNG	-	3416	0.00 %	35BD2B5D9D94A29C203370DF3ECE7E39	7.883	Russian
rcdata	ACSHDI	0x1760DDD0	PNG	-	3341	0.00 %	6FEF838D9DEE8333A861CDEAEDAECE2D	7.882	Russian
rcdata	SC	0x1760FEF4	PNG	-	1076	0.00 %	2CA53C272F3B5101A6792587CEB0C766	7.769	English-Un...
rcdata	SD	0x17610328	PNG	-	1201	0.00 %	3D251FE10576C1AEEF92C8D02E9B5E6B	7.751	English-Un...
rcdata	SE	0x176107DC	PNG	-	417	0.00 %	93AF80836594E23DA1F5CA17508FE370	7.119	English-Un...
rcdata	SF	0x17610980	PNG	-	1649	0.00 %	9051DC17648B2F123F497E19CBAD4D4E	7.829	English-Un...
rcdata	SR	0x17610FF4	PNG	-	1969	0.00 %	573B5A1D2F65E1A8C774139B90920B92	7.840	English-Un...
rcdata	DVCLAL	0x1760EAE0	Delphi	-	16	0.00 %	D8090ABA7197FBF9C7E2631C750965A8	4.000	neutral
rcdata	TBICLOST1	0x176117A8	Delphi	-	7774	0.00 %	6BCF16B293312DA3FAEFD25216C4552B	5.622	neutral
rcdata	TBICLOST2	0x17613608	Delphi	-	257	0.00 %	CF9286117CA5916B9FFC7BDDB2CAA542	5.464	neutral
rcdata	TBICLOST3	0x1761370C	Delphi	-	236	0.00 %	EBD240D46876CA68C0441676F2AF534D	5.522	neutral
rcdata	TBICLOST4	0x176137F8	Delphi	-	236	0.00 %	7489615FB8AB035C5F255664611192B5	5.524	neutral
rcdata	TBICLOST5	0x176138E4	Delphi	-	257	0.00 %	3A4B98BBA229F0B52CBF51CEE54FDE63	5.463	neutral
rcdata	TBICLOST6	0x176139E8	Delphi	-	337	0.00 %	BB3DE71721421A4EF6E0515853059CAE	5.404	neutral
rcdata	TBICLOST7	0x17613B3C	Delphi	-	257	0.00 %	F1B2CE7775E90196B61196D07E038214	5.443	neutral
rcdata	TBICLOST8	0x17613C40	Delphi	-	249	0.00 %	FE9D20F4FAA91EFC46FEEFE19D9377FC	5.466	neutral
rcdata	TBICLOST9	0x17613D3C	Delphi	-	249	0.00 %	C196FDBB863C46129A674BF59F924C38	5.443	neutral
rcdata	TDSSCUBEEDITOR	0x17613E38	Delphi	-	6221	0.00 %	8914D0D58968CC68020A7742AAEDFA22	5.502	neutral
rcdata	TFLIG	0x17615688	Delphi	-	328	0.00 %	937B9CBECFCF883092FE3DD160A68070	5.350	neutral
rcdata	TLOGINDIALOG	0x176157D0	Delphi	-	1172	0.00 %	0A8F8F01D73A44DB20DB5228FF69BFF5	5.417	neutral
rcdata	TPASSWORDDIAL...	0x17615C64	Delphi	-	964	0.00 %	72BFF64571F7BC5F84FE6F90017730D5	5.415	neutral
rcdata	TPROGRESSDIALOG	0x17616028	Delphi	-	572	0.00 %	836220E42275DBF75B0E79D6CE7B0087	5.240	neutral
rcdata	TSCALCFORM	0x17616264	Delphi	-	6919	0.00 %	3D157BD393BA04EE80FAF6464D21DCCE	5.507	neutral
rcdata	TSCOLORDIALOGF...	0x17617D6C	Delphi	-	7053	0.00 %	8DD193EA253623B895A37A864FE296DC	5.618	neutral
ISO	DRGGEEWWW3F	0x00E93FD4	unknown	x	127254186	32.44 %	8B1D55F83552B4A05E6BB593AA99B6FF	2.090	Portuguese
ISO	EWGGEGWGEWG7...	0x087EFE80	unknown	x	249596938	63.63 %	E895A5AD6D1B4118450C46185CEB1FA6	1.581	Portuguese

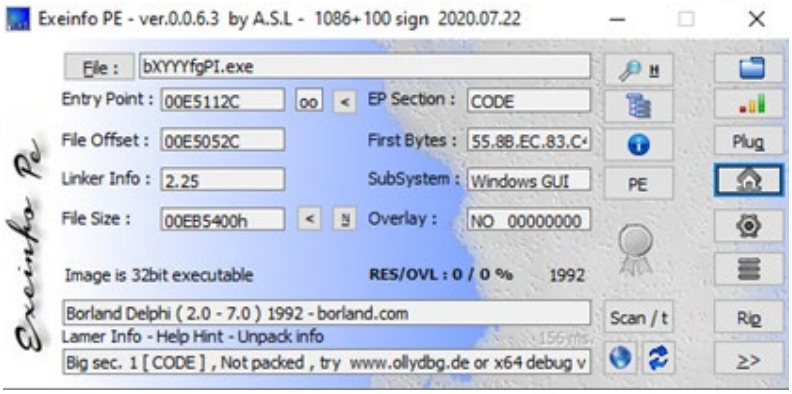
Resaltan los dos últimos elementos que se clasifican como ISO, se identifica el idioma portugués y no se reconoce ninguna firma asociada. Estos dos elementos ocupan el 32.44% y el 63.33% y poseen un nombre aleatorio.

Extrayendo ambos recursos y analizándolos, se observa que son dos imágenes BMP de gran tamaño, una de **124.272 KB** y otra de **243.748 KB**, cuyo contenido es el que se muestra en las siguientes imágenes.

Las imágenes tienen un estilo parecido pero ningún significado concreto o utilidad más allá que aumentar el tamaño del fichero.

Nombre	Fecha de modificación	Tipo	Tamaño
bXYYYfgPI	02/02/2021 11:59	Aplicación	15.061 KB
bXYYYfgPI_original	02/02/2021 11:59	Aplicación	383.080 KB
DRGGEEWWW3F	16/03/2021 9:30	Archivo BMP	124.272 KB
EWGGEGWGEWG7GGE7	16/03/2021 9:30	Archivo BMP	243.748 KB
f0b6839b37afda4e9058ce03166a7de5aeed4a33109b2e3aa3bec928dfee5749	16/03/2021 9:04	Archivo	6.736 KB

Una vez extraídas y eliminadas las imágenes, queda un binario **ejecutable** de **32bit** hecho en **Delphi** de **15.061 KB**:



Su estructura se basa en la creación de formularios a los cuales se les asignan unos *Timers* y unas funciones determinadas. Un formulario es una ventana estándar de una aplicación que puede contener objetos del tipo *Button*, *CheckBox* o *ComboBox* entre otros.

Un *Timer*, como bien indica Microsoft, es una función que se repite cada cierto tiempo y, en este caso, está asociado a un formulario. A continuación, se muestra la definición de sus funciones:

Timer Functions

Name	Description
KillTimer	Destroys the specified timer.
SetTimer	Creates a timer with the specified time-out value.
TimerProc	An application-defined callback function that processes WM_TIMER messages. The TIMERPROC type defines a pointer to this callback function. *TimerProc* is a placeholder for the application-defined function name.

En la siguiente imagen se muestra el *EntryPoint* de la muestra que es muy similar a otros troyanos bancarios hechos en Delphi.

```
EntryPoint
0125112C        push        ebp
0125112D        mov         ebp,esp
0125112F        add         esp,0FFFFFFF0
01251132        mov         eax,1250464
01251137        call        @InitExe
0125113C        mov         eax,[1260DD4];^Application:TApplication
01251141        mov         eax,dword ptr [eax]
01251143        call        TApplication.Initialize
01251148        mov         ecx,dword ptr ds:[1260754];^gvar_0126451C:TBICLOST1
0125114E        mov         eax,[1260DD4];^Application:TApplication
01251153        mov         eax,dword ptr [eax]
01251155        mov         edx,dword ptr ds:[123D4A4];TBICLOST1
0125115B        call        TApplication.CreateForm
01251160        mov         eax,[1260DD4];^Application:TApplication
01251165        mov         eax,dword ptr [eax]
01251167        mov         byte ptr [eax+5B],0;TApplication.FShowMainForm:Boolean
0125116B        mov         eax,[1260DD4];^Application:TApplication
01251170        mov         eax,dword ptr [eax]
01251172        call        TApplication.Run
01251177        call        @Halt0
```

En la llamada a **TApplication.CreateForm** se observa que se le pasa un puntero a **TBICLOST1** como argumento. **TBICLOST1** hace referencia al formulario principal del *malware*.

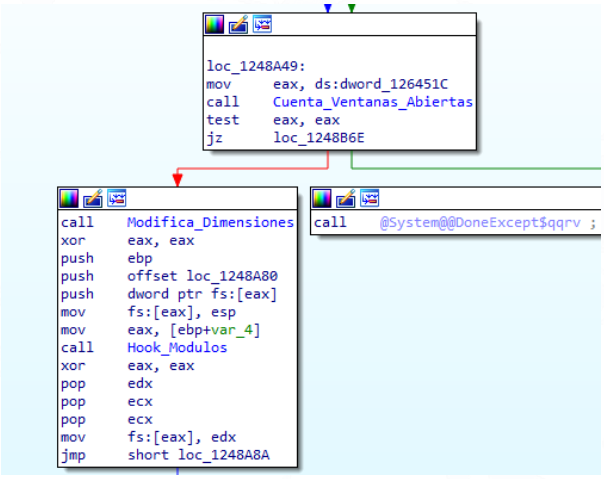
```
TCustomForm #0046CF28 Sz=2F8
  <D>
  <V>
  TForm #0046D22C Sz=2F8
    <V>
    TMessageForm #00437400 Sz=2FC
    TLoginDialog #0115BD3C Sz=324
    TPasswordDialog #0115C464 Sz=31C
    TProgressDialog #01186B90 Sz=328
    TDssCubeEditor #01192D98 Sz=3E4
    TBICLOST2 #011AF300 Sz=2FC
    TBICLOST3 #011AF5A4 Sz=2F8
    TBICLOST4 #011AF760 Sz=2F8
    TFlig #0123A48C Sz=300
    TBICLOST6 #0123ABD8 Sz=2FC
    TBICLOST7 #0123BE64 Sz=2F8
    TBICLOST8 #0123C58C Sz=2F8
    TBICLOST9 #0123CCD4 Sz=2F8
    TBICLOST1 #0123D4A4 Sz=408
      <E>
      <D>
      <V>
```

En la imagen anterior se observan todos los formularios que lo constituyen. Estos formularios son cargados según la función o funciones a realizar, estando asociados a una o varias entidades bancarias objetivo.

A continuación, se muestran las funciones asociadas al formulario principal y a los *Timers* del mismo:

```
TBICLOST1 #0123D4A4 Sz=408
  <E>
    #01248A08 TBICLOST1.FormCreate
    #01248C14 TBICLOST1.TimerReconTimer
    #01248C28 TBICLOST1.TimerkBROWTimer
    #01248C34 TBICLOST1.TimerURLMONITORTimer
    #0124D340 TBICLOST1.TimerHOOKTimer
    #0124D368 TBICLOST1.Timer4LTimer
    #0124D3CC TBICLOST1.Time5LTimer
    #0124D42C TBICLOST1.TimertskTimer
    #0124D4DC TBICLOST1.TRMTimer
    #0124D510 TBICLOST1.TimerBTCTimer
```

Al cargar el formulario, la primera función que se ejecuta es **FormCreate**:



Esta función empieza comprobando que se hayan cargado al menos **5** ventanas en el sistema utilizando **EnumWindows** para almacenar el resultado en un objeto de tipo **TListBox** y posteriormente realizar la comparación. Si el resultado de la comparación es negativo no realizará ninguna de las siguientes acciones. Esto lo realiza para asegurarse que el sistema operativo ha sido cargado completamente y para evitar ciertos análisis automáticos que apenas tienen aplicaciones en ejecución.

A continuación, se modifican las dimensiones del formulario de forma aleatoria y se realizan *hooks* sobre algunos módulos para poder realizar ciertas acciones que serán explicadas en mayor profundidad en el apartado de *Timers*, concretamente en el *HOOKTimer*. Todo ello lo hace a través de **LdrLoadDLL**:

```
loc_1248A8A:
mov     eax, [ebp+var_4]
call    Inicia_Timers
mov     eax, offset dword_12645F8
mov     edx, offset _str_dan1_02022021.Text
call    @System@@LStrAsg$qqrpvpxv ; System::__linkproc__ LStrAsg(void *,void *)
mov     eax, offset dword_126453C
mov     edx, offset _str_mrvfnjjc.Text
call    @System@@LStrAsg$qqrpvpxv ; System::__linkproc__ LStrAsg(void *,void *)
mov     eax, [ebp+var_4]
call    RTCClient_y_Persistencia
lea     edx, [ebp+var_C]
xor     eax, eax
call    StringPorIndice
mov     edx, [ebp+var_C]
lea     ecx, [ebp+var_8]
xor     eax, eax
call    Descifra_String
mov     eax, [ebp+var_8]
call    @System@@LStrToPChar$qqrx17System@AnsiString ; System::__linkproc__ LStrToPChar
push    eax
lea     eax, [ebp+System::AnsiString]
call    Windows_ProductName
mov     eax, [ebp+System::AnsiString] ; System::AnsiString
lea     edx, [ebp+var_10]
call    @Sysutils@Trim$qqrx17System@AnsiString ; Sysutils::Trim(System::AnsiString)
mov     eax, [ebp+var_10]
call    @System@@LStrToPChar$qqrx17System@AnsiString ; System::__linkproc__ LStrToPChar
pop     edx             ; Windows 8
call    @Sysutils@StrPos$qqrpxct1 ; Sysutils::StrPos(char *,char *)
test    eax, eax
jnz     short loc_1248B45
```

El siguiente paso que realiza es iniciar los *Timers* asociados al formulario. Estos *Timers* son:

- ReconTimer
- kBROWTimer
- URLMONITORTimer
- HOOKTimer
- 4LTimer
- 5LTimer
- tskTimer
- TRMTimer
- BTCTimer

Para conseguirlo, realiza las siguientes acciones a cada *Timer*: asigna una función, un intervalo de tiempo para que se vaya ejecutando dicha función y lo activa.

```
xor         ecx,ecx
mov         dl,1
mov         eax,[438B48];TTimer
call        TTimer.Create;TTimer.Create
mov         [1264608],eax;gvar_01264608:TTimer
lea         edx,[ebp-4]
mov         eax,29
call        004B3558
mov         edx,dword ptr [ebp-4]
mov         eax,[1264608];gvar_01264608:TTimer
mov         ecx,dword ptr [eax]
call        dword ptr [ecx+18]
mov         edx,7D0
mov         eax,[1264608];gvar_01264608:TTimer
call        TTimer.SetInterval
push        ebx
push        124D510;TBICLOST1.TimerBTCTimer
mov         eax,[1264608];gvar_01264608:TTimer
call        TTimer.SetOnTimer
mov         dl,1
mov         eax,[1264608];gvar_01264608:TTimer
call        TTimer.SetEnabled
```

Una vez activados los *Timers* se procede a configurar la comunicación con el C2C y establecer la persistencia.

Para implementar la comunicación con el servidor C2C, este *malware* hace uso de RealThinClient SDK. Se trata de un componente que utiliza un protocolo que opera sobre HTTP:

```
mov         dl,1
mov         eax,[123E6A8];TEventHandlersX
call        TObject.Create;TEventHandlersX.Create
mov         [126462C],eax;gvar_0126462C:TEventHandlersX
xor         ecx,ecx
mov         dl,1
mov         eax,[53539C];TRtcHttpPortalClient
call        TRtcHttpPortalClient.Create;TRtcHttpPortalClient.Create
```

Tras configurar el cliente RTC procede a realizar la persistencia:

```
lea         edx,[ebp-14]
mov         eax,[1260DD4];^Application:TApplication
mov         eax,dword ptr [eax]
call        TApplication.GetExeName
mov         edx,dword ptr [ebp-14]
mov         cl,1
mov         eax,[126453C];gvar_0126453C:AnsiString
call        Persistencia
```

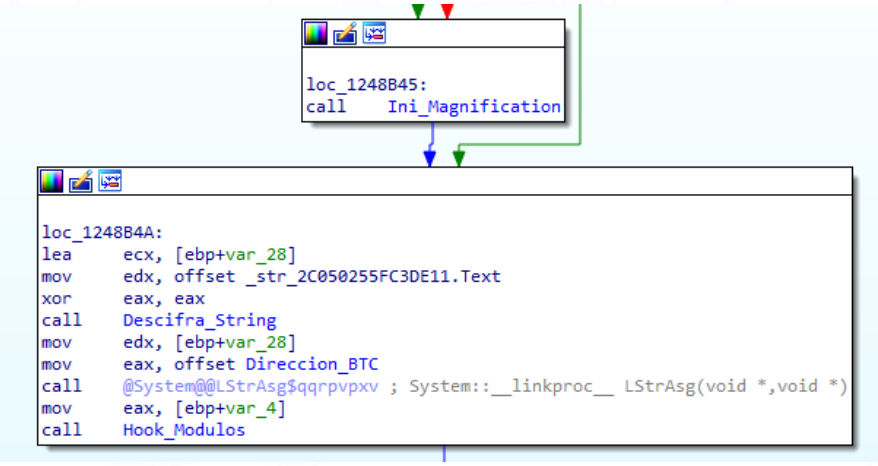
Para ello, empieza obteniendo el nombre del binario.

```
xor         ecx,ecx
mov         dl,1
mov         eax,[43F638];TRegIniFile
call        TRegIniFile.Create;TRegIniFile.Create
mov         ebx,eax
mov         edx,80000001
mov         eax,ebx
call        TRegistry.SetRootKey
mov         eax,dword ptr [ebp-8]
push        eax
push        4B30A4;'Software\Microsoft\Windows\CurrentVersion\Run'
push        dword ptr [ebp-0C]
push        4B30DC;#0
lea         eax,[ebp-14]
mov         edx,3
call        @LStrCatN
mov         edx,dword ptr [ebp-14]
mov         ecx,dword ptr [ebp-4]
mov         eax,ebx
call        TRegIniFile.WriteString
mov         eax,ebx
call        TObject.Free
```

Sigue estableciendo la ruta donde se encuentra el *malware* para que se ejecute en cada inicio del sistema en la clave:

HKCU\Software\Microsoft\Windows\CurrentVersion\Run

A continuación, comprueba si el sistema es Windows 10 o Windows 8: si el nombre de producto se corresponde con Windows 10 no cargaría los módulos correspondientes a la librería DLL “Magnification”. En cualquier otro caso sí los cargaría. Esta librería es utilizada por el *malware* para realizar capturas de pantalla.



Finalmente, guarda en memoria una dirección de un *wallet* de BTC que será utilizada en la función asociada al BTC *Timer* y asegura que los *hooks* necesarios estén establecidos correctamente.

Grandoreiro, al igual que otras familias de troyanos bancarios, utiliza un mecanismo de desofuscación de cadenas basado en dos funciones. En la primera, dado un número entre el 0 y el 590, devuelve la cadena cifrada

correspondiente. Dicha cadena es pasada a la segunda función como argumento y es devuelta descifrada.

El algoritmo para descifrar las cadenas se corresponde con el siguiente código escrito en Python:

```python
def descifrar_string(data_enc, key):
    data_dec = str()

    data_enc = 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

        prev = c
        data_dec += chr(x)

    return data_dec
```

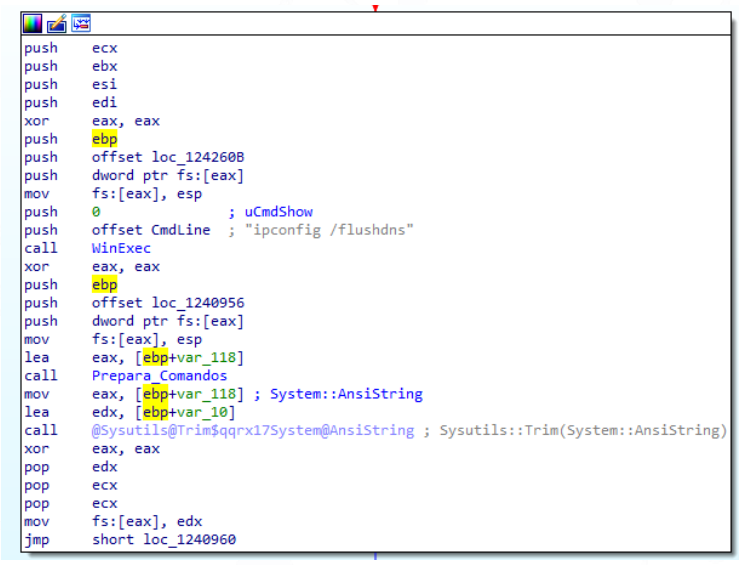
Timers

A continuación, se detallan las funciones asociadas a los principales *Timers* vistos anteriormente.

URLMONITORTimer:

En primer lugar, utiliza **InternalGetWindowText** para obtener el título de la ventana de navegación.

Posteriormente, comprueba si el título contiene alguna de las entidades bancarias objetivo. De ser así, inicia la comunicación con el C2C.



Grandoreiro utiliza dominios dinámicos, por ello el primer paso que realiza es borrar la caché DNS del equipo infectado. A continuación, descifra los comandos que utiliza para poder interpretarlos y luego generar los dominios:

```
mov         eax,dword ptr [ebp-0C]
mov         edx,1242654;'0.0.0.0'
call        @LStrCmp
jne         01240A11
lea         ecx,[ebp-12C]
mov         edx,1242664;'1'
mov         eax,dword ptr [ebp-10]
call        0124DA04
mov         ecx,dword ptr [ebp-12C]
lea         eax,[ebp-128]
mov         edx,1242670;'a'
call        @LStrCat3
mov         eax,dword ptr [ebp-128]
lea         edx,[ebp-124]
call        LowerCase
lea         eax,[ebp-124]
mov         edx,124267C;'.zapto.org'
call        @LStrCat
mov         eax,dword ptr [ebp-124]
lea         edx,[ebp-0C]
call        TCPServer
cmp         dword ptr [ebp-0C],0
je          01240A26
```

Como se puede observar en la imagen anterior, esta versión del *malware* utiliza un Algoritmo Generador de Dominios (**DGA**) compuesto de 3 tipos de cadenas.

1. La primera cadena es la que devuelve la función cuya posición de memoria es 0x0124DA04. Esta función, recibe dos números enteros, realiza una división entre ambos, convierte el resultado a hexadecimal y lo retorna.
2. El segundo son las letras de la "**a**" a la "**f**".
3. Y la tercera cadena son nombres pertenecientes a la siguiente lista:
    - .zapto.org
    - .servequake.com
    - .servehalflife.com
    - .servecounterstrike.com
    - .redirectme.net
    - .myftp.org
    - .hopto.org
    - .ddnsking.com
    - .gotdns.ch
    - .myftp.biz

Se generan un total de **60** nombres de dominio.

```
push        1242A7C;'[BR]['
lea         eax,[ebp-438]
lea         edx,[ebp-111]
mov         ecx,101
call        @LStrFromArray
mov         eax,dword ptr [ebp-438]
lea         edx,[ebp-434]
call        Trim
push        dword ptr [ebp-434]
push        1242A8C;']['
push        dword ptr ds:[1264620];
```

Con los DNS dinámicos generados, se obtiene el nombre de **usuario**, el nombre del **equipo**, del **sistema operativo** (versión y arquitectura) y los mecanismos de **seguridad** instalados y utiliza el cliente RTC configurado anteriormente para enviarlos. Utilizando "**[**" y "**]**" como delimitadores de contenido.

Una vez enviados queda a la espera de instrucciones por parte del C2C para cargar otros formularios relacionados con entidades bancarias o realizar acciones sobre el equipo.

Estas acciones pueden:

- Manipular ventanas
- Capturar pulsaciones del teclado.
- Simular acciones del ratón y del teclado.
- Abrir una URL.
- Reiniciar el equipo.

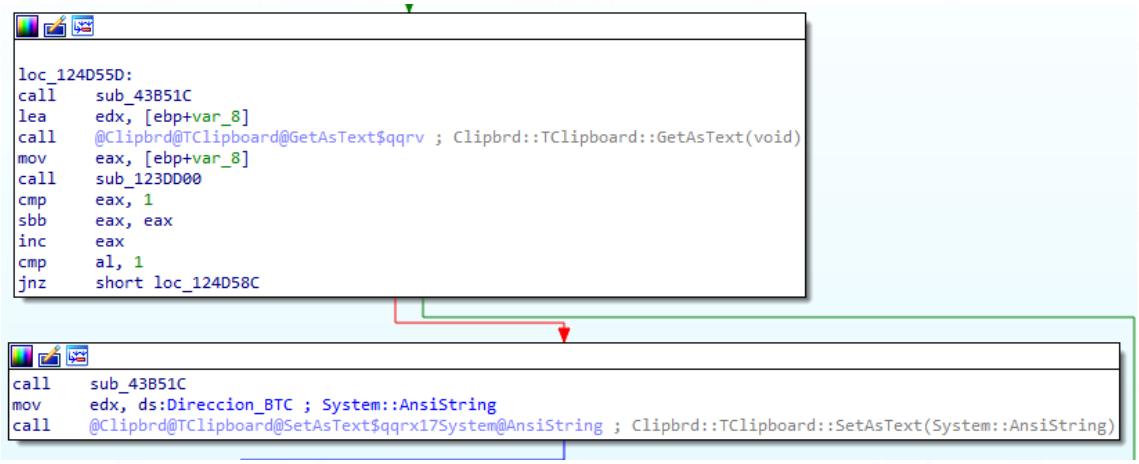
HOOKTimer

La función asociada a este *Timer* se encarga de realizar los *hooks* sobre los módulos necesarios para evadir los mecanismos de protección y poder realizar acciones sobre el equipo. Dichos mecanismos son aplicaciones de terceros (**Trusteer** y **Warsaw**) que son aplicaciones de acceso seguro a la banca en Latinoamérica. También trata de ver si el binario está siendo depurado a través de **IsDebuggerPresent.**

```
mov     esi, [ebp+dwSize]
push    ebx             ; lpProcName
mov     eax, [ebp+lpModuleName]
push    eax             ; lpModuleName
call    GetModuleHandleA_1_0
push    eax             ; hModule
call    GetProcAddress_0
mov     ebx, eax
push    edi             ; dwProcessId
push    0               ; bInheritHandle
push    1F0FFFh         ; dwDesiredAccess
call    OpenProcess
mov     edi, eax
lea     eax, [ebp+flOldProtect]
push    eax             ; lpflOldProtect
push    40h ; '@'       ; flNewProtect
push    esi             ; dwSize
push    ebx             ; lpAddress
push    edi             ; hProcess
call    VirtualProtectEx
lea     eax, [ebp+NumberOfBytesWritten]
push    eax             ; lpNumberOfBytesWritten
push    esi             ; nSize
push    ebx             ; lpBuffer
push    ebx             ; lpBaseAddress
push    edi             ; hProcess
call    WriteProcessMemory
push    0               ; lpflOldProtect
mov     eax, [ebp+flOldProtect]
push    eax             ; flNewProtect
push    esi             ; dwSize
push    ebx             ; lpAddress
push    edi             ; hProcess
call    VirtualProtectEx
```

BTCTimer

La función asociada a este *Timer* se encarga de revisar el contenido del portapapeles y, si tiene el formato de una dirección de una *wallet* de BTC, la sustituye por otra dirección con la finalidad de que la víctima envíe los BTC a la dirección de los autores.



Otras Versiones

En otras versiones analizadas se han observado algunos cambios en la configuración y funcionalidad que se detallan a continuación:

1. Los formularios relacionados con las entidades bancarias no se almacenan en propio binario, sino que se almacenan en librerías DLL.

2. Se detectan además las versiones concretas de Windows 10 Home y Windows Server.
3. El algoritmo de generación de dominios se basa en https://sites.google.com/view/ y la fecha y hora.
4. La persistencia no se crea en la clave de registro "Run", sino que se crea a través de un fichero ".LNK".
5. La configuración se almacena en una clave de registro.
6. Aparecen nuevos *hooks* para detectar entornos virtuales (VMWare y Virtual PC) y los procesos relacionados con "RegMon" y "Whireshark".

Técnicas MITRE ATT&CK

Initial Access	T1192	Spearphishing Link
Persistence	T1060	Registry Run Keys / Startup Folder
Defense Evasion	T1009	Binary Padding
	T1089	Disabling Security Tools
	T1140	Deobfuscate/Decode Files or Information
	T1222	File and Directory Permissions Modification
	T1036	Masquerading
	T1112	Modify Registry
	T1497	Virtualization/Sandbox Evasion
Credential Access	T1503	Credentials from Web Browsers
	T1081	Credentials in Files
Discovery	T1010	Application Window Discovery
	T1083	File and Directory Discovery
	T1057	Process Discovery
	T1063	Security Software Discovery
	T1082	System Information Discovery

Collection	T1056	Input Capture
Command and Control	T1483	Domain Generation Algorithms
	T1071	Standard Application Layer Protocol
Exfiltration	T1040	Exfiltration Over Command and Control Channel

En el Apéndice A se puede consultar el mapa de tácticas y técnicas utilizadas por Grandoleiro.

3. MITIGACIÓN

Medidas a nivel de endpoint

A diferencia de otras variantes de troyano bancario, el código de Grandoreiro no está firmado, por lo que implementar una política que no permita la ejecución de binarios que no estén firmados podría prevenir la ejecución de este troyano bancario y de otro tipo de malware. No obstante, gran cantidad de desarrolladores y paquetes de software no distribuyen sus productos firmados, por lo que esta estrategia podría no resultar práctica en algunos casos.

En concordancia con lo anterior, pero empleando mecanismos más generales, se recomienda que las organizaciones prohíban o, al menos, monitoricen la ejecución de binarios no conocidos previamente dentro de ella o aquellos no provenientes de fuentes confiables. Aunque imperfecto, por la forma en la que se crea y distribuye el software legítimo, esta medida puede servir como una alarma inicial para impulsar una mayor investigación y, posiblemente, limitar su propagación.

Medidas a nivel de red

Si se dispone de los mecanismos para inspeccionar el tráfico que ocurre dentro de la red, se debería identificar la transferencia de binarios desconocidos dentro de ella.

Por otro lado, es altamente recomendable mantener una segmentación adecuada de la red para evitar desplazamientos laterales y que finalmente se alcancen los sistemas críticos de la organización.

En adición y conociendo el comportamiento de esta familia de troyano bancario, se podría analizar el tráfico saliente y generar reglas que comprueben el cliente RTC que utiliza Grandoreiro, de forma que, si se encuentra en ejecución y se intenta conectar con el exterior, se puede parar lo antes posible o detectar al menos la intrusión.

Medidas y consideraciones adicionales

En caso de incidente con este malware, se debe de contactar con la entidad bancaria a través de un dispositivo no infectado y comunicar lo sucedido.

Aunque ante un incidente de este tipo el foco principal sea restaurar los sistemas, se debe considerar que la ejecución del troyano bancario puede buscar como objetivo ocultar la verdadera razón de la intrusión. Por tanto, se recomienda mantener una buena política de almacenamiento de registros, para poder realizar una revisión posterior. De esta manera, se podrá verificar cuáles han sido las acciones tomadas por los atacantes en todo momento y si se trata realmente de un secuestro de información o de la necesidad de eliminar su rastro.

Así mismo, es recomendable usar métodos de seguridad adicionales en el inicio de sesión para servicios como VPN y *webmail* como, por ejemplo, doble factor

de autenticación, con el propósito de evitar el robo de credenciales o de convertirse en víctima de un ataque de phishing.

También es recomendable utilizar sistemas de listas de exclusión para dificultar la ejecución inicial de código malicioso proveniente del navegador o del correo electrónico.

Además, es recomendable limitar las comunicaciones entre equipos adyacentes a excepción de aquellas estrictamente necesarias, con la finalidad de dificultar los desplazamientos laterales.

En adición, aplicar un modelo por capas del sistema Active Directory, garantizando de esta manera una segmentación de los datos según su nivel de privilegio. De esta manera solamente aquellos sistemas con altos privilegios podrán acceder a otros con menores y no al contrario, lo que ralentiza mucho las tareas de los atacantes.

En último lugar, atendiendo al comportamiento del malware y al mapeo de las técnicas y subtécnicas en relación a las distintas tácticas que propone MITRE, se puede consultar en la Matriz Enterprise cada una de las técnicas para conocer sus correspondientes medidas de detección y mitigación.

4. INDICADORES DE COMPROMISO

Los indicadores de compromiso y reglas de detección también están disponibles para su consulta y descarga en el repositorio público del Basque Cybersecurity Centre:

https://github.com/basquecscentre/technical-reports

Hashes

SHA256:

4bf7f8be989a8d521b5fbbdddfc3bfed858267d49db22c7a04d4794b8e2f0db3

02a3f0925b510c652fbe62dac87d56da816bc94a1a2e9486e209cbf076a2f976

47e1e510bcc987001d1e5879ba32c0b12ae950a54256d99dbf860486a886df3e

f40cef1c65e9979bab096a727851c46d5217e3ab1c05d0583af09b39449a65c8

0081d018c66d8a96df8688dcc0c1c9465a834b90b6fe9edd25a351665de77da9

YARA rules

```
rule win_grandoreiro_auto {

    meta:
        author = "Felix Bilstein - yara-signator at cocacoding
dot com"
        date = "2020-10-14"
        version = "1"
        description = "autogenerated rule brought to you by
yara-signator"
        tool = "yara-signator v0.5.0"
        signator_config = "callsandjumps;datarefs;binvalue"

strings:
        $sq_0 = { c3 51 894210 8b4040 891424 8bd4 83c004 }
        $sq_1 = { c1ff02 8b4dd8 66897c8a02 e9???????? c645d040
8a45f4 2a45e4 }
        $sq_2 = { 33c0 898300010000 c78304010000ff000000
8dbbb0000000 be???????? b908000000 f3a5 }
        $sq_3 = { 7512 b201 8b45f4 8b08 ff91d8000000 e9????????
8b45f4 }
```

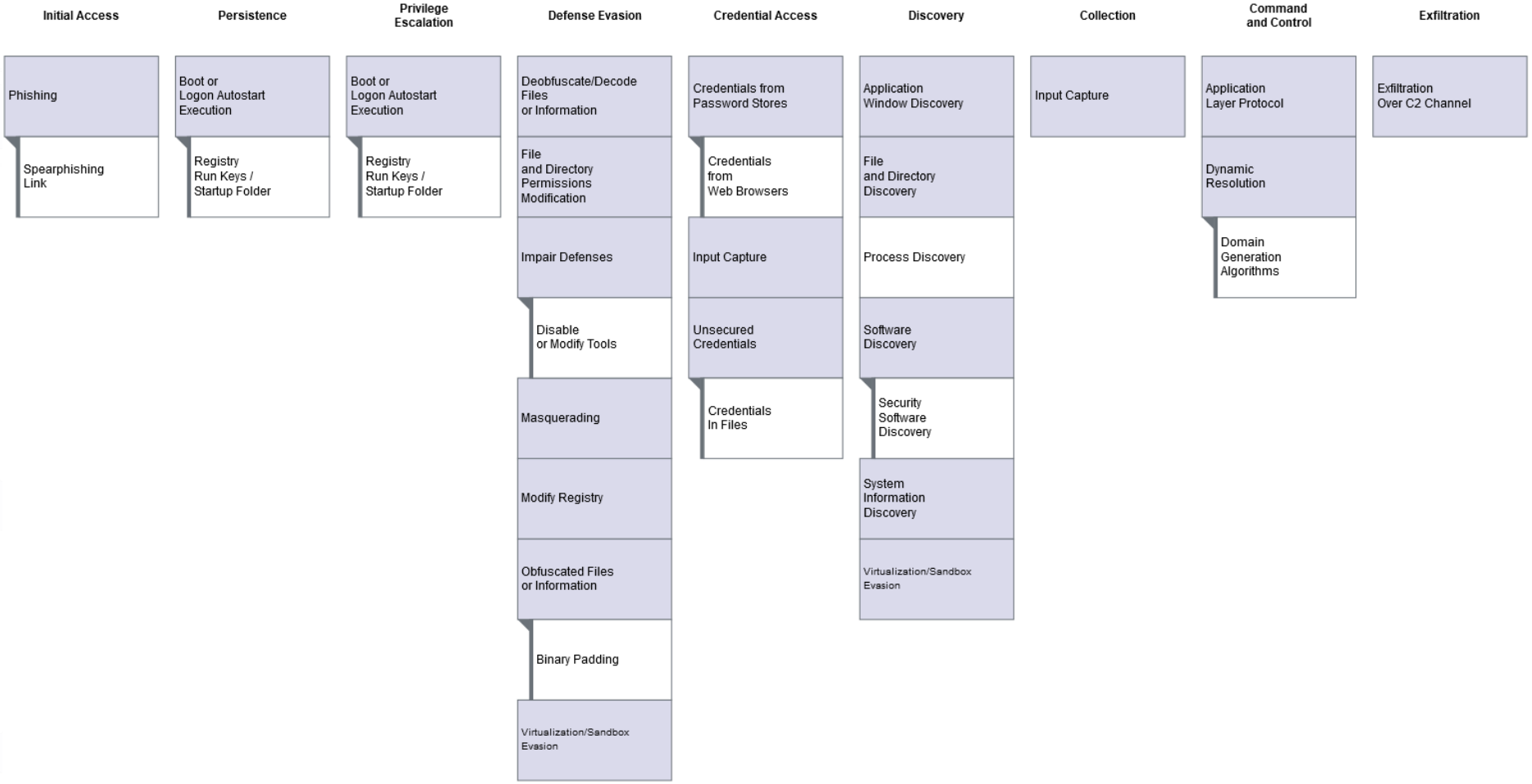
```
        $sq_4 = { eb14 8b45fc 8b4824 8b45fc 8b5030 8b45f8
e8???????? }
        $sq_5 = { 648922 8d55e8 8bc6 e8???????? 84c0
0f8400010000 84db }
        $sq_6 = { e9???????? 8b45f8 8b4068 80780800 0f84be000000
33c0 55 }
        $sq_7 = { eb08 a1???????? 094660 8bc6 8b10 ff92c0000000
c686e701000001 }
        $sq_8 = { 8b45fc eb03 8b45f4 85c0 7505 b801000000 8b55f8
}
        $sq_9 = { 0fb704adbcb77200 66014608 0fb704add4b77200
6601460a 8bd3 8bc6 e8???????? }

    condition:
        7 of them and filesize < 7602176
}
```

5. REFERENCIAS ADICIONALES

- https://www.welivesecurity.com/la-es/2020/04/28/grandoreiro-troyano-bancario-dirigido-brasil-espana-mexico-peru/
- https://www.welivesecurity.com/2020/04/28/grandoreiro-how-engorged-can-exe-get/
- https://botechfpi.com/campana-grandoreiro-mercadona

APÉNDICE A: MAPA DE TÉCNICAS MITRE ATT&CK



Reportar incidente

Si has detectado algún incidente de ciberseguridad, avísanos para que tomemos las medidas oportunas para evitar su propagación.

900 104 891

incidencias@bcsc.eus

Catálogo de ciberseguridad

¿Necesitas ayuda con tu ciberseguridad o la de tu empresa?