

El orden de los *bits* sí altera el producto: Talón de Aquiles de los Antivirus.

González Luis Fernando.
lfgonzalez@iq-online.com
 Image Quality Outsourcing S.A.

Resumen—El artículo presenta los resultados obtenidos en una prueba de vulnerabilidad realizada a varios software antivirus comerciales y no comerciales producto de la necesidad de la compañía, haciendo énfasis en la cambiante vulnerabilidad resultante de la combinación o alteración de algunos bits en los virus informáticos; enmascarando la potencial amenaza del virus en un archivo inofensivo para el antivirus.

Índice de Términos— Antivirus, Cifrado, Firmas, Offset, Heurísticos, Prueba de vulnerabilidad

I. INTRODUCCIÓN

El nivel primario de seguridad ofrecido a la información almacenada en un computador personal, o en un servidor corporativo, es un software con la capacidad de detectar y analizar comportamientos inusuales en el sistema que podrían causar daño a la información o al mismo sistema; software comúnmente conocido como antivirus. La tranquilidad y confiabilidad que estas aplicaciones dan al usuario para asegurar sus sistemas en gran porcentaje es positivo, y mas aun cuando cuentan con actualizaciones periódicas y detalladas de firmas o patrones de comportamiento que los protegen contra las últimas amenazas informáticas, en algunos casos como los corporativos la confianza en estas aplicaciones es mayor dado que cuentan con el apoyo de la casa distribuidora del software para analizar cualquier incidente que se pueda presentar con una amenaza informática.

Pero que pasaría ¿si este primer anillo de seguridad es violentado? o si el control ofrecido por

estas aplicaciones no fuera eficaz para algunas amenazas, seguramente se materializarían diversos y grandes riesgos en la plataforma tecnológica, se procedería a actualizar la base de datos de firmas del antivirus analizando todo el sistema en busca del virus y en el mejor de los casos se solicitaría apoyo a la casa que distribuyó el software para que nos brindara la ayuda necesaria; pero todas estas acciones se realizarían “Post Mortem”.

La mayoría de estas excepciones son el resultado de leves alteraciones en el cuerpo del virus producto de patrones aleatorios que estos incorporan para evadir los antivirus, pero aun mejor estos patrones de cambio son escogidos cuidadosamente de acuerdo al antivirus que se desee violentar dado que no todos los antivirus se pueden evadir con la misma técnica.

El artículo se enfoca en explicar la técnica con la cual se alternan los virus y como se realizó la prueba de vulnerabilidad [1] a 27 software antivirus el 21 de Febrero del presente año en la compañía.

II. EXPLICACIÓN DETALLADA DE LA AMENAZA INVOLUCRADA EN LA PRUEBA DE VULNERABILIDAD

Para alterar un virus de tal modo que sea indetectable al control del antivirus, se deben modificar unos cuantos bits en el archivo de tal modo que la firma que se genera del archivo difiera de las firmas que poseen los antivirus, pero la parte fundamental para que esta alteración sea optima y permita que el archivo sea funcional e indetectable posterior al cambio es saber en que offset's se debe inyectar la modificación, solo basta con cambiar algunos bits de posición si así se requiere y la tarea quedaría cumplida. En otras palabras y siendo mas idealista se podría afirmar que la ley matemática para la suma y multiplicación la cual reza que “El

orden de los factores no altera el producto” no es aplicable para la suma, multiplicación o cualquier operación algorítmica entre bits que generen un hash o firma como es el caso de la identificación de virus, dado que con unos cuantos movimientos posicionales de bits el resultado no es el mismo, la firma de identificación del virus es alterada, y el producto podría llegar a ser la contaminación total del sistema.

Vale la pena reforzar que no en cualquier segmento del archivo virus se puede realizar la alteración de los bits o bytes si se trabaja en hexa, dado que si se realizará el cambio en los offset's vitales del archivo, este quedaría inservible y si por el contrario se realizara la alteración en algún offset que no fuera parte de la firma o del patrón de identificación, el archivo simplemente quedaría modificado pero aun seguiría siendo detectable para los antivirus. Lo ideal es conseguir que el archivo sea sutil y eficazmente alterado sin afectar su estabilidad.

A. Modificando el virus

Continuando con este orden de ideas, a continuación se ilustrará a modo de ejemplo como se realiza la alteración en un virus ejecutable. Para efectos de la práctica esta actividad se desarrolla apoyado en dos herramientas: la primera Cactus Methamorph [2] con la cual se realizará la alteración del archivo y la segunda es un editor hexadecimal [3] con la cual se analizaran los offset's que han sido alterados.

1) Alteración mediante Cactus Methamorph

Esta herramienta provee la facilidad de elegir el modo en que se desea alterar los offset's del archivo virus, a modo diccionario o a modo Clonación.

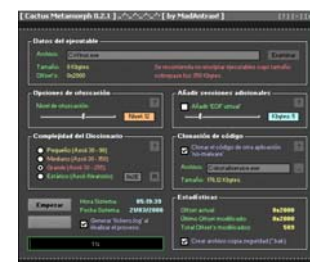
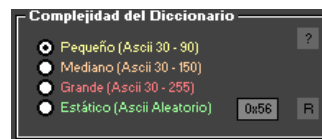


Fig. 1

ALTERACIÓN POR MODO DICCIONARIO

El modo diccionario (Fig.1) permite alterar bits de manera aleatoria en los offset's o secciones que la aplicación determine, se proponen tres tipos de diccionario, entre mas grande sea el diccionario mayor será la alteración en el archivo.

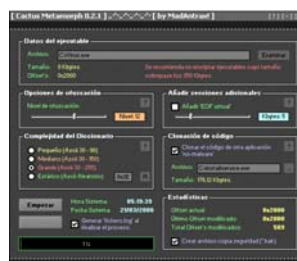


Fig. 2

ALTERACIÓN POR MODO CLONACIÓN

El modo clonación (Fig.2) permite alterar bits en los offset's seleccionados con fragmentos de bits o código de otra aplicación benigna, buscando evadir así algunos antivirus. Para el caso de este ejemplo se escoge la opción de alteración por clonación.

La aplicación incorpora otras funcionalidades como el nivel de ofuscación que es útil para evaluar que tan alterado va a quedar el archivo, de igual forma permite adicionar segmentos de código u offset's adicionales para aumentar el tamaño del archivo (Fig.3). Para el caso de este ejemplo el nivel de ofuscación será de 12 en una escala de 1 a 20 siendo 1 el valor mas bajo de ofuscación y 20 el mas alto, no se adicionarán offset's para aumentar el tamaño de archivo.



Fig. 3
NIVEL DE OFUSCACIÓN - ADICIÓN DE OFFSET'S

Definido el archivo virus ha alterar y el archivo que proporcionará los segmentos de offset's para ofuscar el contenido del virus, se procede ha ejecutar la tarea de modificación (Fig.4).

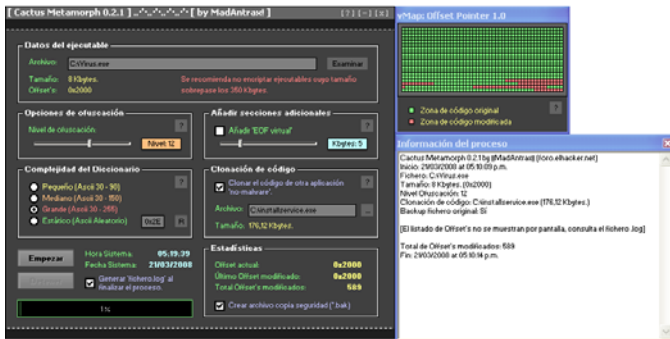


Fig. 4
EJECUCIÓN DEL PROCESO DE ALTERACIÓN

Alterado ya el archivo se puede evidenciar el cambio mediante el mapa genérico de código (Fig.5). Las zonas de color rojo son aquellas que han sido modificadas producto de la clonación con otro archivo.

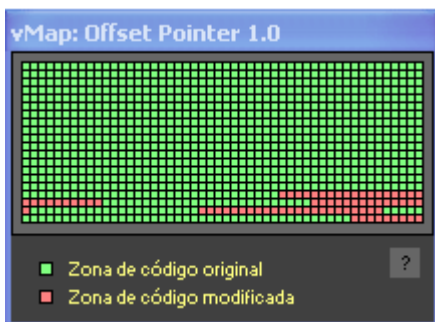


Fig. 5
MAPA GENERICO DE CODIGO

El total de offset's modificados en el archivo fueron 589, esta información se almacena en un log que genera la aplicación posterior al cambio (Fig.6).

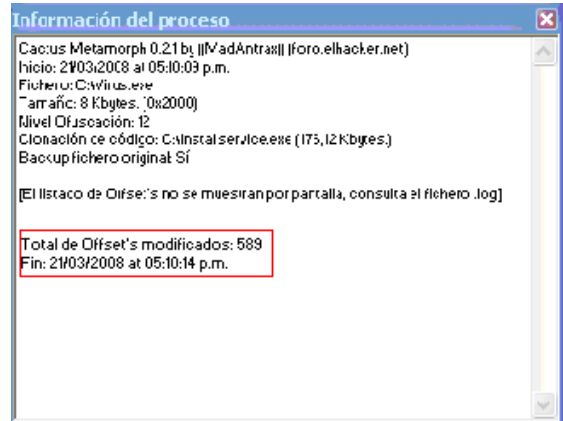


Fig. 6
INFORMACIÓN DEL PROCESO

2) Análisis del cambio mediante un editor hexadecimal

Para evidenciar el cambio realizado en los offset's, se compara el archivo ejecutable alterado con el archivo original mediante el editor hexadecimal (Fig.7).

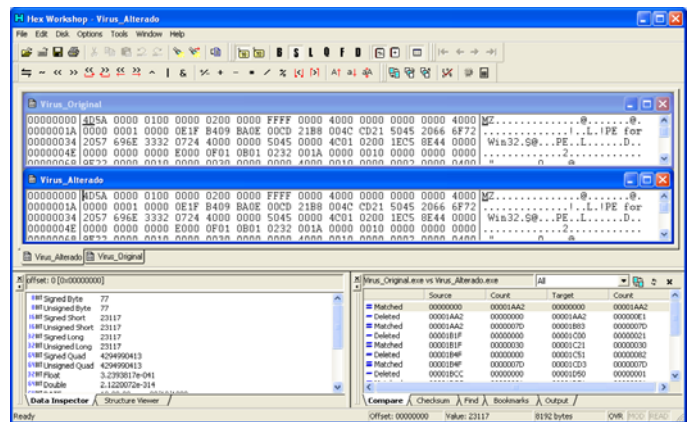


Fig. 7
EDITOR HEXADECIMAL

De los 589 Offset's modificados se tomaron dos segmentos que mejor ejemplificaran el cambio en referencia.

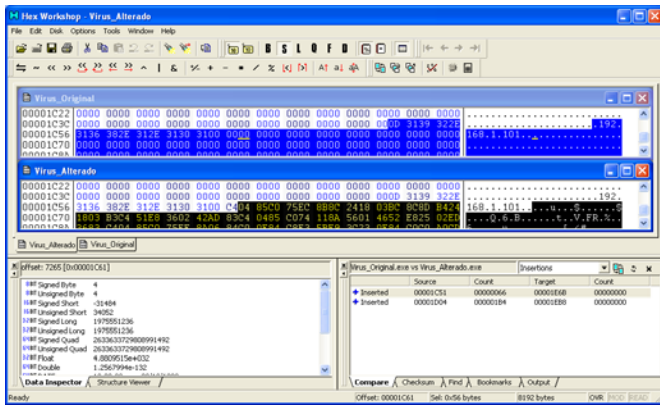


Fig. 8

COMPARACION DE ARCHIVOS EN EL EDITOR HEXADESIMAL

El primer segmento comprendido entre el Offset 0x1C61 hasta el Offset 0x1CB7 (Fig.8).

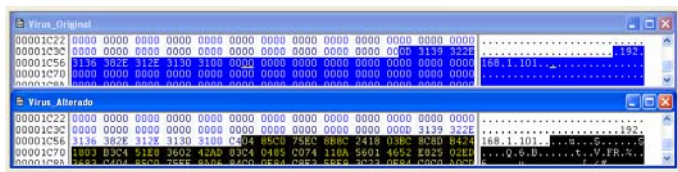


Fig. 9

COMPARACIÓN PRIMER SEGMENTO

En este caso se evidencia una inserción de 86 bytes en el segmento (Fig.9).

El segundo segmento es comprendido entre el Offset 0x1FBC hasta el Offset 0x1FFF que es el final del archivo.

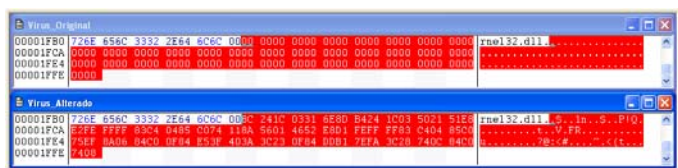


Fig. 10

COMPARACIÓN SEGUNDO SEGMENTO

En este caso se evidencia una remplazo de 69 bytes con respecto al archivo original (Fig.10).

Alterado el archivo de la forma como se ejemplificó, el virus esta listo para ser evaluado por los antivirus y posiblemente esta amenaza se materializará si el antivirus no tiene patrones de detección robustos que no solo se guíen en la firma sino que adicional incorporen patrones de detección

heurísticos y realicen comprobaciones en memoria.

III. PRUEBA DE VULNERABILIDAD

Referenciado el funcionamiento de la alteración de los virus, se procede ha argumentar el desarrollo de la prueba de vulnerabilidad que se ejecutó con 10 algoritmos de cifrado de virus y los cuales sus resultados fueron cotejados con 27 aplicaciones antivirus observando su comportamiento.

Esta prueba surgió de la necesidad de evaluar el software antivirus adquirido por la compañía. El alcance de esta prueba se amplió a otros antivirus con el ánimo de presentar y valorar posibles soluciones producto de unos resultados veraces y efectivos. Es de resaltar que previo a esta prueba se evaluaron otras alternativas que validaran la efectividad del antivirus, pero se evidenció que estas no proporcionaban el alcance deseado, como es el caso del laboratorio propuesto por Eicar [4], y en otros casos se encontró teoría que hacia referencia a pruebas de vulnerabilidad similares pero carecían de validez en el tiempo [5].

A. Adecuación de la Prueba

1) Virus encargado de la evaluación.

La labor de intentar vulnerar cada uno de los antivirus estuvo a cargo de una herramienta orientada a prestar un servicio cliente – servidor donde su principal función es tomar el control total de la máquina victima, y para lo cual genera un archivo tipo virus troyano de conexión inversa remota que hace las veces de servidor.

Las bondades ofrecidas por la herramienta Poison Ivy [6] se adaptaban a las necesidades de la prueba (Fig.11).

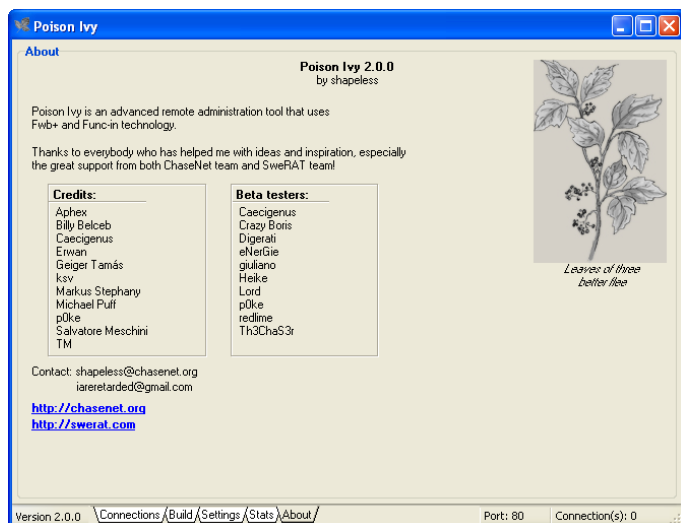


Fig. 11
APLICACIÓN POISON IVY

Como resultado la herramienta crea un archivo ejecutable no mayor a 10 Kb, siendo inversamente proporcional su tamaño físico con la potencial amenaza que este representa, su labor principal es heredar los permisos de la cuenta que esta operando en el momento de su ejecución y así poder ceder el control total de la maquina a quien está al otro lado de la conexión, para ubicar el archivo en la maquina victima existen varias alternativas, por nombrar una de las tantas se puede enviar mediante correo electrónico, posicionar en una carpeta compartida si se esta trabajando en ambiente de red y persuadir su ejecución mediante técnicas de ingeniería social escritas o verbales, para que sea mucho mas creíble para la victima el archivo se clona con algún otro archivo benigno pudiendo ser este una presentación en PowerPoint .

Tal cual como es creado el virus es una presa muy fácil para cualquier antivirus dado que los patrones y firmas de comportamiento de esta herramienta es bien conocida por la mayoría de los software antivirus, pero el punto clave es hacer indetectable ese pequeño archivo de gran potencial, y es hay cuando la prueba de vulnerabilidad toma gran validez evaluando realmente el eficaz control que puede ofrecer el antivirus.

2) Antivirus participantes en la prueba.

Los 27 software antivirus que participaron en la

prueba se adquirieron de distintas fuentes de distribución (Tabla I).

TABLA I
VERSION Y ORIGEN DE DISTRIBUCION DE LOS ANTIVIRUS INVOLUCRADOS EN LA PRUEBA

Software Antivirus	Origen de la Distribución del Antivirus	Versión
Ahnl-ab-V3	http://global.ahnlab.com/	2008.2.22.0
AntiVir	http://antivir.es/cms/	7.6.0.67
Avast	http://www.avast.com/esp/download-avast-home.html	4.7.1098.0
AVG	CD de instalación	7.5.0.516
BitDefender	http://www.bitdefender.es/	7.2
ClamAV	http://www32.clamav.net/	0.92.1
DrWeb	http://download.drweb.com/	4.44.0.09170
eSafe	http://www.aladdin.com/esafe/anti-virus.aspx	7.0.15.0
eTrust-Vet	http://www.mininova.org/for418677	31.3.5552
Ewido	http://www.zonavirus.com/datos/descargas/190/Ewido_Security_Suite.asp	4.0
F-Prot	http://www.f-prot.com/download/	4.4.2.54
F-Secure	http://www.f-secure.com/home/user/support_and_downloads/	6.70.13260
Ikarus	http://www.ikarus-software.at/	13.1.1.20
Kaspersky	CD de instalación	7.0.0.125
McAfee	CD de instalación	5234
NOD32v2	CD de instalación	2.8.9.3
Norman	CD de instalación	5.80.02
Panda	CD de instalación	9.0.0.4
Prevx1	http://www.abcdatos.com/programas/programa/z6393.html	V2.3.0.10
Rising	http://www.brothersoft.com/rising-antivirus-2007-download-57129.html	19166
Sophos	http://esp.sophos.com/	4.26.0
Sunbelt CounterSpy	http://www.sunbelt-software.com/Business/CounterSpy-Enterprise/	3.0
Symantec-Norton	CD de instalación	10
TheHacker	http://www.badongo.com/es/file/3681107	6.2.9.225
TrendMicro	CD de instalación	7.3
VBA32	http://www.anti-virus.by/en/	3.12.6.1
VirusBuster	http://www.virusbuster.hu/en/downloads/	4.3

Cada antivirus se actualizó en sus firmas y patrones de detección con fecha 21 de Febrero del 2008, fecha en la cual se desarrolló la prueba de vulnerabilidad.

3) Adecuación de las maquinas físicas y virtuales participantes en la prueba.

Se adecuaron dos maquinas físicas y cuatro maquinas virtuales en donde se ejecutaron las pruebas y en las cuales se instalaron la mayoría de los software antivirus a evaluar, previo a la prueba se garantizó que el rendimiento de las maquinas fuera el ideal para cada motor antivirus.

Las maquinas físicas y virtuales fueron adecuadas con sistema operativo Windows XP Profesional con SP2.

4) El sabor dulce de la prueba.

Buscando cumplir el objetivo de la prueba y posterior de un minucioso análisis comparativo se eligió una herramienta [7] que ofrece 10 distintos tipos de cifrado para evasión de antivirus, herramienta que hará las veces de juez y parte dado que gracias a sus características se evaluará que tanto puede modificar o alterar los archivos tipo virus y siendo juez al poner en evidencia la vulnerabilidad de los antivirus que no logren

detectar las amenazas producto de sus características de cifrado (Fig.12).



Fig. 12
FIRMAS DE CIFRADO DE VIRUS

La herramienta realiza funciones de alteración en el cuerpo del archivo como se registró en la sección II, pero ahora siguiendo un patrón definido según el algoritmo de cifrado seleccionado. El cifrado realizado a los diferentes segmentos de offset's en el cuerpo del archivo es de doble vía, de tal modo que al momento de ser revisado por el antivirus este no detecte el contenido malicioso que lleva inmerso, pero cuando este se ejecute en memoria, realice proceso de descifrado y ejecute correctamente la lógica embebida en el mismo.

B. Ejecución y Resultados de la Prueba

Una vez adecuada la plataforma para la prueba se procedió en primera instancia a crear el archivo de conexión inversa remota mediante la consola de Poison Ivy (Fig.13), el archivo se creo para que se conectara por el puerto 80 a la dirección IP donde estaba instalada la consola cliente.

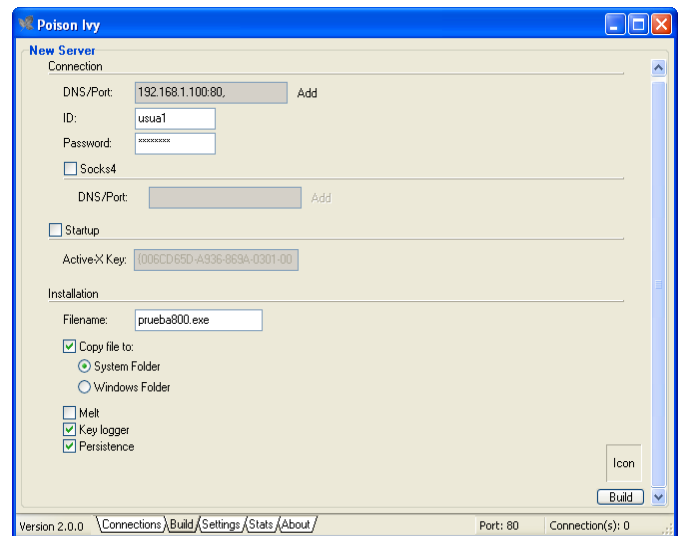


Fig. 13
INTERFAZ DE CONFIGURACIÓN Y CREACIÓN DEL ARCHIVO TIPO VIRUS TRYANO

Una vez creado el archivo Virus.exe, se procedió a cifrarlo para hacerlo indetectable con cada uno de los 10 algoritmos que ofrece la herramienta de cifrado: Rijndael, Blowfish, Des, Gost, Skipjack, Tea, Twofish, Xor, Eqv, Not.

Se inició por cifrar el archivo con la firma Rijndael (Fig.14), y así sucesivamente con cada uno de los 9 algoritmos restantes.



Fig. 14
PROCESO DE CIFRADO DEL VIRUS CON LA FIRMA RIJNDAEL

De la ejecución reiterada de esta herramienta con cada uno de los 10 algoritmos, se obtuvieron 10 archivos cifrados (Fig.15).

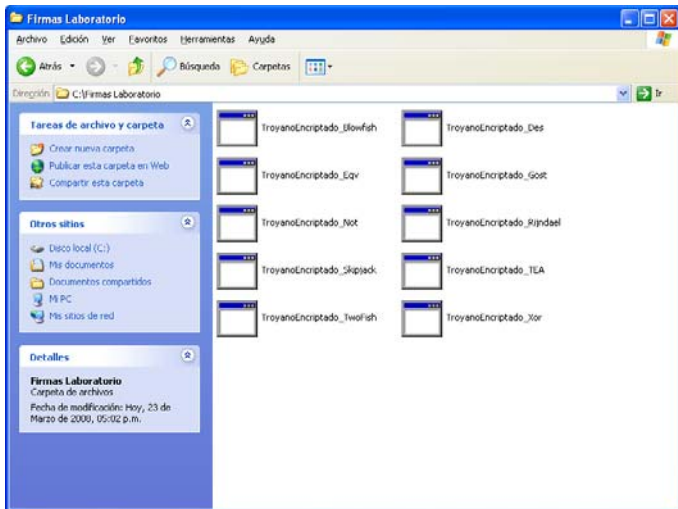


Fig. 15

ARCHIVOS EJECUTABLES CIFRADOS CON CADA UNA DE LAS 10 FIRMAS

Para efectos de esta práctica se registra el análisis comparativo realizado al archivo cifrado con DES evaluando los cambios frente al archivo virus original.

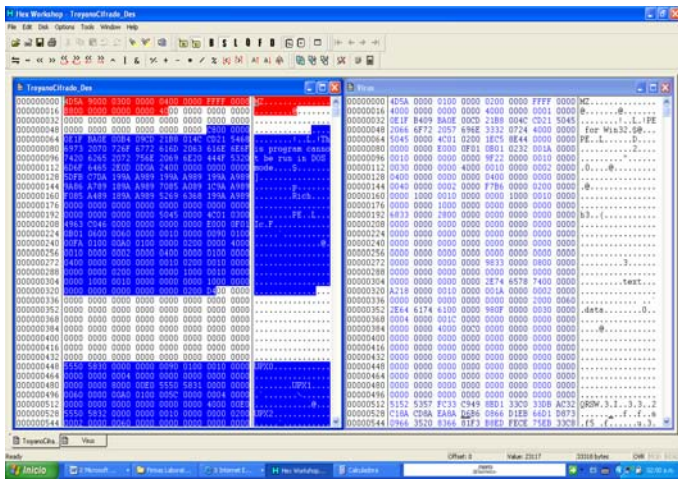


Fig. 16

ANÁLISIS COMPARATIVO DEL ARCHIVO VIRUS CIFRADO CON ALGORITMO DES Y EL ARCHIVO VIRUS SIN CIFRAR

En el lado izquierdo se visualiza el archivo virus cifrado con DES en el lado derecho se visualiza el archivo virus original (Fig.16). Producto de la comparación que realiza la herramienta los cambios entre uno y otro archivo se visualizan con diferentes colores, el color rojo significa eliminación e inserción de offset's el color azul significa únicamente inserción de offset's.

Los cambios realizados fueron los siguientes:
Se realizó una eliminación e inserción del

segmento en el archivo cifrado desde el offset 0 hasta el offset 25, es de anotar que este primer segmento está comprendido por offset's vitales ya que es el inicio del archivo, el procedimiento ejecutado por la herramienta no altero los dos primeros offset ni el offset 25 dado que si lo hace el archivo queda inservible y se desvirtuaría la funcionalidad del procedimiento. Del offset 61 al offset 333 se realizó una inserción con datos predefinidos, esta fracción hace parte de la firma genérica del virus. Del offset 448 al 568 se realizo una inserción empaquetada mediante UPX [8]. El verdadero trabajo de cifrado se inicia desde el offset 621, dado que de hay en adelante se registra información primordial como la cadena de conexión del troyano (Fig.17), la ruta de alojamiento del mismo e información del proceso que obviamente hace parte de las características del virus y no debe ser detectado por el antivirus; de igual forma operan los nueve mecanismos restantes de cifrado ofrecidos por la herramienta variando lógicamente en el algoritmo.

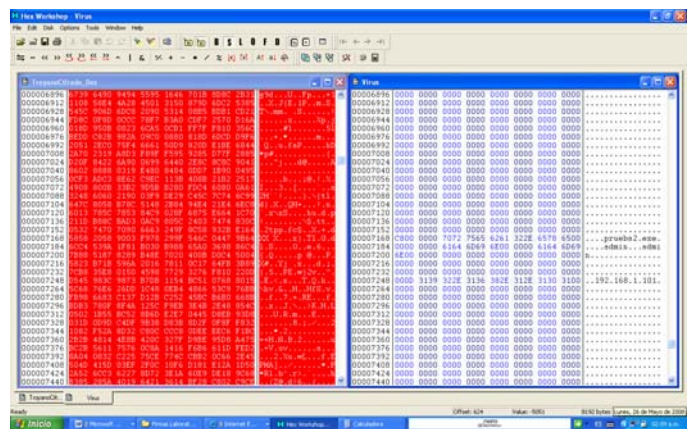


Fig. 17

CIFRADO DE LA CADENA DE CONEXIÓN DEL VIRUS TROYANO

Una vez creados los archivos cifrados, se procedió a escanear cada uno de los 27 antivirus seleccionados, tomando atenta nota del comportamiento individual en la matriz de resultados (Tabla II).

A modo de referencia del proceso que se ejecutó en la prueba, se registran 4 procesos de escaneo de virus por distintos antivirus (Fig.18) (Fig.19) (Fig.20) (Fig.21).

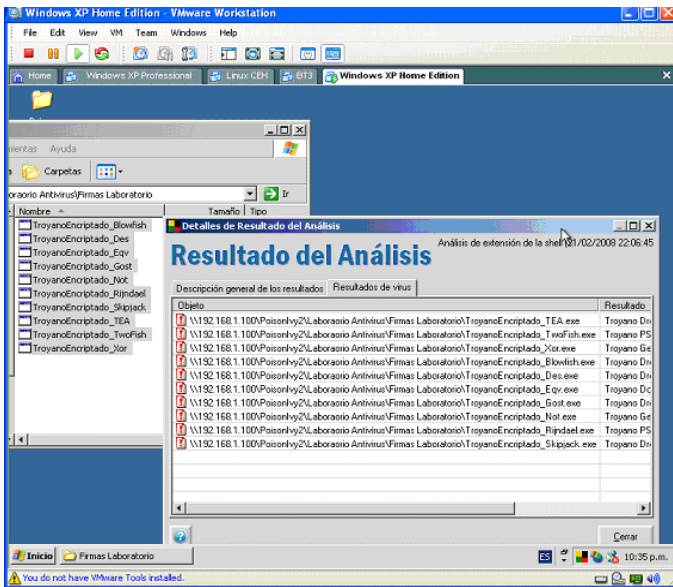


Fig. 18
ESCAÑO DE ARCHIVOS CON AVG

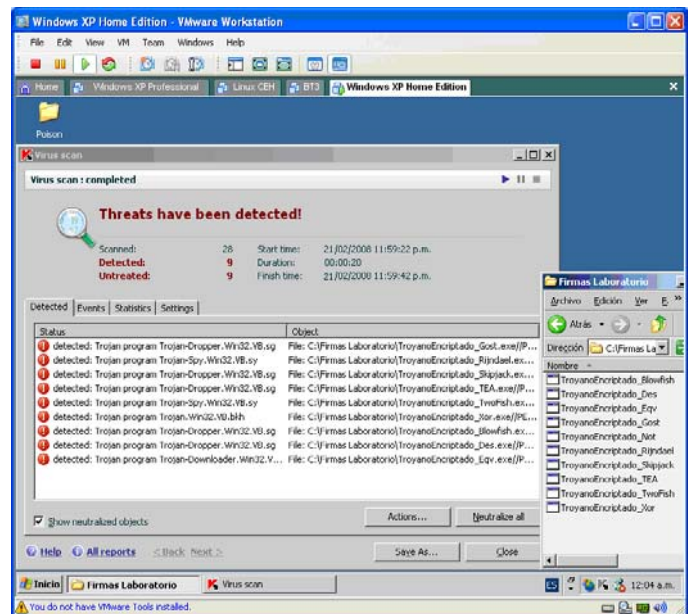


Fig. 20
ESCAÑO DE ARCHIVOS CON KASPERSKY

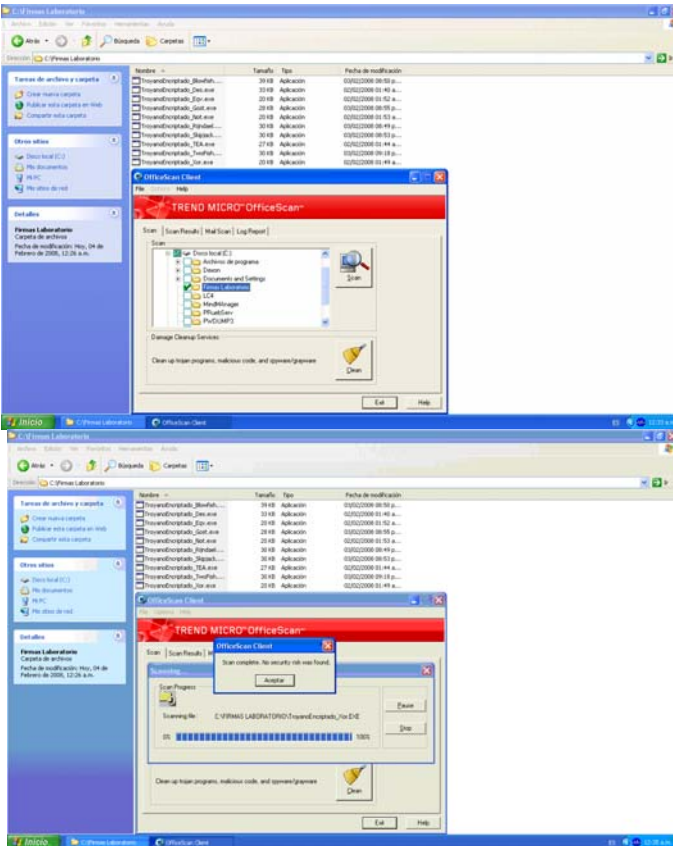


Fig. 19
ESCAÑO DE ARCHIVOS CON TRENDMICRO

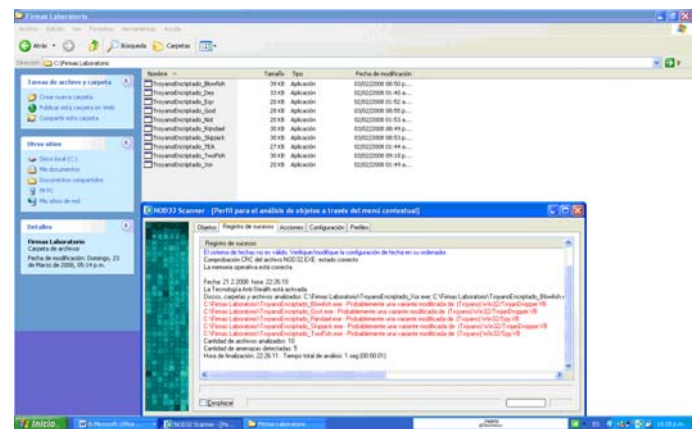


Fig. 21
ESCAÑO DE ARCHIVOS CON NOD32

C. Análisis de los Resultados

Culminada la prueba, de la matriz de resultados (Tabla II) se puede deducir que:

- 8 antivirus lograron detectar y mitigar el 100 % de las amenazas.
 - Antivir
 - Avg
 - eSafe
 - F-Secure
 - Ikarus
 - Norman
 - Panda

- Sophos
- 3 antivirus lograron detectar y mitigar el 90 % de las amenazas.
 - Ewido.
 - Kaspersky.
 - VBA32

Es de anotar que el antivirus Kaspersky no detectó la amenaza cifrada con el algoritmo Not en el momento del escaneo (Fig.20), pero al momento de ejecutar el archivo el modulo de comprobación de memoria sí logró detectarlo, evitando así la conexión remota y la materialización de la amenaza (Fig.22).

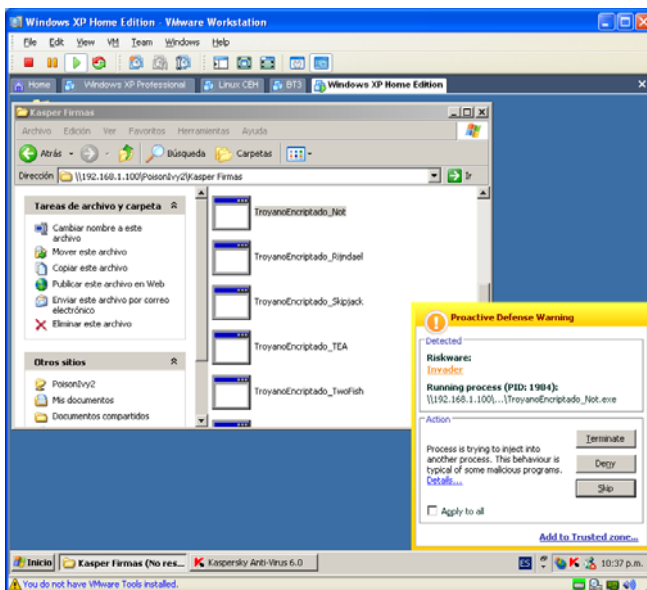


Fig. 22

COMPROBACIÓN EN PROCESOS DE MEMORIA DE KASPERSKY

- 1 antivirus logró detectar y mitigar el 80% de las amenazas.
 - TheHacker
- 2 antivirus lograron detectar y mitigar el 60 % de las amenazas.
 - Avast
 - Symantec-Norton
- 1 antivirus logró detectar y mitigar el 50% de las amenazas.
 - NOD32

- 1 antivirus logró detectar y mitigar el 40% de las amenazas.
 - ClamAV
- 4 antivirus lograron detectar y mitigar el 20 % de las amenazas.
 - BitDefender
 - F-Prot
 - McAfee
 - Prevx1
- 2 antivirus lograron detectar y mitigar el 10 % de las amenazas.
 - DrWeb
 - Rising
- 5 antivirus no lograron detectar alguna de las amenazas.
 - AhnLab-V3
 - eTrust-Vet
 - Sunbelt-CounterSpy
 - TrendMicro
 - VirusBuster

De los 27 Software antivirus, solo 9 incluido Kaspersky no son vulnerables a estas amenazas, los 18 restantes por alguno de los 10 algoritmos de cifrado de virus puede ser vulnerados.

Es importante hacer claridad que previo a la ejecución de las pruebas cada uno de los antivirus fue actualizado en sus firmas y patrones de detección a fecha del 21 de Febrero del 2008 y su versión de instalación para cada uno se referencia en la Tabla I. A la fecha de la lectura de este artículo las vulnerabilidades evidenciadas quizá ya hayan sido mitigadas por cada uno de los antivirus participantes. Los resultados de la prueba así como la información contenida en este artículo no tiene como objetivo principal promocionar o desacreditar algún software antivirus, solo busca mostrar la realización y resultados de una prueba de vulnerabilidad veraz que se ejecutó en un lapso de tiempo determinado, se deja a la interpretación del lector la calificación cuantitativa o cualitativa que se le quiera dar a cada uno de los antivirus

participantes producto de los resultados.

D. Materialización del Riesgo.

Para evidenciar la forma en la cual es materializado el riesgo en la plataforma y a modo de referencia en el artículo, se eligió un antivirus participante y un virus cifrado el cual vulnerara dicho antivirus.

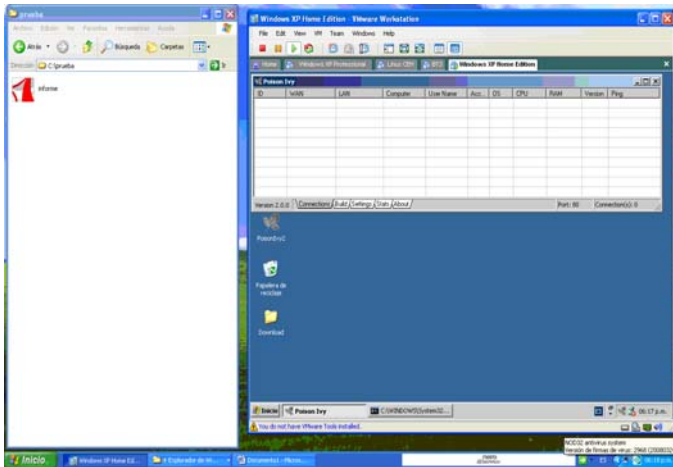


Fig. 23
EVASIÓN DEL ANTIVIRUS

La figura 23 muestra en la parte derecha la maquina virtual con la consola de administración del virus la cual cumple las veces de cliente, en la parte izquierda de observa la maquina física con un archivo tipo pdf el cual tiene incorporado el virus cifrado con el algoritmo NOT, en la parte derecha abajo se observa el antivirus ejecutándose normalmente, sin detectar el virus cifrado.

Si la maquina afectada fuera la del administrador de red o de dominio de una compañía, el riesgo seria incalculable.

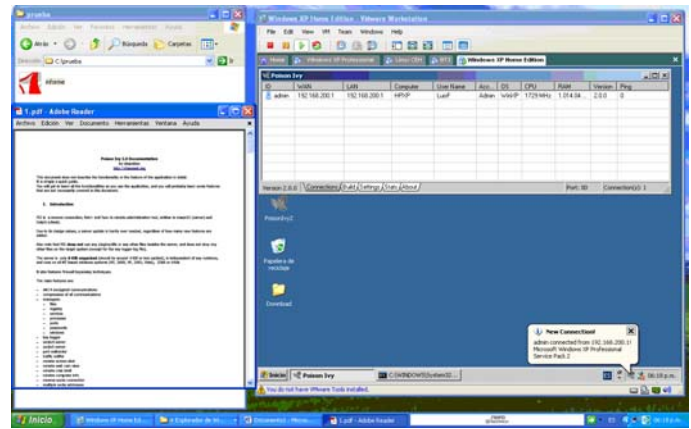


Fig. 24
EJECUCIÓN Y CONEXIÓN DEL TROYANO

Se ejecuta el archivo pdf el cual se abre normalmente, el usuario que estuviese en esa estación no vería nada anormal dado que el antivirus no reporta ninguna anomalía y el archivo pdf se puede manipular común y corriente. En la maquina virtual, la conexión se realizó y los privilegios de la cuenta que ejecutó el archivo pdf quedaron disposición del atacante (Fig.24).

Obtenidos los privilegios de la cuenta, la maquina queda a merced del atacante, pudiendo realizar todas las actividades inherentes de administración (Fig.25).

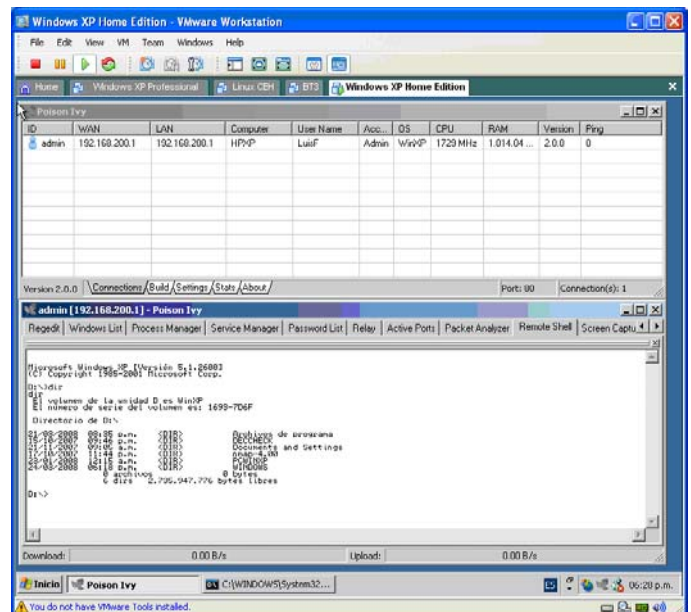


Fig. 25
ADMINISTRACIÓN DE LA MAQUINA VICTIMA POR EDITOR DE LINEAS DE COMANDOS

TABLA II
MATRIZ DE RESULTADOS EVALUACIÓN DE ANTIVIRUS

Software Antivirus	Evaluación Virus cifrado Firma RIJNDAEL	Evaluación Virus cifrado Firma TWFISH	Evaluación Virus cifrado Firma BLOWFISH	Evaluación Virus cifrado Firma DES	Evaluación Virus cifrado Firma EQV	Evaluación Virus cifrado Firma GOST	Evaluación Virus cifrado Firma NOT	Evaluación Virus cifrado Firma SKIPJACK	Evaluación Virus cifrado Firma TEA	Evaluación Virus cifrado Firma XOR
AhnLab-V3										
AntiVir	TR/Spy.VB.SY.2	TR/Spy.VB.SY.2	TR/Drop.VB.SG.37	TR/Drop.VB.SG.22	TR/Dropper.Gen	TR/Drop.VB.SG.3	TR/Crypt.XDR.Gen	TR/Drop.VB.SG.32	TR/Drop.VB.SG.4	TR/Dropper.Gen
Avast	Win32:VB-FLF	Win32:VB-FLF	Win32:Trojan-gen (UPX)		Win32:VB-GVE	Win32:Trojan-gen (UPX)	Win32:Poison-gen			
AVG	PSW.Generic5.JPU	PSW.Generic5.JPU	Dropper.Generic.QLY	Dropper.Generic.QOR	Downloader.Generic6.WJC	Dropper.Generic.QLX	Generic9.AQCB	Dropper.Generic.QOU	Dropper.Generic.QOG	Generic9.AQCA
BitDefender	Trojan.Agent.AFHZ	Trojan.Agent.AFHZ								
ClamAV	Trojan.VB-1335	Trojan.VB-1335	Trojan.Dropper-3054			Trojan.Dropper-4506				
DrWeb										Trojan.Microjo.32
eSafe	suspicious Trojan/Worm	suspicious Trojan/Worm	suspicious Trojan/Worm	suspicious Trojan/Worm	suspicious Trojan/Worm	suspicious Trojan/Worm	suspicious Trojan/Worm	suspicious Trojan/Worm	suspicious Trojan/Worm	suspicious Trojan/Worm
eTrust-Vet										
Ewido	Logger.VB.sy	Logger.VB.sy	Dropper.VB.sg	Dropper.VB.sg	Downloader.VB.buy	Dropper.VB.sg		Dropper.VB.sg	Dropper.VB.sg	Trojan.VB.bkh
F-Prot	W32/Trojan.BXRR	W32/Trojan.BXRR								
F-Secure	W32/VBTroj.HGM	W32/VBTroj.HGM	W32/VBTroj.HXZ	W32/VBTroj.IBZ	W32/DLoader.EHYZ	W32/VBTroj.HXY	W32/Smalltroj.BVJK	W32/VBTroj.INO	W32/VBTroj.HYT	W32/VBTroj.KMQ
Ikarus	Trojan-Spy.Win32.VB.sy	Trojan-Spy.Win32.VB.sy	Trojan.Win32.VB.azd	Trojan.Win32.VB.azd	Trojan.Win32.VB.ayo	Trojan.Win32.VB.azd	Trojan.Win32.VB.ayo	Trojan.Win32.VB.azd	Trojan-Dropper.Win32.VB.sgo	Trojan.Win32.VB.ay
Kaspersky	Trojan-Spy.Win32.VB.sy	Trojan-Spy.Win32.VB.sy	Trojan-Dropper.Win32.VB.sg	Trojan-Dropper.Win32.VB.sg	Trojan-Downloader.Win32.VB.buy	Trojan-Dropper.Win32.VB.sg		Trojan-Dropper.Win32.VB.sg	Trojan-Dropper.Win32.VB.sgh	Trojan.Win32.VB.bk
McAfee	W32/Hilln.worm	W32/Hilln.worm								
NOD32v2	Win32/Spy.VB	Win32/Spy.VB	Win32/TrojanDropper.VB			Win32/TrojanDropper.V		Win32/TrojanDropper.VB		
Norman	W32/VBTroj.HGM	W32/VBTroj.HGM	W32/VBTroj.HXZ	W32/VBTroj.IBZ	W32/DLoader.EHYZ	W32/VBTroj.HXY	W32/Smalltroj.BVJK	W32/VBTroj.INO	W32/VBTroj.HYT	W32/VBTroj.KMQ
Panda	Suspicious file	Suspicious file	Suspicious file	Suspicious file	Trj/Downloader.SIE	Suspicious file	Suspicious file	Suspicious file	Trj/Dropper.YS	Suspicious file
Prevx1	TROJAN.AGENT.GEN	TROJAN.AGENT.GEN								
Rising			Dropper.Win32.VB.sg							
Sophos	Sus/Behav-169	Sus/Behav-169	Sus/Behav-169	Sus/Behav-169	Sus/Behav-169	Sus/Behav-169	Mal/Behav-096	Sus/Behav-169	Sus/Behav-169	Sus/Behav-169
Sunbelt CounterSpy										
Symantec-Norton	Trojan Horse	Trojan Horse	Downloader	Trojan Horse		Trojan Horse			Trojan Horse	
TheHacker	Trojan/Spy.VB.sy	Trojan/Spy.VB.sy	Trojan/Dropper.VB.sg	Trojan/Dropper.VB.sg		Trojan/Dropper.VB.sg		Trojan/Dropper.VB.sg	Trojan/Dropper.VB.sg	Trojan/VB.bkh
TrendMicro										
VBA32	Trojan-Spy.Win32.VB.sy	Trojan-Spy.Win32.VB.sy	Trojan-Dropper.Win32.VB.sg	Trojan-Dropper.Win32.VB.sg	Trojan-Downloader.Win32.VB.buy	Trojan-Dropper.Win32.VB.sg		Trojan-Dropper.Win32.VB.sg	Trojan-Dropper.Win32.VB.sgh	Trojan.Win32.VB.bk
VirusBuster										

IV. CONCLUSIONES

Cuando se habla de una prueba de vulnerabilidad en una compañía, lo primero en que se piensa es en un PenTest para verificar los controles de seguridad perimetral, vpn y red wireless, lo expuesto en este artículo evidencia la necesidad de incluir entre las pruebas de vulnerabilidad la plataforma antivirus, dado que hoy en día no solo con tener actualizada versión y firmas de detección se está cien por ciento asegurado, los software antivirus deben estar siendo probados periódicamente bajo un ambiente de pruebas detectando proactivamente recientes amenazas.

Los controles adicionales que antes se vendían como un valor agregado de los antivirus, hoy por hoy son una necesidad de primer nivel dado que la detección solo por firmas es lo que más se busca vulnerar y lo están logrando con gran certeza, podríamos decir que aplicaciones antivirus que no tengan incorporados patrones de detección heurísticos y comprobaciones de procesos de memoria serían poco útiles u obsoletos, dado que como control preventivo las firmas de detección pueden servir pero cuando estas fallan o son vulneradas deben existir controles reactivos que mitigue el riesgo como los mencionados anteriormente.

En este artículo solo evaluamos 10 distintas formas de cifrado de virus, pero en algún rincón de la tierra debe existir gente trabajando para desarrollar más y más algoritmos de cifrado, podríamos decir que esta es otra forma de argumentar la necesidad de enfocar esfuerzos y trabajar arduamente en patrones de detección heurísticos sin dejar de recordar que el riesgo siempre estará presente, lo que se busca al implementar este tipo de controles es mitigar la probabilidad de materialización de la amenaza y por ende el impacto generado.

Las compañías y en especial sus áreas de seguridad informática debe ser exigentes cada vez más con la calidad del servicio que presta el antivirus, dado que de poco sirve que se tenga una infraestructura robusta de seguridad perimetral y controles internos adicionales, cuando al interior de la compañía un funcionario con no buenas

intenciones y poco sentido de pertenencia podría explotar esta vulnerabilidad en solo tres pasos al ver que su antivirus es vulnerable, la confidencialidad, integridad y disponibilidad de la información y servicios se verían en grave riesgo, ya que con solo enviar un correo que lleve consigo un archivo adjunto contaminado y enmascarando en una presentación corporativa al administrador del dominio, red, etc, a la vuelta de unos minutos tendría el control total de su objetivo.

Por todo lo ya mencionado la plataforma antivirus debe ser un activo más a evaluar en las pruebas de vulnerabilidad.

REFERENCIAS

- [1] "Pruebas de Vulnerabilidad". Disponible en: <http://blownx.com/index.php/seguridad-informatica/44-seguridad-informatica/72-pruebas-de-vulnerabilidad>
- [2] Madantrax. "Cactus Methamorph". Disponible en: <http://www.elhacker.net>
- [3] BreakPoint Software. "Hex Workshop". Disponible en: <http://www.bpssoft.com>.
- [4] "The Anti-Virus or Anti-Malware Test File". Disponible en: http://www.eicar.org/anti_virus_test_file.htm
- [5] "Trece antivirus a examen". Disponible en: <http://www.terra.es/tecnologia/articulo/html/tec6237.htm>
- [6] Shapeless. "Poison Ivy". Disponible en: <http://chasetnet.org>
- [7] Nhaalckiemr. "Crypter". Disponible en: <http://www.elhacker.net>
- [8] "Heurística en antivirus". Disponible en: [http://es.wikipedia.org/wiki/Heur%C3%ADstica_\(antivirus\)](http://es.wikipedia.org/wiki/Heur%C3%ADstica_(antivirus))

Autor

Luis Fernando González Vargas.
Ingeniero de Sistemas.
Certified Ethical Hacker (CEH), Ec-Council.
5 años de experiencia en temas de seguridad Informática.
Líder de Riesgos
IQ Outsourcing S.A.

El orden de los *bits* sí altera el producto,
Talón de Aquiles de los Antivirus.

Nombre del proponente:
Luis Fernando González Vargas.

Filiación (empresa):
iQ Outsourcing S.A

Correo electrónico:
lfgonzalez@iq-online.com
luisfernando.gonzalez@gmail.com

Teléfono:
310-3026014
300-3137007

Dirección física:
Carrera 13ª No 29-24 piso 6
Bogota-Colombia