



SIN CLASIFICAR



Informe Código Dañino CCN-CERT ID-02/16

Win32.Gozi

Febrero de 2016

SIN CLASIFICAR

**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	4
2. RESUMEN EJECUTIVO	5
3. CARACTERÍSTICAS DEL CÓDIGO DAÑINO	5
4. DETALLES GENERALES	5
5. PROCEDIMIENTO DE INFECCIÓN.....	6
6. CARACTERÍSTICAS TÉCNICAS	6
7. OFUSCACIÓN.....	9
8. PERSISTENCIA EN EL SISTEMA	10
9. CONEXIONES DE RED	10
9.1 USER-AGENT	10
10.ARCHIVOS RELACIONADOS	11
11.DETECCIÓN	11
11.1 AUTORUNS.EXE	11
11.2 MANDIANT.....	12
12.DESINFECCIÓN.....	13
13.INFORMACIÓN DEL ATACANTE	13
13.1 WITHOUTHETERMS.COM	13
13.1.1 WHOIS	13
13.1.2DIRECCIÓN IP	14
14.REFERENCIAS	14
15.ANEXOS	15
ANEXO I - REGLA SNORT	15
ANEXO II - INDICADOR DE COMPROMISO (IOC)	16
ANEXO III – REGLA YARA	18

1. SOBRE CCN-CERT

El CCN-CERT (www.ccn-cert.cni.es) es la Capacidad de Respuesta a incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN. 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 Centro Nacional de Inteligencia, el RD 421/2004 de regulación del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad.

De acuerdo a todas ellas, el CCN-CERT tiene responsabilidad en ciberataques sobre **sistemas clasificados** y sobre sistemas de las **Administraciones Públicas** y de **empresas y organizaciones de interés estratégico** para el país. 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.

2. RESUMEN EJECUTIVO

El presente documento recoge el análisis de la familia de troyanos identificada como "**Win32.Gozi**", que ha sido diseñada para la obtención de información del equipo infectado y su envío a un servidor de Mando y Control (C2) controlado por los atacantes.

El código dañino tiene embebida una librería DLL que se encuentra ofuscada en el interior de la aplicación, siendo esta librería la que contiene realmente la carga dañina y la aplicación ejecutable.

3. CARACTERÍSTICAS DEL CÓDIGO DAÑINO

El código dañino realiza las siguientes acciones:

- Carga el código dañino en el sistema
- Aplica sistemas de persistencia
- Realiza réplicas de la aplicación dañina
- Realiza capturas de pantalla
- Ejecuta código inyectando librerías en otros procesos
- Cambia el fondo de escritorio
- Finaliza procesos
- Obtiene las cookies del navegador
- Obtiene las contraseñas de Windows Mail
- Envía información y ficheros al servidor de Mando y Control

4. DETALLES GENERALES

Las muestras analizadas se identifican con las siguientes firmas MD5:

cd2d9c7bd5de6d12718785f495ce1bb4
cc0c51ce339a29344015591c87b7dfe5

Todos son ficheros con formato PE (*Portable Executable*), es decir, son ejecutables para sistemas operativos *Windows*, concretamente para arquitecturas de 32 bits.

5. PROCEDIMIENTO DE INFECCIÓN

La infección en el equipo se produce al arrancar el fichero que contiene el código dañino. Una vez que comienza la ejecución del código dañino, realiza las siguientes acciones en el equipo de la víctima:

- Realiza una réplica de sí mismo
- Aplica métodos de persistencia
- Toma una captura de pantalla que pone como fondo de escritorio
- Finaliza la ejecución del proceso *explorer.exe*
- Ejecuta un nuevo proceso de *explorer.exe* e inyecta una librería DLL sobre él
- Se finaliza a sí mismo
- Desde el proceso *explorer.exe* se comunica con el servidor de Mando y Control (C2)

6. CARACTERÍSTICAS TÉCNICAS

El código dañino utiliza un empaquetador propio para impedir la detección por los sistemas antivirus y como mecanismo de ofuscación debido a que la carga maliciosa sólo es accesible mediante un análisis dinámico.

El código dañino, durante su ejecución, utiliza llamadas a la API del sistema nativo, es decir, funciones reservadas para el sistema operativo que habitualmente no son utilizadas por las aplicaciones. Algunas de estas funciones son las siguientes:

ZwQueryInformationProcess
NtSetContextThread
ZwOpenProcess

Cuando el código dañino se pone en ejecución emplea mecanismos de persistencia, copiándose a sí mismo y modificando el registro de Windows, de forma que se volverá a poner en ejecución si el equipo es reiniciado.

Una vez completados los mecanismos de persistencia, realiza una captura de pantalla que almacena de forma temporal en un fichero ubicado en:

%ALLUSERPROFILE%/image.bmp

Una vez realizada la captura de pantalla, el código dañino desofusca una librería DLL que tiene embebida en el código.

Posteriormente crea un nuevo proceso de la aplicación *explorer.exe* en estado suspendido. Sobre este proceso recién creado, inyecta la librería y lo pone en ejecución durante 300 milisegundos. Después suspende de nuevo la ejecución del mismo.

PUSH DWORD PTR DS:[ESI+4]	
CALL EBX	kernel32.ResumeThread
PUSH EDI	
CALL DWORD PTR DS:[405060]	kernel32.Sleep
PUSH DWORD PTR DS:[ESI+4]	
CALL DWORD PTR DS:[405104]	kernel32.SuspendThread

Ilustración 1. El proceso se ejecuta durante 300ms

La intención de crear, arrancar y parar este proceso inyectado es la de realizar correctamente la carga en memoria del mismo para que, más adelante, pueda manipular este proceso sin que se produzcan errores.

A continuación el código dañino cambia el fondo de pantalla por la captura que realiza al comienzo de la infección. De esta forma, y como se podrá observar más adelante, al finalizar la aplicación *explorer.exe* original, y con ella la desaparición del escritorio, no se percibe visualmente esta acción.

El código dañino utiliza una llamada al API de sistema *SystemParametersInfoW* pasando como argumento la ruta del archivo que debe establecer como fondo de pantalla.

DS:[0040517C]=7E419F06 (USER32.SystemParametersInfoW)				
Address	ASCII			
00406880	C:\.W.I.N.D.O.	01E9FE98	00000014	Action = SPI_SETDESKWALLPAPER
00406890	W.S.\.w.e.b.\.w.	01E9FE9C	00000000	wParam = 0
004068A0	a.l.l.p.a.p.e.r.	01E9FEA0	00406880	pParam = 1f03d447.00406880
004068B0	\.B.l.i.s.s.+b.	01E9FEA4	00000001	UpdateProfile = SPIF_UPDATEINIFILE
004068C0	m.p.....	01E9FEA8	00000001	
004068D0		01E9FEAC	7C809BE7	kernel32.CloseHandle
004068E0		01E9FEB0	00000000	

Ilustración 2. Llamada al sistema para cambiar el fondo de escritorio

Cuando el fondo de pantalla se encuentra cambiado, crea y activa un hilo en el proceso *explorer.exe* que ha creado anteriormente. Al estar suspendido el proceso, el hilo no se pone en ejecución aunque se active. Este hilo se encarga de ejecutar el código de la librería DLL que tiene inyectada.

Seguidamente finaliza la ejecución del proceso *explorer.exe*, que se ejecuta inicialmente funcionando en el sistema informático y restaura la ejecución completa del proceso creado por la aplicación dañina *explorer.exe*, que se encontraba suspendido.

El hilo que ejecuta el código dañino del nuevo proceso *explorer.exe* se encarga de eliminar la captura de pantalla que fue realizada al comienzo de la ejecución de la aplicación dañina y de restaurar el fondo de pantalla que se estuviese utilizando previamente.

A continuación crea un fichero con un nombre aleatorio de 8 dígitos numéricos y extensión .BAT en la misma ruta en la que se encuentre la aplicación dañina con el siguiente contenido:

```
attrib -r -s -h %1
:427109
del %1
if exist %1 goto 427109
del %0
```

La aplicación dañina comprueba si se está ejecutando desde la ruta de persistencia y en caso contrario ejecuta el fichero .BAT con la ruta del código dañino como parámetro.

El código del fichero .BAT quita los atributos de lectura, aplicación de sistema y oculto del fichero que se le pase como parámetro. Después trata indefinidamente de eliminar el fichero al que ha cambiado los atributos. Una vez realizado, trata de eliminarse a sí mismo.

Por último finaliza la ejecución de la aplicación que ha iniciado la infección.

En este momento la carga dañina está contenida en el proceso que se encuentra en ejecución *explorer.exe*. Desde este proceso se conecta al servidor de Mando y Control (C2) mediante peticiones HTTP.

La comunicación con el servidor de Mando y Control (C2) se realiza mediante peticiones de tipo GET para obtener información desde el servidor y peticiones de tipo POST para suministrar información desde el código dañino. Estas peticiones se encuentran ofuscadas mediante una variante de base64.

```
HTTP 389 GET /tnneoxhvib.php?dripytyxm=2aduVzsuUydExTMLmea1x2E3jFun3o1kpV
```

Ilustración 3. Petición HTTP realizada por el código dañino

En el siguiente cuadro de texto se puede observar la primera petición que realiza la aplicación dañina para darse de alta en el C2.

```
tcgutyqnis.php?qeuhamh=yITNQpm/cS/LvVPaUrAjoRX2qZj3UYg2s1jDhxNR7eSA3grgc6x4OZigFi/FqKaSTANNdsspLoFh1F+jNLZhNpLd2CteaQJKu11vDVLB1Ls79ok2an0RjP1pBo/EeeRt
```

Una vez desofuscado se observan los siguientes campos:

version	212277
user	0a1abd645ce20b1aee75506fb167fc53
server	12
id	1920
crc	2e6e739

El servidor de Mando y Control responde con una nueva cadena ofuscada que incluye la lista de tareas que debe realizar el código dañino.

La lista de tareas que puede recibir son:

- Enviar al C2 las contraseñas obtenidas
- Enviar al C2 archivos
- Descargar y ejecutar nuevas aplicaciones

La petición para darse de alta en el C2 se repite de forma periódica cada dos minutos. De esta forma el código dañino recibe nuevas indicaciones o se mantiene en espera hasta la siguiente solicitud.

Si el servidor de Mando y Control solicita que se le envíen archivos o información desde el equipo en el que se está ejecutando la aplicación dañina, ésta envía una nueva petición HTTP mediante el protocolo POST con los siguientes campos una vez desofuscados:

version	212277
user	0a1abd645ce20b1aee75506fb167fc53
server	12
id	1920
crc	<cambia en función de los datos>
wdata	<Información enviada>

Durante el análisis ha sido posible observar trazas de código y cadenas de texto que hacen referencia a un procesamiento de las cookies del equipo y la obtención de credenciales almacenadas en la aplicación Windows Mail aunque no se ha observado que se lleguen a utilizar estas funciones a pesar de encontrarse disponibles en el código dañino.

```
UNICODE "\\Mozilla\\Firefox\\Profiles\\"
UNICODE "cookies.sqlite"
UNICODE "cookies.sqlite-journal"
ASCII "Software\\Microsoft\\Windows Mail"
ASCII "Software\\Microsoft\\Windows Live Mail"
UNICODE "\\cookie.ff"
UNICODE "\\cookie.ie"
```

Ilustración 4. Cadenas de texto que hacen referencia a Windows Mail

7. OFUSCACIÓN

La aplicación dañina utiliza un sistema de ofuscación propietario, tanto para el código como para las cadenas de texto incluidas en el binario. Además se ha detectado el empaquetador UPX como método para disminuir el tamaño del ejecutable analizado y para la ofuscación del código.

El código dañino utiliza una variante de base64 en las comunicaciones que establece con el servidor de C2 impidiendo de esta forma visualizar la información intercambiada.

8. PERSISTENCIA EN EL SISTEMA

El código dañino, con el fin de asegurarse su persistencia en cada reinicio del equipo, en su primera ejecución crea una réplica de sí mismo en la siguiente ruta en caso de disponer de permisos administrativos:

```
%WINDIR%\system32\<nombre de 8 caracteres>.exe
```

Si no dispone de permisos de administrador, realiza una réplica de sí mismo en la siguiente ruta:

```
%TMP%\<nombre de 8 caracteres>.exe
```

Una vez ha realizado la réplica, añade una entrada en el registro de Windows diferente según sea el rol del usuario. Si el usuario tiene permisos de administrador crea la siguiente entrada:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\run]
"<nombre de 8 caracteres>"="%WINDIR%\system32\<nombre de 8 caracteres>.exe"
```

Si el usuario no dispone de permisos de administrador crea la siguiente entrada:

```
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\run]
"<nombre de 8 caracteres>"="%TMP%\<nombre de 8 caracteres>.exe"
```

Todos los nombres usados, tanto de claves como de nombres de ficheros, son aleatorios y, además, el nombre de fichero y de la clave de registro también son diferentes.

 atmpperf

REG_SZ

C:\WINDOWS\system32\d3drml2r.exe

Ilustración 5. Ejemplo de persistencia con permisos administrativos

9. CONEXIONES DE RED

Las conexiones establecidas utilizan el protocolo HTTP, por lo que es posible observar el contenido de la petición que se realiza, así como la respuesta, aunque, para proteger la confidencialidad de la comunicación, la aplicación dañina utiliza una versión modificada de base64 sobre estos datos para que no sean directamente inteligibles.

9.1 USER-AGENT

El código dañino emplea como User-Agent la siguiente cadena:

```
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1)
```

10. ARCHIVOS RELACIONADOS

A continuación se muestran los archivos relacionados con el código dañino:

Nombre	Fecha Creación	Tamaño bytes	Hash MD5
%WINDIR%\system32			
d3drml2r.exe	01/09/2015	294kB	cd2d9c7bd5de6d12718785f495ce1bb4
%ALLUSERSPROFILE%			
Image.bmp	01/09/2015	3MB	461ca7810f477805aca21bedd82228f2
Ruta desde la que se ejecute el código dañino			
36245553.bat	01/09/2015	74B	268a9cf1b9f8d2a5476bc436b4424c0b

El siguiente fichero es la librería DLL extraída desde la memoria del equipo, por lo que no se relaciona con ninguna ruta del equipo:

Nombre	Fecha Creación	Tamaño bytes	Hash MD5
<MEMORIA>			
DLL.DLL	16/12/2015	125,3kB	CE1FE36693DA1053A8CDB79E5A0B7C63

11. DETECCIÓN

Para detectar si un equipo se encuentra o ha estado infectado para cualquiera de sus usuarios, se recomiendan utilizar las siguientes herramientas de Microsoft Windows Sysinternals¹:

- Autoruns.exe

También se puede usar alguna de las herramientas de Mandiant como el "Mandiant IOC Finder" o el colector generado por RedLine@ con los indicadores de compromiso generados para su detección.

Se recomienda iniciar sesión con un usuario que posea **privilegios administrativos** en el sistema con el fin de determinar si el equipo se encuentra infectado por el código dañino.

11.1 AUTORUNS.EXE

Esta aplicación monitoriza aquellas aplicaciones que serán ejecutadas de forma automática durante el proceso de inicio del equipo informático.

Para comprobar si el equipo se encuentra infectado se iniciará la utilidad *autoruns.exe* y se pulsará en la pestaña *Logon*. Desde esta ventana es posible observar la ruta de registro que utiliza la aplicación dañina para establecer la persistencia, que variará dependiendo de si la infección se produjo con un usuario normal o administrador. Las dos rutas a verificar son:

¹ <https://technet.microsoft.com/en-us/sysinternals/bb545021.aspx>

[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\run]²

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\run]³

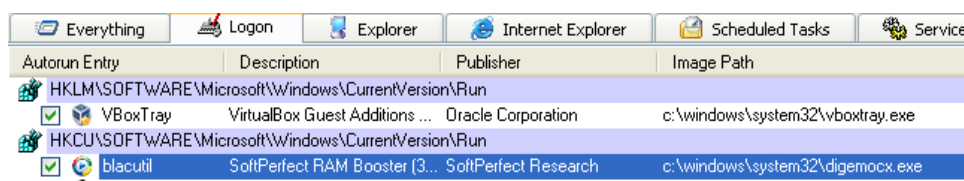
Si existen entradas de registro, hay que observar la ruta que aparece en *Imagen Path* y comprobar si se corresponde con la siguiente, en caso de ser un usuario normal:

%TMP%/<nombre de 8 caracteres>.exe

En el caso de ser un usuario administrador:

%WINDIR%\system32\<nombre de 8 caracteres>.exe

En ambos casos, los dos nombres, el del fichero y el de la clave de registro, tienen 8 caracteres alfanuméricos aleatorios.



Autorun Entry	Description	Publisher	Image Path
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run			
VBxTray	VirtualBox Guest Additions ...	Oracle Corporation	c:\windows\system32\vbboxtray.exe
HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run			
blacutil	SoftPerfect RAM Booster (3...	SoftPerfect Research	c:\windows\system32\digemocx.exe

Ilustración 6. Archivo dañino localizado mediante Autoruns

En caso de existir el fichero, se deberá identificar si es desconocido y dañino. Para ello se recomienda el uso de herramientas antivirus o multiantivirus⁴.

11.2 MANDIANT

Se ha generado un nuevo archivo indicador de compromiso adjunto al informe.

Se utilizará el indicador con alguna de las herramientas de las que dispone Mandiant como "Mandiant_ioc_finder" o para la confección de un recolector de evidencias mediante "Mandiant RedLine".

Se recomienda consultar la guía de seguridad CCN-STIC-423 *Indicadores de Compromiso (IOC)*, donde se recoge qué es un indicador de compromiso, cómo crearlo y cómo identificar equipos comprometidos.

² En el caso de haberse producido la infección con un usuario normal.

³ En el caso de haberse producido la infección con un usuario administrador.

⁴ <https://www.virustotal.com/>

<https://nodistribute.com/>

<http://v2.scan.majyx.net/>

<http://fuckingscan.me/>

12. DESINFECCIÓN

De forma automática, se aconseja la utilización de herramientas antivirus actualizadas para la desinfección del equipo. De forma manual se debe acceder al registro de Windows y buscar alguna de las siguientes entradas:

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\run] "<nombre de 8 caracteres>"="%WINDIR%\system32\<nombre de 8 caracteres>.exe"
[HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\run] "<nombre de 8 caracteres>"="%TMP%\<nombre de 8 caracteres>.exe"

Los nombres de claves y ficheros serán un literal de 8 caracteres alfanuméricos.

Una vez ha sido localizada la entrada en el Registro de Windows y comprobado que los ficheros son aplicaciones maliciosas (para lo cual se recomienda el uso de algún antivirus actualizado o un multiantivirus), se debe eliminar el fichero y la entrada del registro. Posteriormente se debe reiniciar el equipo.

En última instancia, se aconseja el formateo y la reinstalación completa del sistema informático (siguiendo lo indicado en las guías CCN-STIC correspondientes) de todos aquellos equipos en los que se haya detectado algún indicador de compromiso o encontrado algún archivo o clave de registro indicados.

13. INFORMACIÓN DEL ATACANTE

13.1 WITHOUTHETERMS.COM

13.1.1 WHOIS

El dominio *withouttheterms.com* actualmente se encuentra libre y no tiene ningún propietario.

Whois Record for WithoutTheTerms.com

Domain Available

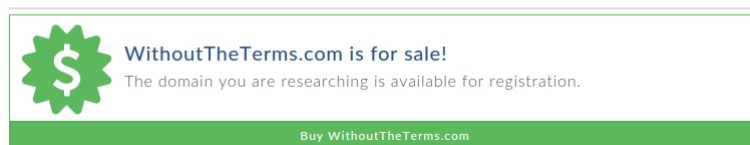


Ilustración 7. El dominio *withouttheterms.com* se encuentra disponible

Los últimos registros obtenidos para este dominio datan del 28/05/2014:

```

Domain Name: WITHOUTTHERMS.COM
Registry Domain ID:
Registrar WHOIS Server: whois.evonames.com
Registrar URL: https://evonames.com/
Updated Date: 2015-04-29 15:41:53.180461
Creation Date: 2014-05-28
Registrar Registration Expiration Date: 2015-05-28
Registrar: EVOPLUS LTD
Registrar IANA ID: 1418
Registrar Abuse Contact Email: abuse@evonames.com
Registrar Abuse Contact Phone: +1.5144959001
Reseller: https://evo.ru-tld.ru/
Domain Status: ok
Registry Registrant ID: MR_5579692WP
Registrant Name: WhoisProtectService.net
Registrant Organization: PROTECTSERVICE, LTD.
Registrant Street: 27 Old Gloucester Street
Registrant City: London
Registrant State/Province:
Registrant Postal Code: WC1N 3AX
Registrant Country: United Kingdom
Registrant Phone: +44.02074195061
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
Registrant Email: withouttheterms.com@whoisprotectservice.net
Registry Admin ID: MR_5579692WP
Admin Name: WhoisProtectService.net
Admin Organization: PROTECTSERVICE, LTD.
Admin Street: 27 Old Gloucester Street
Admin City: London
Admin State/Province:
Admin Postal Code: WC1N 3AX
Admin Country: United Kingdom
Admin Phone: +44.02074195061

```

Ilustración 8. Información del histórico de domaintools.com para withouttheterms.com

13.1.2 DIRECCIÓN IP

Actualmente el dominio *withouttheterms.com* no tiene asignada ninguna dirección IP. A continuación se muestra el historial de cambios en el direccionamiento IP que ha tenido el dominio:

IP Address History

Event Date	Action	Pre-Action IP	Post-Action IP
2014-05-29	New	-none-	69.197.18.190
2014-06-15	Change	69.197.18.190	69.197.18.184
2014-07-21	Change	69.197.18.184	69.197.18.183
2014-08-14	Not Resolvable	69.197.18.183	-none-
2014-08-19	New	-none-	212.227.252.198
2015-07-21	Not Resolvable	212.227.252.198	-none-

Ilustración 9. Historial de cambios de direccionamiento IP que ha tenido el dominio *withouttheterms.com*

14. REFERENCIAS

El código dañino examinado se ha observado de forma muy activa a principios de 2007 y es posible encontrar más información en los siguientes enlaces:

- <http://www.anvisoft.com/resources/automated-and-manual-method-to-remove-gozi-trojan/>
- <http://www.secureworks.com/cyber-threat-intelligence/threats/gozi/>

15. ANEXOS

ANEXO I - REGLA SNORT

```
trojan.rules:alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"ET TROJAN Gozi check-in / update"; flow:established,to_server;
content:"?user_id="; nocase; http_uri; content:"&version_id="; nocase;
http_uri; content:"&crc="; nocase; http_uri;
reference:url,www.secureworks.com/research/threats/gozi;
reference:url,doc.emergingthreats.net/2009410;
classtype:trojan-activity; sid:2009410; rev:4;)

trojan.rules:alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS
(msg:"ET TROJAN Gozi/Ursnif/Papras Connectivity Check";
flow:established,to_server; content:"GET"; http_method; urilen:13;
content:"/usdeclar.txt"; http_uri; fast_pattern:only; content:!"Accept";
http_header; content:!"Referer|3a|"; http_header;
reference:md5,5f3530edbe1fce44e05ad0c96e54efb4;
reference:md5,279fc5e6181d58f883a15d5089ce541b;
reference:url,krebsonsecurity.com/2013/01/three-men-charged-in-connection-with-gozi-
trojan/;
reference:url,csis.dk/en/csis/news/4472/; classtype:trojan-activity;
sid:2019380; rev:3;)
```

ANEXO II - INDICADOR DE COMPROMISO (IOC)

```
<?xml version="1.0" encoding="us-ascii"?>
<ioc
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  id="8233aeb8-6d0f-4ff6-b70d-85c39c354287"
  last-modified="2015-08-12T06:40:22"
  xmlns="http://schemas.mandiant.com/2010/ioc">
  <short_description>win32.Gozi</short_description>
  <authored_date>2015-08-31T08:31:45</authored_date>
  <links />
  <definition>
    <Indicator operator="OR" id="1ffa112b-3be1-411f-87c6-cd20ffa85b65">
      <IndicatorItem id="2ebf6622-8193-4079-bc11-774d4939472e" condition="is">
        <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
        <Content type="md5">db2f2b85a7ac86b5a8c2f2d240619210</Content>
      </IndicatorItem>
      <IndicatorItem id="6e2a0028-d196-4ee4-8a84-f6a637167ced" condition="is">
        <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
        <Content type="md5">cd2d9c7bd5de6d12718785f495ce1bb4</Content>
      </IndicatorItem>
      <IndicatorItem id="9a6fe9a2-868f-4916-a8c4-6f6eadf5b280" condition="is">
        <Context document="FileItem" search="FileItem/Md5sum" type="mir" />
        <Content type="md5">cc0c51ce339a29344015591c87b7dfe5</Content>
      </IndicatorItem>
      <Indicator operator="AND" id="fcc8112e-6cd6-4788-8600-c7debfd3984d">
        <IndicatorItem
          id="20fa87e1-4565-4aeb-9500-cb4847ba1977"
          condition="contains">
          <Context document="Network" search="Network/DNS" type="mir" />
          <Content type="string">withouttheterms.com</Content>
        </IndicatorItem>
        <Indicator operator="OR" id="f950790b-5056-4563-96ee-734afc25ae49">
          <IndicatorItem
            id="d5871938-4c5a-4a50-8e9f-4d5ed04296b6"
            condition="contains">
            <Context document="FileItem" search="FileItem/FileExtension" type="mir" />
            <Content type="string">.bat</Content>
          </IndicatorItem>
          <Indicator operator="AND" id="6c2bd416-dfea-4811-bace-726ad911d719">
            <IndicatorItem
              id="20030340-099c-4578-b99c-99d1f1f0affe"
              condition="contains">
              <Context document="FileItem" search="FileItem/StringList/string"
                type="mir" />
              <Content type="string">attrib -r -s -h %1</Content>
            </IndicatorItem>
            <Indicator operator="AND" id="8b6e2475-f38a-4241-8e0f-5e60111d1444">
              <IndicatorItem
                id="4089356d-04c1-4430-9582-b8a41c8706e2"
```

```
condition="contains">
  <Context document="FileItem" search="FileItem/StringList/string"
type="mir" />
  <Content type="string">del %1</Content>
  </IndicatorItem>
</Indicator>
</Indicator>
</Indicator>
</Indicator>
<Indicator operator="OR" id="226de452-69d8-4f43-8af4-ff6d4892638a">
  <IndicatorItem id="383d24fd-171f-4f88-9875-237e1ee7514c"
condition="contains">
    <Context document="RegistryItem" search="RegistryItem/Path" type="mir" />
    <Content
type="string">SOFTWARE\Microsoft\Windows\CurrentVersion\run</Content>
    </IndicatorItem>
    <Indicator operator="AND" id="d4531315-5830-4542-b696-af0ae0fa390c">
      <IndicatorItem id="f596251c-e3e4-440f-a50e-b60642af681c"
condition="contains">
        <Context document="RegistryItem" search="RegistryItem/Value" type="mir"
/>
        <Content type="string">C:\windows\system32</Content>
        </IndicatorItem>
      </Indicator>
    </Indicator>
  </Indicator>
</definition>
</ioc>
```

ANEXO III – REGLA YARA

Utilizando sobre la memoria de un equipo la siguiente firma YARA, es posible comprobar si el sistema se encuentra infectado.

```
rule GoziRule : Gozi Family {
meta:
  description = "Win32.Gozi"
  author = "CCN-CERT"
  version = "1.0"
strings:
  $ = {63 00 6F 00 6F 00 6B 00 69 00 65 00 73 00 2E 00 73 00 71 00 6C 00 69 00 74
00 65 00 2D 00 6A 00 6F 00 75 00 72 00 6E 00 61 00 6C 00 00 00 4F 50 45 52 41 2E 45
58 45 00}
condition:
  all of them
}
```