UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

METODOLOGÍA PARA EL ANÁLISIS DE COMPORTAMIENTO DE CÓDIGOS MALICIOSOS

TESINA

QUE PARA OPTAR POR EL GRADO DE:

INGENIERO EN COMPUTACIÓN

REDES Y SEGURIDAD

P R E S E N T A:

GARCÍA VIZCAÍNO JULIO CÉSAR

TUTOR:

Ing. AQUINO LUNA RUBÉN

Ciudad Universitaria **Febrero 2009**

Jurado asignado:

Presidente:	M.C. Jaquelina López Barrientos.
Secretario:	Ing. Rubén Aquino Luna.
1^{er} Vocal:	M.C. Alejandro Velázquez Mena.
1^{er} Suplente:	M.C. Cintia Quezada Reyes.
2^{o} Suplente:	Ing. Rafael Sandoval Vázquez.

Lugar donde se realizó la tesis:

Facultad de Ingeniería, de la Universidad Nacional Autónoma de México (UNAM).
Febrero de 2009.

Tutor de tesis:

Ing. Rubén Aquino Luna

Redacción y edición de tesina

con LaTeX 2ε, *GNU-Vim*

y sistema operativo libre Debian

GNU/LINUX.

A ti:

mi familia

♡

Quiso volar, igual que las gaviotas,
libre en el aire, por el aire libre.
Y los demás dijeron: pobre idiota,
no sabe que volar, es imposible

Alberto Cortez

Eetam eepitás

Anónimo

Walk on through the wind,
Walk on through the rain,
Tho' your dreams be tossed and blown.
Walk on, walk on with hope in your heart
And you'll never walk alone,
You'll never walk alone!

Rodgers and Hammerstein

Las máquinas y las computadoras deberían volverse una parte funcional en un sistema social orientado por la vida y no un cáncer que empieza por hacer estragos y acaba por matar al sistema.

Erich Fromm

Índice general

Índice de figuras

Índice de tablas

Introducción

Objetivo: Diseñar una metodología para analizar el comportamiento (análisis dinámico) de códigos maliciosos.

Definición del problema: Muchas organizaciones que realizan análisis de códigos maliciosos dedican gran tiempo y recursos en obtener un reporte del comportamiento de códigos maliciosos, es por ello que surge la inquietud y necesidad de definir procedimientos que permitan realizar dicho análisis de forma oportuna y eficiente.

Actualmente en el mercado existe una variedad de herramientas para el análisis dinámico de códigos maliciosos así como también distintos enfoques para analizar el comportamiento de los códigos maliciosos. Sin embargo no siempre el uso de una herramienta proporciona toda la información de la actividad que un código malicioso genera, esto se debe a la complejidad del código malicioso y/o a las tcnicas utilizadas por el mismo para evitar su análisis.

En consecuencia se propone diseñar una metodología que permita a los analistas de códigos maliciosos obtener un reporte eficaz que les permita actuar de forma oportuna para detener la propagación de los códigos maliciosos.

Capítulo 1

Códigos maliciosos

Los códigos maliciosos o software malicioso son programas informáticos utilizados para causar daños económicos, daños a la estructura informática de una organización y/o individuo, delitos, problemas legales, sociales, etcétera. La diversidad de códigos maliciosos que existen actualmente hace necesario su análisis y erradicación ya que ellos están diseñados para afectar la confidencialidad, integridad y disponibilidad de la información almacenada y procesada en los equipos de cómputo, es por ello que se abordarán en este capítulo los principales tipos de códigos maliciosos, en orden alfabético.

El término código malicioso es también conocido como *malware* en inglés y una definición apropiada de código malicioso se encuentra en [Skoudis and Zeltser, 2003] .

Un código malicioso es un conjunto de instrucciones que se ejecutan en una computadora y hacen que el sistema haga lo que el atacante quiere que haga.

El código malicioso es realizado por personas con propósitos que los beneficien a ellos o a otras personas e incluso con la finalidad de dañar los sistemas que sean infectados por el código. Las personas que se dedican a diseñar, escribir y ejecutar códigos maliciosos se conocen como atacantes, son individuos cuyo fin es el dañar de forma lógica, física e incluso financiera y económica e incluso pueden realizar sabotajes, fraudes o mal uso de los sistemas de cómputo de las organizaciones y las personas.

Algunas de las actividades que pueden realizar los códigos maliciosos son:

- Borrar archivos de configuración importantes del disco duro.
- Infectar la computadora y usarla para esparcir código malicioso hacia otros equipos.
- Guardar un registro de la actividad del usuario y/o usuarios del equipo con fines maliciosos.
- Permitir al atacante ejecutar instrucciones, comandos y/o programas en el equipo.

- Obtener información sensible del equipo, la red y/o el usuario.
- Realizar actividades ilegales desde el equipo infectado.

El comportamiento de un código malicioso depende en gran medida por las capacidades y el objetivo del atacante. Una vez que el atacante lo instala con éxito en el equipo de cómputo o engaña al usuario, al sistema operativo y/o software instalado en el equipo, es posible que el atacante pueda tomar posesión del equipo y con ello se puede cambiar la forma de operar del mismo desde ese momento. El éxito y el auge que los códigos maliciosos tienen entre los atacantes residen en la posibilidad de obtener y/o dañar grandes cantidades de recursos informáticos y controlarlos sin necesidad de tener acceso físicamente a ellos, además de la capacidad de expandir sus códigos hacia otros equipos de cómputo.

El rango de operación de los códigos maliciosos va desde:

- Los sistemas operativos diseñados para equipos de escritorio y portátil [XiTi Monitor, 2008].
    - Microsoft Windows
    - Mac OS
    - Distribuciones GNU/Linux
    - Otros sistemas operativos
- Aplicaciones de escritorio.
- Aplicaciones basadas en web.
- Dispositivos de red.
- Dispositivos móviles.
    - Celulares.
    - PDAs.
    - Smartphones.
- Aplicaciones multiplataforma.

Los atacantes para escoger el objetivo a atacar evalúan varios factores entre los que se pueden mencionar:

- La organización y/o usuario a quien se quiere dirigir el ataque.
- El resultado final que se obtiene del ataque: reputación, fama, ganancia económica, sabotaje (político, económico, financiero, militar).
- Porcentaje de uso del sistema operativo, aplicación y/o dispositivo. El objetivo de ataque necesita ser lo bastante popular para que sea accesible a los atacantes.

- Documentación disponible de las aplicaciones, sistemas operativos, etcétera. Una buena documentación incluye descripciones de los servicios disponibles y las reglas para escribir códigos compatibles.
- Número de vulnerabilidades documentadas y no documentadas.

Estos factores, sin ningún orden de importancia, son claves para que los programadores de códigos maliciosos enfoquen sus baterías hacia un sistema de cómputo y/o aplicación determinados.

Como la tecnología ha evolucionado, los códigos maliciosos también lo han hecho. A lo largo de casi 3 décadas ha habido grandes cambios en la tecnología y los programadores de códigos maliciosos se han mantenido a la vanguardia también con las nuevas tecnologías. Mientras los primeros códigos de los 80's tenían como objetivo una variedad de sistemas operativos y redes, los de hoy tienen como objetivo principal el software basado en el sistema operativo Windows, los protocolos y las aplicaciones independientes de la plataforma de operación.

1.1. Descripción, evolución, e impacto de los tipos de códigos maliciosos

Existe una gran variedad de códigos maliciosos, cada uno de ellos con características particulares, por ejemplo algunos requieren de la acción del usuario para infectar un sistema, otros se propagan a sí mismos e incluso existen amenazas combinadas (la combinación de más de un código malicioso en uno solo). La necesidad de entender la estructura, comportamiento, evolución e impacto (pérdida económica, de información, de reputación, etcétera.) nos permite tener un conocimiento de la complejidad e importancia que tiene el combatir cada uno de los tipos de códigos maliciosos.

1.1.1. Bot

Un bot es un programa que reside en un equipo sin el consentimiento del usuario, con la intención de establecer una red con otros bots. Una vez establecida la red de bots llamada botnet, el dueño de la botnet pueda emitir órdenes que son obedecidas por cada bot. La similitud que hay entre una botnet con un ejército no se aleja demasiado de la realidad, ya que los bots se apoderan del equipo en el cual se encuentran y realizan las actividades asignadas de forma pronta y puntual; entre las actividades que se pueden ordenar a un bot se encuentran el esparcir distintos tipos de códigos maliciosos, realizar ataques hacia otros equipos, coordinar ataques hacia un objetivo, el envío de spam, etcétera.

Un bot es una puerta trasera que está específicamente diseñado para crear botnets.[Kaspersky Labs., 2008]

Una botnet es una red de computadoras constituida por máquinas infectadas con una puerta trasera (bot) maliciosa.[Kaspersky Labs., 2008]

La puerta trasera permite a los atacantes controlar a los equipos infectados de manera remota, esto puede incluir a un solo equipo, un subconjunto de equipos infectados e incluso toda la red de equipos infectados.

El potencial y peligrosidad de un bot radica en sus características, no es perceptible por el usuario (se presenta al usuario como una aplicación válida), está a la escucha de nuevas órdenes, es posible su actualización, posee el poder de cómputo del equipo donde reside, etcétera. Además de que con frecuencia se utilizan para obtener dinero de forma ilegal. Su potencial radica en la posibilidad de controlar grandes recursos de cómputo desde lugares físicos distantes, de forma anónima. Los equipos infectados con un bot pueden ser controlados de forma directa o indirecta, es decir, el atacante puede administrar el bot o el mismo bot puede ejecutar las órdenes dadas de forma automática. El usuario del equipo no se percata de la presencia del bot, es por ello que también se les conoce como equipos zombi.

El envío de span es uno de los usos más simples de una botnet pero también de los más redituables, ya que de acuerdo con la compañía de antivirus Kaspersky un spammer promedio puede obtener ganancias entre US$50,000.00 y US$100,000 al año.[Kaspersky Labs., 2008]

El segundo uso más recurrente de los bots es para dirigir ataques DDoS. La pérdida económica debido a este ataque es incalculable, además de que en ocasiones los mismos atacantes extorsionan a sus víctimas ya que piden cierta cantidad de dinero a cambio de detener el ataque. Por el otro lado muchas de las organizaciones afectadas prefieren inhabilitar la red que les causó el ataque en lugar de denunciar a las autoridades.

Otro uso es para realizar ataques a otros equipos pero con la característica que el ataque será identificado hacia la máquina infectada con el bot y no la del propio atacante, lo que se traduce en un anonimato difícil de rastrear.

Como ya se ha mencionado, los bots tienen la posibilidad de ejecutar acciones de acuerdo con las órdenes recibidas por el atacante y/o por algún servidor. Entre los comandos que se pueden implementar en un bot están:

- Update: descarga y ejecuta un archivo ejecutable o módulo diseñado desde un servidor específico.
- Flood: crea un flujo de peticiones falsas a un servidor específico de Internet.
- Spam: descarga una plantilla de un mensaje de spam y envía el spam a un conjunto de direcciones de correo electrónico ya asignado.
- Proxy: usa al equipo para conectarse a otros a través de él.

Los botnets se clasifican de acuerdo con su arquitectura:

- **Botnets centralizadas.** Todos los equipos están conectados a un solo centro de control y comando, éste espera por nuevos bots a conectarse, los registra en su base de datos, registra su estatus y envía los comandos seleccionados por el dueño de la botnet de una lista de comandos bot. El dueño de la red zombie requiere forzosamente del acceso al centro de control y comando para administrar la botnet centralizada.
- **Botnets P2P o descentralizadas.** Los bots se conectan a varias máquinas infectadas en una red botnet. Los comandos son transmitidos de bot en bot. Cada bot tiene un registro de sus vecinos de tal forma que al recibir un comando éste lo reenvía hacia sus vecinos y así sucesivamente. En este escenario el atacante necesita tener acceso a al menos una computadora zombi para poder controlar la red zombi.

Además se clasifican también de acuerdo con el protocolo de red utilizado:

- **IRC.** Cada equipo infectado se conecta a un servidor IRC indicado en el cuerpo del programa bot y se pone a la espera del comando a ejecutar del maestro en cierto canal.
- **Mensajería Instantánea.** Se comunican mediante el uso de servicios de mensajería instantánea, aunque este tipo de botnets no son tan comunes debido a que cada bot debe estar conectado a la red y mantenerse en línea todo el tiempo e incluso cada bot necesita su propia cuenta de mensajería instantánea.
- **Web.** Un bot se conecta a un servidor web predefinido, recibe los comandos de él y transfiere los datos al servidor en respuesta. Su popularidad radica en lo fácil de crear y administrar.
- **Otros.** Dentro de esta categoría se agrupan las botnets que se comunican con su propio protocolo basado en TCP/IP.

Las botnets comenzaron cuando las puertas traseras se utilizaron para que ellas mismas realizaran conexiones y que pudieran ser visibles en línea todo el tiempo. Estos primeros bots se comunicaban haciendo uso de IRC: se conectaba a un servidor de IRC en un canal predeterminado como visitantes y esperaban mensajes provenientes del dueño de la botnet a la cual pertenece el bot. El dueño podía estar en línea a cualquier hora, ver la lista de bots, enviar comandos a todos los equipos infectados o enviar un mensaje privado a un equipo infectado.

La siguiente fase evolutiva de las botnets fue trasladar los centros de control hacia lugares geográficos distantes, es decir, aprovechar las capacidades de Internet. Los primeros atacantes que utilizaban bots, desarrollaron sus herramientas de administración remota mediante lenguajes como Perl y PHP. Luego se desarrolló un método por el cual una computadora en una red local podía conectarse a un servidor en Internet. El gran salto surgió al cambiar el enfoque de operación de

las botnets, es decir, redefinir la forma de operar del centro de comandos. Ahora cada botnet es su propio centro de comandos, lo que se conoce como botnets P2P.

Ahora bien la razón del auge de las botnets se debe a que actualmente los atacantes no requieren de elevados conocimientos o grandes cantidades de dinero para tener acceso a una botnet. Actualmente el mercado negro del software provee todo lo necesario para implementar una botnet, desde software para crear una botnet hasta redes zombi listas para su uso además de servicios de hosting anónimo a bajos precios; el precio fijado por el mercado negro del software, desde los US$5 hasta US$1000[Kaspersky Labs., 2008] está determinado por la difusión de la bot, si es detectada por productos antivirus, los comandos soportados, soporta cifrado, etcétera.

Actualmente las botnets son una de las principales amenazas además de que representa una gran entrada económica para el mercado negro del software. Su peligrosidad radica en la creciente facilidad de implementación. Las botnets no solamente pueden ser utilizadas por los atacantes, los gobiernos pueden apoderarse de ellas por motivos políticos.

1.1.2. Caballo de Troya

Cuando un atacante esconde una puerta trasera dentro de otro programa se denomina a esa aplicación como caballo de Troya, su nombre se debe al parecido con la historia de la Íliada de Homero. Mas formalmente:

Un caballo de Troya es un programa que parece tener algún propósito útil o benigno, pero que en realidad enmascara alguna funcionalidad maliciosa escondida[Skoudis and Zeltser, 2003].

Las dos principales metas que persigue un caballo de Troya son:

- Engañar a un usuario o administrador del sistema para instalar el caballo de Troya. En este caso el caballo de Troya y el usuario se convierten en el vehículo de entrada al sistema.
- Mezclarse con programas ya instalados en el sistema. El caballo de Troya se camufla para aparentar que pertenece al sistema y en consecuencia los usuarios no detectan su presencia.

Una de las estrategias más simples de los caballos de Troya es usar un nombre de un programa benigno, es decir, se engaña al usuario de tal forma que no se dé cuenta de la presencia del caballo de Troya y ante una posible revisión del sistema el caballo de Troya parezca un programa útil del equipo. En algunas ocasiones los caballos de Troya toman su nombre incluyendo muchos espacios entre el nombre del programa y la extensión en un equipo También los atacantes escogen extensiones de programas o nombres de programas que parecerían ser inofensivos para el usuario del equipo. Por ejemplo, “soytroyano.txt.exe” puede engañar al usuario creyendo que el archivo es de texto en lugar de un ejecutable.

Los atacantes también combinan diversos códigos maliciosos para formar uno solo. Con frecuencia los códigos maliciosos se combinan con algún caballo de Troya con la intención de que el usuario crea que el programa es válido y lo ejecuta. Además, otro de los medios de distribución de un caballo de Troya son los sitios web o los correos electrónicos. En el portal `www.malware.unam.mx` se encuentran casos documentados referentes a este tipo de actividad, por ejemplo el 28 de agosto de 2008 se documenta un caso de *pharming* en donde se hace uso de un caballo de Troya que intenta hacerse pasar por una aplicación del SAT (Sistema de Administración Tributario). Los nombres que utiliza este caballo de Troya son CalcRFC.exe, CalsT58.exe, CalcImpSAT.exe.

La historia de los caballos de Troya se remonta al año de 1989, cuando se entregaba un disquete aludiendo ser una base de datos de información del SIDA por correo a investigadores del SIDA y a suscriptores de una revista de computación británica. Los disquetes contenían un caballo de Troya que hacía a las computadoras inutilizables y demandaba enviar US$378 a PC Cyborg Corporation a un apartado postal en Panamá[Koch, 2007].

1.1.3. Código malicioso para dispositivos móviles

Los códigos maliciosos han evolucionado junto con la tecnología y la aparición de dispositivos móviles tales como: teléfonos celulares, asistentes personales, minicomputadoras, reproductores de multimedia personales, entre otros, ha generado la posibilidad de crear códigos maliciosos específicos para dichos dispositivos. La mayoría de estos dispositivos tienen la posibilidad de comunicarse con otros mediante el uso de conexiones alámbricas a la computadora, vía infrarroja, bluetooth, Wi-Fi, comunicación celular e Internet. El gran auge y popularidad de estos dispositivos ha hecho que se vuelvan un blanco a atacar. La forma de crear algunos de los códigos maliciosos para estos dispositivos es haciendo uso de la habilidad de los mismos para procesar las instrucciones de los códigos maliciosos.

Entre las actividades que ocasionan un código malicioso para dispositivos móviles están:

- Abusar de los servicios proporcionados por el dispositivo móvil: la mayoría de las compañías telefónicas limitan el número de mensajes de texto que se pueden enviar y recibir. Por ejemplo si un atacante envía una cantidad de mensajes de texto superior a la permitida por el operador del servicio, el cliente podría ser obligado a pagar cuotas adicionales por el abuso del servicio. Un atacante igualmente podría infectar el dispositivo móvil de tal forma que le permitiera hacer uso de los servicios proporcionados a dicho dispositivo. Debido a que estos ataques se generan y desarrollan en el dispositivo móvil, el dueño del mismo será el responsable de cubrir todos los gastos generados debidos al código malicioso.

- Visitar sitios web maliciosos: con el abanico de medios de conexión habilitados en los dispositivos móviles, un atacante podría enviar mensajes de texto a dichos dispositivos, estos mensajes supuestamente enviados por una compañía legítima podrían tratar de convencer al

usuario de visitar un sitio malicioso aduciendo que hay un problema con la cuenta del usuario o que el usuario se suscribió a un servicio. Al visitar el sitio, el usuario podría ser atraído a ingresar información personal o descargar un archivo malicioso.

- Usar el dispositivo móvil en un ataque: los atacantes pueden hacer uso del dispositivo móvil para atacar a otros dispositivos. Con la ventaja evidente de esconder la identidad real del atacante, además de permitir incrementar el número de potenciales víctimas.

Los códigos maliciosos para dispositivos móviles podría tomar varias formas en los siguientes años. Conforme los dispositivos móviles adquieran más funcionalidades, sus características los podrán hacer vulnerables a códigos maliciosos diseñados específicamente para ellos. Por ejemplo los teléfonos inteligentes y asistentes personales desarrollan nuevas capacidades como el trabajar con lenguajes de scripting, podrían volverse más susceptibles a caballos de Troya, virus y gusanos.

Algunos dispositivos podrían ser capaces de procesar scripts que contienen códigos maliciosos que podrían cambiar la libreta de direcciones o los datos del usuario y potencialmente transferirlos a otros usuarios. También los asistentes personales pueden ser usados para enviar correos electrónicos infectados a las direcciones contenidas en la libreta de direcciones de la víctima, tal y como sucedió con el virus Melissa.

De acuerdo con [F-secure, 2007a] el gusano Beselo se propaga vía MMS y bluetooth. Usando ingeniería social, engaña al usuario para ejecutar un archivo de instalación de la aplicación SIS. La forma de operar de Beselo es que usa extensiones de archivo multimedia comunes. Esto lleva a creer al usuario que está descargando una imagen o archivo de sonido en lugar de una aplicación Symbian.

Muchas aplicaciones móviles son programables y otros programas podrían interactuar con ellas mediante APIs, permitiendo a un software malicioso ejecutar código en la aplicación, por ejemplo enviar mensajes a otros dispositivos haciendo uso de la aplicación almacenada en el dispositivo.

El código malicioso para los dispositivos móviles será capaz de traspasar la frontera de las comunicaciones habituales en equipos de cómputo e infectar otros dispositivos móviles haciendo uso de las redes telefónicas y celulares, que afectan a un mayor segmento de usuarios de manera más tangible. La forma más sencilla de alertar sobre el potencial de estos códigos maliciosos es respondiendo la pregunta: ¿Tienes un dispositivo móvil?

1.1.4. Gusano

La capacidad de infectar de forma consecutiva una red de computadoras la posee el gusano. Un gusano es un tipo especial de código malicioso ya que tiene por características la habilidad de propagarse por sí mismo por la red.

Un gusano es un pedazo de código auto replicable que es capaz de propagarse por sí mismo en la red, usualmente no requiere intervención humana para propagarse[Szor, 2005].

Una característica típica de los gusanos es que no necesitan infectar archivos sino que se propagan ellos mismos. Además varios gusanos pueden tener el control de los equipos de forma remota sin intervención del usuario, usualmente explotando una vulnerabilidad o un conjunto de vulnerabilidades.

Cada gusano tiene algunos componentes que son comunes a todos ellos, tales como los módulos para ubicar el objetivo y el propagador de infección; además de los módulos opcionales para la administración remota, la interfaz de actualización, el administrador del ciclo de vida y las rutinas de payload.

El módulo para ubicar el objetivo es necesario para los gusanos ya que requieren esparcirse en la red y encontrar nuevos equipos a infectar. Algunas de las técnicas que utiliza este módulo son:

- Buscar en el sistema direcciones de correo electrónico y simplemente enviar copias de ellos mismos a esas direcciones. Esto es conveniente para los atacantes porque las organizaciones típicamente necesitan permitir los mensajes de correo electrónico a través de sus firewalls, convirtiéndose en un punto de penetración sencillo para el gusano.
- Desarrollan técnicas para escanear la red en busca de nodos a nivel de IP e incluso verificando si dicho sistema es vulnerable.

El módulo propagador de infección es un componente muy importante del gusano ya que es ahí donde se define la forma en cómo se transmite el gusano hacia un nuevo nodo e infecta el sistema remoto. La mayoría de los gusanos asumen cierto tipo de sistema y envían un gusano compatible con tal sistema.

La interfaz de control remoto y actualización permite al atacante enviar mensajes de control a las copias de los gusanos, la interfaz de actualización es importante, ya que permite al atacante actualizar el código del gusano en un sistema ya comprometido. El interés del atacante es cambiar el comportamiento del gusano e incluso enviar nuevas estrategias de infección a todos los nodos que sean posibles.

El módulo de payload es un componente opcional en los gusanos. Un efecto colateral del uso de payloads en gusanos es la posibilidad de tener ataques accidentales de DoS como resultado de redes sobrecargadas. Los gusanos también pueden ser utilizados para comprometer sistemas como una súper computadora. Un caso particular de esto es W32/Opaserv[Perriot, 2002] que intentó romper una clave secreta DES[National Bureau of Standards, 1997] repartiendo el trabajo entre los nodos infectados.

Los gusanos pueden clasificarse de acuerdo con:

- Método de propagación
- Método de instalación
- Características propias del gusano, tales como la capacidad de tener otros códigos maliciosos en su estructura.

Una vez que el gusano se ha instalado en el equipo, el siguiente paso es buscar medios de propagación residentes en el equipo:

- Escaneo local de medios de propagación:

  Correo electrónico

  Vulnerabilidades

  Malas configuraciones

- Escaneo de la red interna donde reside el gusano.
- Uso de mensajería instantánea.
- Uso combinado de las estrategias mencionadas.

Los gusanos de mensajería instantánea se esparcen utilizando aplicaciones de mensajería instantánea enviando enlaces a sitios web infectados a toda la lista de contactos. La diferencia con los gusanos de correo electrónico es el medio por el cual envían los enlaces.

Los gusanos de Internet usan otras técnicas para su distribución, entre las cuales se pueden mencionar las siguientes:

- **Copiar el gusano a los recursos de red.** Los gusanos localizan computadoras remotas y se autocopian en directorios con permisos de lectura y escritura.
- **Explotar vulnerabilidades propias del sistema operativo y/o aplicaciones para penetrar a equipos y/o redes.** Estos gusanos escanean todos los recursos de red disponibles usando los servicios del sistema operativo y/o escanean la red en busca de equipos vulnerables. Intentarán conectarse a esos equipos y ganar acceso a ellos.
- **Penetrar redes públicas.** Escanean Internet en busca de equipos que no han sido actualizados, por ejemplo sistemas operativos con vulnerabilidades críticas aún abiertas para explotarse. El gusano envía paquetes de datos o peticiones que instalan ya sea el cuerpo entero del gusano o sólo el módulo para descargar el payload. Finalmente ejecuta su código y el ciclo continúa.

- **Usar otro código malicioso para que actúe como el medio de transporte del gusano.** Utilizan otros códigos maliciosos para su proliferación, la forma de viajar de estos gusanos es como un módulo extra del código malicioso usado como medio de transporte.
- Muchos gusanos utilizan más de un método de propagación siendo más eficientes al penetrar en potenciales equipos víctima.

El objetivo de los gusanos de IRC son los canales de chat propios de IRC e igualmente utilizan los métodos de propagación descritos anteriormente.

Los gusanos P2P se autocopian en un directorio compartido, usualmente en la máquina local. Una vez colocado el gusano, la red P2P informa a los demás usuarios acerca del nuevo recurso y provee la infraestructura necesaria para descargar y ejecutar el archivo infectado.

La importancia de detener la proliferación de los gusanos nos la da la misma historia de ellos, ya que han sido uno de los códigos maliciosos que más desastres han causado a la estructura de redes corporativas e incluso a Internet, ver tabla 1.1 en la página 13.

Nombre del gusano	Fecha	Objetivo	Características
Gusano Morris	Noviembre 1988	UNIX	Desactivó la mayor parte del backbone del inicio de Internet.
Melissa	Marzo 1999	Microsoft Outlook	Era un virus macro de Microsoft Word propagado por Outlook, actuando como un virus (infectando archivos .doc) y un gusano (esparcido por la red).
Love bug	Mayo 2000	Microsoft Outlook	Este gusano era un script de Visual Basic que se difundía por Outlook. Fue tal el impacto que algunas organizaciones se desconectaron de Internet por un par de días, esperando que pasara el gusano.
Ramen	Enero 2001	Linux	Usaba tres diferentes vulnerabilidades de buffer overflow.
Código rojo	Julio 2001	Servidor Windows IIS	Infectó 250,000 sistemas en menos de 9 horas. Desde sistemas de todo el mundo, planeó un ataque dirigido contra la dirección IP de www.whitehouse.gov.

Nimda	Septiembre 2001	Internet explorer, compartir archivos, servidor web IIS, Microsoft Outlook	Este gusano multiexploit incluyó aproximadamente 12 diferentes mecanismos de propagación. Liberado justo una semana después de los ataques terroristas del 11 de septiembre de 2001, fue uno de los más rápidos en expandirse.
Klez	Enero 2002	Microsoft Outlook y compartir archivos en Windows	Este gusano tenía un componente polimórfico con su aleatoriedad del asunto y el tipo de archivos adjuntos. Klez también intentó desactivar los productos antivirus.
Slapper	Septiembre 2002	Apache con OpenSSL en sistemas Linux	Este gusano se propagaba mediante una vulnerabilidad en el código SSL usado por los servidores web Apache. Al mismo tiempo que se propagaba construía una red distribuida punto a punto de negación de servicio, a la espera de un comando del atacante para lanzar una inundación masiva.
SQL Slammer	Enero 2003	Sistemas Windows ejecutando el servidor de base de datos SQL	Inició su propagación afectando la conectividad a Internet de Corea del Sur por varias horas y apagando miles de cajeros automáticos en Estados Unidos.
Bagle	Enero 2004	Sistemas Windows	El gusano envía mensajes con el asunto 'Hi' y adjuntos ejecutables con nombres aleatorios. El gusano instala una puerta trasera en los equipos infectados.
Bropia	Febrero 2005	Sistemas Windows	Es un gusano residente en memoria que se propaga el mismo por medio de MSN Messenger enviando una copia del gusano usando diferentes nombres de archivos para todos los contactos disponibles o en línea.
Warezov	Septiembre 2006	Sistemas Windows	El gusano llega como un correo electrónico que envía copias de si mismo como adjuntos de correo electrónico a direcciones encontradas en el equipo infectado.

Storm Worm	Enero 2007	Sistemas Windows ejecutando Microsoft Internet Information Services (IIS) no actualizado.	Ejecuta un ataque DoS a `http://www.microsoft.com`
Beselo	Enero 2008	SymbOS	Beselo se propaga por Bluetooth y MMS (Multimedia Message System) como archivos de instalación Symbian SIS.

Tabla 1.1: Historia de los gusanos más representativos[Skoudis and Zeltser, 2003].

1.1.5. Puerta trasera

Una puerta trasera o también conocido como *backdoor*, en inglés, es un mecanismo lógico que permite a un individuo el acceso a un sistema sin necesidad de autenticarse de la forma habitual o usando una autenticación que solamente dicho individuo conoce. La peligrosidad de este tipo de programas radica en el uso maligno que se les puede dar.

> *Una puerta trasera es un mecanismo que permite evitar la autenticación normal, asegurar el acceso remoto, obtener acceso a texto plano, etcétera, mientras intenta mantenerse sin ser detectado. La puerta trasera puede ser un programa instalado o puede ser la modificación de un programa existente o de un dispositivo de hardware.[Skoudis and Zeltser, 2003]*

Muy frecuentemente la gente confunde el término de puerta trasera con un caballo de Troya, sin embargo, esto no debe de ser así ya que hay diferencias sustanciales entre cada uno de ellos. La principal diferencia radica en que una puerta trasera únicamente permite el acceso al sistema y el caballo de Troya incluye además otras características.

Las puertas traseras son uno de los códigos maliciosos más difundidos, ya que algunas puertas traseras son utilidades de administración remota que dejan máquinas infectadas para su posterior control a través de la red local e incluso desde Internet. De manera similar a los programas de administración remota legales usados por administradores. Esto hace que su detección sea mucho más difícil.

Una vez que una puerta trasera es instalada en el equipo local sin el conocimiento del usuario, se pone a la espera de conexiones del atacante, además de que la puerta trasera no es visible en la bitácora de programas activos.

Las acciones que un atacante puede realizar al utilizar una puerta trasera son:

- Enviar y recibir archivos.
- Ejecutar y borrar archivos.
- Borrar datos.
- Reiniciar el equipo.

En resumen, una puerta trasera es usada por los atacantes para obtener información confidencial (del equipo y del usuario), ejecutar otros códigos maliciosos, destruir datos, agregar el equipo a redes bot.

La historia de las puertas traseras se remonta a finales de los 90s, cuando hicieron su aparición dos de las más famosas puertas traseras:

- NetBus
- BackOrifice2000

Ambas implementaban tecnologías innovadoras para su época. Incluían un conjunto completo de funciones que hacían posible administrar remotamente las computadoras infectadas; permitiendo a los atacantes realizar operaciones en los equipos remotos, ejecutar nuevos programas, obtener screenshots, abrir o cerrar la unidad de CD-ROM, etcétera.

Las primeras puertas traseras trabajaban en redes de área local basadas en la pila de protocolo TCP/IP y con ello lograron demostrar las posibilidades de explotar el API de Windows para controlar una máquina de forma remota.

Además estas primeras puertas traseras abrían un puerto predefinido y esperaban pasivamente por el maestro (equipo que activa a la computadora cliente) a que se conecte. La siguiente fase evolutiva de las puertas traseras fue desarrollar la capacidad de conectarse ellos mismos al equipo del atacante y que deberían estar siempre en línea (en la condición de que la máquina esté encendida y funcionando).

Existen diferentes modos de operación de las puertas traseras:

- Escalamiento de privilegio local: permite al atacante con una cuenta en el sistema cambiar su nivel de privilegio a root o administrador.
- Ejecución individual de comandos remotos: permite al atacante enviar un mensaje a la máquina objetivo para ejecutar en un simple comando, la puerta trasera ejecuta el comando del atacante y le envía la salida de dicho comando.
- Acceso remoto de línea de comando: permite al atacante teclear comandos desde el prompt de comandos de la víctima, con la consecuencia de que puede acceder a todos los recursos del equipo, es conocido como shell remoto.
- Control remoto de la GUI de la víctima.

Una puerta trasera utiliza distintas técnicas para su instalación en un equipo víctima y éstos varían de acuerdo con el tipo de sistema operativo.

En un equipo basado en Microsoft Windows XP y superiores, los lugares más comunes son[Skoudis and Zeltser, 2003]:

- Archivos y directorios de inicio

  C:\Documents and Settings\[usuario]\Start Menu\Programs\Startup

  C:\Windows\Win.ini, C:\Windows\System.ini, C:\Windows\Wininit.ini

- Llaves de registro

  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServicesOnce

  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices

  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx

  HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit

  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ShellServiceObjectDelayLoad

  HKLM\SOFTWARE\Policies\Microsoft\Windows\System\Scripts

  HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

  HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServicesOnce

  HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices

  HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

  HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

  HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnceEx

  HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

  HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\Run

  HKCU\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\Load

  HKCU\SOFTWARE\Policies\Microsoft\Windows\System\Scripts

  HKCR\Exefiles\Shell\Open\Command

- Programador de tareas

Un aspecto a resaltar de las puertas traseras es la gran utilidad que representan para los atacantes, ya que este tipo de código malicioso no sólo se encuentra como un código solitario, sino que en ocasiones acompaña a otros códigos tales como botnets, virus, caballos de Troya, etcétera.

Es difícil determinar las pérdidas económicas que representa un ataque hecho por una puerta trasera. A pesar de eso es posible ejemplificar el impacto que tiene su uso en un ataque real. Por ejemplo, un phishing scam podría dirigir al usuario a un sitio que ha sido comprometido con inyección de contenido, que instala una puerta trasera en la computadora de la víctima mediante una vulnerabilidad de seguridad de su explorador. Esta puerta trasera es entonces usada para cambiar el archivo de hosts[1] y permite un ataque de pharming. Los siguientes intentos de solicitud al sitio web legítimo serán dirigidos a sitios phishing, donde la información confidencial del usuario es comprometida usando un ataque de man-in-the-middle. Al mismo tiempo, otro

[1] Archivo que almacena información de dónde encontrar una computadora en la red.

software malicioso puede también ser instalado usando la puerta trasera. Participar en un ataque de negación de servicio distribuido cuando recibe la orden de hacerlo.

Con este ejemplo se pretende alertar al lector del potencial maléfico que tiene una puerta trasera.

1.1.6. Software espía

El software espía (o spyware en inglés) roba información confidencial así como hábitos de uso de software del usuario.

> *Spyware es un término general usado para describir software que realiza ciertos comportamientos tales como publicidad, recolectar información personal o cambiar la configuración del equipo donde reside el código malicioso, generalmente sin el consentimiento del usuario[Microsoft, 2006].*

Normalmente el software espía se asocia con la difusión de publicidad, sin embargo, no es la única actividad que realiza. La peligrosidad radica en la capacidad no sólo de mostrar publicidad sino también de:

- Redirigir las peticiones web hacia otras direcciones.
- Monitorear el tráfico generado al navegar en Internet.
- Grabar las teclas presionadas del teclado.

Todo esto tiene como consecuencias:

- Molestias del usuario.
- Degradación en el rendimiento del equipo.
- Robo de identidad, utilizar información personal del usuario para identificar a otra persona.

Normalmente el usuario es capaz de detectar cuando tiene un software espía instalado en su equipo. Ya que el software espía realiza las siguientes actividades:

- Múltiples ventanas anunciando productos de distintos tipos.
- El explorador de Internet es modificado de tal forma que muestra páginas web no pedidas por el usuario.
- Un cambio repentino y/o repetitivo de la página de inicio de Internet del usuario.
- Nuevas e inesperadas barras de herramientas.
- Teclas que no funcionan de forma adecuada.

- Mensajes de error aleatorios.
- Rendimiento bajo del equipo al abrir programas o guardar archivos.

La forma de infectar un equipo con software espía es utilizando distintas técnicas, tales como:

- Al navegar en algunos sitios web, el software espía se auto descarga e instala sin el consentimiento del usuario.
- Se utiliza la ingeniería social para hacer que el usuario de clic en algún enlace que active el software espía.
- El software espía se incluye en algún instalador de software. Ésta es una de las formas más comunes de propagación ya que es utilizado como un mecanismo de financiamiento de algunas compañías de software.
- Por medio de archivos adjuntos en correos electrónicos.
- Distintos programas:

  Barras de herramientas para el explorador de Internet o el escritorio.

  Juegos, rompecabezas, en general programas de entretenimiento gratuito.

  Protectores de pantalla.

  Programas bloqueadores de pop-up gratuitos.

  Archivos descargados de servicios de uso compartido de archivos.

El software espía engloba diferentes subtipos[Walker, 2005]:

- **Software soportado en anuncios, o Adware en inglés.** Es una variante que además de espiar la actividad del usuario muestra anuncios, algunos muestran anuncios basados en los hábitos (uso del equipo, aplicaciones utilizadas, sitios web visitados, etcétera.) recolectados del usuario.
- **Software para fisgonear, o Snoopware en inglés.** Permite observar los hábitos del usuario en nombre de otro usuario, usualmente alguien conocido.
- **Secuestradores de navegador, o Browser hijacker en inglés.** Es uno de los spyware más difíciles de eliminar, sobrescribe la página de inicio del navegador y redirige el tráfico del usuario hacia otros sitios con contenido no deseado por él.
- **Registrador de tecla, o Key logger en inglés.** Se ejecuta como un proceso secreto en la memoria del equipo y captura todo lo tecleado por el usuario, es guardado para posteriormente ser analizado por un tercero (normalmente un atacante).
- **Marcadores telefónicos, o dialers en inglés.** Son programas que utilizan el módem del equipo para realizar llamadas a líneas de pago y conectarse a sitios web.

1.1.7. Rootkit

Uno de los códigos maliciosos que tiene características interesantes es el rootkit. Este código tiene la capacidad de ganar acceso y esconderse en el sistema. De acuerdo con [Skoudis and Zeltser, 2003]:

> *Un rootkit es un caballo de Troya con puerta trasera que modifica el software existente del sistema operativo de tal forma que el atacante pueda mantener el acceso al mismo y esconderse en el equipo.*

Los rootkits se hacen acompañar de versiones modificadas de programas del sistema operativo que aparentan ser programas benignos, pero que en realidad esconden características útiles para un atacante, por ejemplo ocultar su presencia en el sistema operativo. Una de las razones del uso de los rootkits es que pueden esconder los códigos maliciosos en el sistema, a diferencia de otros códigos en donde utilizan técnicas como utilizar nombres de programas válidos o cambiar la extensión del software. En el caso del rootkit no es necesario ya que modifica el sistema de tal forma que no es visible para el usuario.

Varios rootkits ofrecen acceso al sistema mediante puertas traseras protegidas mediante contraseña, un shell remoto, u otras posibilidades de acceso de puerta trasera. El rootkit reemplaza aplicaciones del sistema por sus propias versiones las cuales proporcionan las mismas características que las originales, pero con el pequeño detalle de ocultar al usuario de la presencia de los códigos maliciosos. La mayoría de los rootkits permiten a un atacante acceder al sistema sin generar ningún registro en las bitácoras del sistema. Además los rootkits permiten al atacante esconder archivos, procesos y uso de la red en el sistema.

1.1.8. Virus

Un virus es el término que más asocia la gente con un código malicioso, además es el primer código que apareció. Ese factor ha sido la causa de que muchos usuarios asocien todos los tipos de códigos maliciosos con el concepto de virus. Es común escuchar la frase “tienes un virus” al referirse a la infección debido a un código maliciosos. Es aconsejable concienciar al usuario sobre los distintos tipos de códigos maliciosos, es decir, una mejor cultura de la seguridad informática. Se caracterizan por tener un tamaño pequeño, algunos sólo se limitan a reproducirse pero otros pueden producir daños que afectan a los sistemas. La razón de llamar virus a estos códigos maliciosos surge de la analogía que existe entre éstos y los virus biológicos. Mientras los virus biológicos son agentes externos que invaden células para alterar su información genética y reproducirse; los virus informáticos son programas y/o códigos capaces de infectar archivos y reproducirse una y otra vez cuando se accede a dichos archivos, por lo que dañan la información existente en la memoria o en algunos de los dispositivos de la computadora.

Una definición formal de virus informático la menciona [Skoudis and Zeltser, 2003]:

Un virus es un código auto replicable que se adjunta él mismo a otros programas y usualmente requiere interacción para propagarse.

Una característica interesante de los virus es su incapacidad de funcionar como un solo ejecutable. Esto se debe a que requieren de un agente externo (programa, archivo, etcétera.) para su propagación. Tienen diferentes finalidades:

- Infectar otros archivos o sistemas.
- Alterar datos.
- Eliminar datos.
- Mostrar mensajes.

Pero su fin último es propagarse. Es necesario mencionar que el daño potencial ocasionado por un virus no depende de su complejidad sino del entorno donde actúa.

Las características de un virus son:

- Es muy pequeño.
- Ejecutable o potencialmente ejecutable.
- Se reproduce a sí mismo.
- Toma el control o modifica otros programas.
- Convierte otros objetos ejecutables en clónicos víricos, se vuelven portadores del virus.

El comienzo de los virus se remonta al lejano año de 1962 en donde los investigadores de los laboratorios Bell, Victor Vyssotsky, Douglas McIlroy y Robert Morris; llegaron con un juego de computadora llamado Darwin. Cuyo objetivo era escribir programas que pelearán por la dominación de una región de memoria designada. El objetivo del juego era sobrevivir, los programas tenían la habilidad de matar a otros y podían crear copias de ellos mismos. La primera implementación confirmada de un código auto replicable fue PERVADE [Skoudis and Zeltser, 2003]. PERVADE fue escrito por John Walker en 1975, consistía en una rutina de propósito general que podía ser llamada por cualquier programa que requiriera capacidades de propagación. El único programa conocido que almacenaba a PERVADE era ANIMAL (un juego popular en el que la computadora trata de adivinar que animal tiene el jugador en mente). El programa tenía propiedades virales ya que permitía propagarse de directorio a directorio e incluso cuando los usuarios compartían el juego, se propagaba a otros sistemas. En 1982 los equipos Apple II comenzaron a verse afectados por un virus llamado Elk Cloner que presentaba un mensaje en forma de poema, ver tabla 1.2 en la página siguiente.

Nombre del virus	Fecha	Características
Darwin	1962	En este juego de computadora, los programas pelean por sobrevivir matando a los demás y replicándose en memoria.
PERVADE	1975	Esta rutina, adjunto a un juego llamado ANIMAL, permitía al programa propagar copias de sí mismo a través de todo el sistema.
Elk Cloner	1982	Virus desarrollados para los equipos Apple II.
Core War	1984	Esta es una versión de Darwin que formalizó y popularizó las reglas y objetivos del juego.
Brain	1986	Fue el primer virus conocido para computadoras con MS-DOS, se propagaba haciendo uso del sector boot (arranque) de los disquetes.
Virdem	1986	Uno de los primeros virus para computadoras MS-DOS, este espécimen se propagaba él mismo a archivos COM.

Tabla 1.2: Primeros virus en la historia. [Skoudis and Zeltser, 2003]

Un virus necesita adjuntarse él mismo a un programa para funcionar. El objetivo potencial es cualquier archivo que puede contener instrucciones ejecutables, tales como un ejecutable estándar, un sector de arranque de un disco o un documento que soporte macros.

Al infectar archivos ejecutables el virus se asegura que será activado cuando el usuario ejecute el programa infectado. La mayoría de sistemas operativos tiene varios tipos de ejecutables, mediante los cuales un virus puede aprovecharse para infectar al equipo.

- En sistemas UNIX:

  Archivos binarios.

  Variedad de tipos de script.

- En sistemas Windows:

  Archivos COM.

  Archivos EXE.

- En el kernel del sistema operativo.

La forma de infectar esos tipos de archivos es haciendo uso de las siguientes técnicas:

- **De acompañamiento.** El virus se nombra de tal forma que sea ejecutado por el sistema operativo antes que el verdadero archivo. Esto es, al intentar ejecutar un archivo si no se le da la extensión, el primero en ejecutarse será en el caso de Windows aquellos con extensiones COM en lugar de un EXE. En ocasiones el virus ejecuta el archivo auténtico para que el

usuario no note la ejecución del mismo. Otra variante es poner el virus con el mismo nombre que el programa benigno y colocándolo antes en la ruta de ejecución que el programa benigno.

- **De sobre escritura.** El virus infecta un ejecutable reemplazando porciones de código del ejecutable.
- **Infección al inicio.** El virus se inserta al inicio del archivo que infecta, permitiendo que el virus sea ejecutado primero al llamar al archivo infectado.
- **Infección al final.** El virus se inserta al final del archivo que infecta, el virus necesita crear un salto en el archivo infectado para garantizar su ejecución.
- **Infección en sectores de arranque.** El virus se ejecuta cuando el equipo se enciende. Un ejemplo es el virus Miguel Ángel de 1991, cuya función era ejecutarse en el sector de arranque en la fecha de nacimiento de Miguel Ángel, 6 de marzo, una vez infectado el virus infectaba a cada disquete insertado en el sistema.
- **Infección de documentos.** Los documentos susceptibles a este tipo de infección son aquellos con capacidad de ejecutar scripts o programas incluidos en el documento, por ejemplo los macros de Microsoft Office.

Hasta el momento se ha mencionado la forma de infectar de un virus, sin embargo el virus requiere de algún medio que le permita continuar infectando. Es por ello que los virus tienen algunos mecanismos de propagación:

- Almacenamiento removible: en esta categoría están aquellos virus que hacen uso de cualquier medio de almacenamiento removible tal como memoria FLASH, CD, DVD, discos duros externos, etcétera.
- Correo electrónico y descargas de archivos: se utilizan los adjuntos del correo electrónico o se solicita al usuario descargar el virus, engañándolo primero, de algún sitio de Internet.
- Directorios compartidos y redes P2P: los virus se aprovechan del hecho que un archivo es accesible por más de un usuario.

La idea que el software puede propagarse haciendo copias de sí mismo y adjuntarse a programas benignos es poderosa. Al ser el virus el primer código malicioso de aparición en la historia de la computación, esa característica fue la base para que los atacantes hagan uso de dichas técnicas para construir nuevos códigos maliciosos como una combinación de varios de ellos.

En la siguiente tabla se muestra una semblanza de los diferentes códigos maliciosos a lo largo de la historia, tabla 1.3 en la página 28.

Año	Códigos maliciosos
1986	Virus Brain, desarrollado por el programador pakistaní Basit Farrq Alvi y sus hermanos Shahid y Amjad, era capaz de infectar la zona de arranque, cambiar el nombre del disco a "(c) Brain" y fue el primero en utilizar técnicas tipo Stealth (esconder modificaciones realizadas a los archivos o al sector de arranque) para ocultar su presencia.
1987	Ralph Burger crea el virus Vienna. Vienna era un virus extremadamente sencillo, no residente en memoria, capaz de infectar sólo archivos .COM sobre DOS 2.0 o superior y de modificar los segundos de la hora del sistema al valor "62". También aparece Lehigh (aparentemente experimentos de Cohen y Ken van Wyk) en la universidad que le dio nombre (EE.UU.), siendo uno de los primeros virus dañinos. Aparición del primer virus capaz de infectar Macintosh: MacMag. Un empleado de Aldus Corporation se infecta con un disco de juegos y luego el virus es distribuido al público en copias de su software Aldus Frenhand 10. A finales de ese año hace su aparición el virus Jerusalem o Viernes 13 (modificaciones de su antecesor suriV) y capaz de infectar archivos .EXE y .COM. Su primera aparición fue reportada en la Universidad Hebrea de Jerusalem y llegó a ser uno de los más famosos de la historia. En Alemania, el investigador Wolfang Stiller reporta la aparición del complejo Cascade o Falling Letters capaz de infectar archivos .COM y el primer virus encriptado en conocerse.
1988	Aparece Stoned, también conocido como Marijuana debido a su popular mensaje "LEGALISE MARIJUANA". Se cree que su origen es Nueva Zelanda, ya que desde allí se reportaron los primeros casos. En marzo es encontrado en la Universidad de Turín el virus Ping Pong haciendo honor al juego de la pelotita que rebotaba en pantalla e infectando la zona de arranque del disco. El 2 de noviembre de 1988, Robert Tappan Morris (hijo de Robert Thomas), estudiante de 23 años del MIT (Instituto Tecnológico de Massachusetts), crea el primer gusano de reproducción masiva, infectando y colapsando el 10 % de ARPANET (incluyendo la NASA y el MIT) durante 72 horas.
1989	En Bulgaria nace el virus 512 (The number of the beast), el cual infectaba archivos .COM y tenía capacidades stealth.

	Surgen los virus de la "fábrica búlgara de virus" algunos de los virus mas reconocidos son: Dark Avenger, Dir, Dir II, Int13, Murphy, Nomenclatura, Darth Vader y Vacsina; todos ellos con técnicas de infección exclusivas y no explotadas hasta ese momento. También aparece el peligroso Datacrime que formateaba a bajo nivel el cilindro cero (donde se aloja la FAT) del disco y que sólo actuaba desde el 13 de octubre hasta el 31 de diciembre.
1990	La revista inglesa PC Today distribuye, por error, el virus DiskKiller en una de sus publicaciones y el mismo se trasforma en epidemia. En la segunda mitad del año aparecen Frodo y Whale, que usaban un complejo algoritmo para camuflarse en el sistema. Mark Washburn crea Chameleon, el cual era capaz de mutar con cada infección (polimórfico).
1991	Se crea CARO (Computer Antivirus Research Organization) para combatir a los virus. Además CARO decide que los virus deben bautizarse de la siguiente manera (según "A New Virus Naming Convention"): Family_Name.Group_Name.Major_Variant.Minor_Variant[Modifier]. Nomenclatura que se sigue respetando (Prefijo.Nombre.Variante).
1992	El 6 de marzo, la aparición de Michelangelo (virus de arranque, variante de Stoned) hace que este año sea un hito en la historia, ya que los virus informáticos son masivamente expuestos a la opinión pública. Por su parte, las capacidades de Dark Avenger seguían creciendo y este virus dio origen a varios motores automáticos de creación de virus. El primero de éstos fue el MtE (Self Mutating Engine) creado por el mismo Dark Avenger y con manual de uso incluido. Otros constructores dignos de ser mencionados son VCL y PS-MPC. Aparece Peach, primer virus capaz de "atacar" la base de datos de un antivirus y EXEBug capaz de controlar la CMOS para prevenir el arranque desde disquetes limpios. También se descubre Win.Vir_1_4, el primer virus capaz de infectar ejecutables de Windows 3.x.
1996	En febrero de este año en Australia es detectado Boza (o Bizatch según sus creadores del grupo VLAD), el primer virus capaz de infectar archivos de 32 bits de Windows NT y del recién estrenado Windows 95. Es hallado virus Zhengxi un complejo virus para Windows 95, escrito por el ruso Denis Petrovym, Zhengxi es un virus polimórfico, residente en memoria, infector de archivos EXE, LIB y OBJ, stealth y capaz de insertar droppers (archivo ejecutable que contiene otros archivos en su interior) en formato COM en los ficheros ZIP, ARJ, RAR, HA, y en EXE self-extracting.

	En junio aparece el virus AEP, el primer virus capaz de infectar archivos ejecutables de OS/2 EXE. En julio, se descubre en Alaska y África, Laroux, el primer virus capaz de infectar macros en archivos de Microsoft Excel.
1997	En febrero aparecen Staog (escrito en ensamblador por el grupo Quantum/VLAD) y Bliss, los primeros virus para archivos ELF del emergente sistema operativo Linux. Aparece ShareFun, el primer macrovirus con capacidades de enviar documentos infectados por correo electrónico a través de MSMail. Una nueva era había comenzado. Aparece Homer, un gusano que se propagaba por FTP; y a Esperanto (del grupo 29A), un intento de virus multiplataforma que podría infectar DOS, Windows y MacOs.
1998	Sigue el auge de los virus de macros tales como Cross, el primero en infectar dos aplicaciones de la familia Office: Word y Access; y Triplicate (o Tristate) capaz de infectar Word, Excel y PowerPoint. Aparece Marburg un virus polimórfico, creado por GriYo del grupo 29A, capaz de infectar ejecutables de Win32 y distribuido en los CD de algunas revistas europeas. Aparecen BackOrifice (del mítico grupo Culto de la Vaca Muerta o cDc por sus siglas en inglés de "the Cult of the Dead Cow"), NetBus, Phase y D.I.R.T.; caballos de Troya. Lo más destacable de este año sin duda lo logra el virus taiwanés CIH (iniciales de su autor, el estudiante Chen Ing-Hou) o Chernovyl detectado en junio y activado 26 de abril de año siguiente (aniversario del accidente en Chernobyl) o el 26 de cada mes, según la versión.
1999	Surge el caballo de Troya Happy (conocido como Ska en mención a su autor, el francés Spanska), estrenando una nueva moda que persiste hasta ahora: los gusanos para MS Outlook. Happy se caracteriza por su mensaje "Happy New Year 1999 !!" y sus fuegos artificiales. Debido a su capacidad de modificar ciertos archivos del sistema es capaz de enviarse a sí mismo a cada persona a quien el usuario envía un correo. El 26 marzo Melissa (en memoria a una bailarina exótica) comenzó a llegar a miles de correos en un archivo adjunto y enviado por alguien conocido. Las capacidades de Melissa y la confianza del usuario en quien le enviaba el correo, hizo que este macrovirus se convirtiera en una epidemia rápidamente y causara grandes pérdidas económicas.

	En junio aparece ZippedFiles: un gusano que llega por correo en formato .EXE y con capacidades de replicación en recursos compartidos; y dos virus conceptuales para Windows NT: Remote Explorer desarrollado como un servicio en modo usuario e Infis primero en ejecutarse en modo Kernel. El virus Parvo, otro polimórfico de GriYo, capaz de infectar Windows 95, 98 y NT. Implementa su propia rutina SMTP, desarrollada en Assembler. Es capaz de enviarse a múltiples direcciones de correo con sólo estar conectado a Internet. Fue uno de los primeros virus en utilizar direcciones de origen del correo falsas (spoof) para facilitar su propagación, lo que lo hace similar a los gusanos actuales.
2000	El virus Anna Kournikova (detectado como SteeLee y que llegó a infectar a la NASA), generado por el joven holandés OnTheFly, en agosto. El "gusano del amor": LoveLetter en Manila, Filipinas. Llegaba por correo con un adjunto y su nombre se debe a que en uno de los asunto del mensaje era "ILOVEYOU". Los daños ocasionados por este gusano se calcularon en millones de máquinas infectadas y millones de dólares en pérdidas.
2001	Aparece Pirus, el primer virus desarrollado en el lenguaje PHP y que sólo se ejecuta en servidores Web. En enero nace Ramen y en marzo Lion, gusanos para el sistema operativo Linux, que aprovechan diversas vulnerabilidades en RPC, wuftpd y BIND. Aparece el gusano polimórfico Magistr utilizando rutinas de envío SMTP propias evitando así la utilización de clientes de correo. En abril aparece el peligroso gusano BadTrans capaz de propagarse a través del correo utilizando el Microsoft Outlook. Permite el robo de información confidencial, y su mala programación hace caer a los servidores de correo. En julio aparece CodeRed que se propaga buscando servidores con IIS 5.0 (Internet Information Server) vulnerables. Cuando encuentra un servidor, el gusano intenta ingresar al sistema a través del puerto 80, explotando una vulnerabilidad. En julio, el caballo de Troya SirCam, escrito en México en el lenguaje Borland Delphi, es capaz de enviarse a sí mismo a todos los usuarios de la libreta de direcciones de Windows y a direcciones encontradas en los archivos temporales de Internet, además de aprovecharse de los recursos compartidos y de contener una peligrosa rutina de destrucción. En septiembre aparece el caballo de Troya Nimda (admin., de administrador, invertido) que se propaga por correo al visualizar páginas web, a través de recursos compartidos y atacando servidores web (ISS de Microsoft).

	Aparece el virus polimórfico Elkern que es propagado por el gusano Klez (explotando las mismas vulnerabilidades que Nimda).
2002	El descubrimiento de Frethem y Bugbear (o tanatos) marcan la aparición de malware empaquetados para evitar su detección por parte de los antivirus. En estos casos el empaquetador utilizado era UPX (Ultimate Packer for eXecutables) aunque actualmente existen cientos de tipos. En cuanto a Linux aparece Slapper un gusano que intenta aprovecharse de la vulnerabilidad de desbordamiento de buffer en el componente OpenSSL en servidores Apache.
2003	El gusano Slammer utilizando una vulnerabilidad del servidor Microsoft SQL infectó menos computadoras que CodeRed, pero actuó dos veces más rápido infectando más del 90 % de las computadoras vulnerables tan sólo 10 minutos después de iniciar su propagación [CAIDA, 2008]. El gusano Blaster, que apareció en agosto aprovechando vulnerabilidades en Remote Procedure Call (RPC) de Windows para reproducirse. Comienzan a conocerse y a utilizarse las botnets utilizando Agobot (o Gaobot o Morphine o Phatbot o Forbot o XtremBot), RBot (o SDBot o UrBot o UrXBot) y Mydoom/Mytob.
2004	En enero aparece el destructivo Mydoom, un gusano que se propaga por correo electrónico y la red de intercambio de archivos Kazaa, permitiendo el control remoto del equipo infectado. Nace una nueva epidemia: Bagle (o Beagle), demostrando ser el virus más persistente e "inteligente" desde la existencia de Internet. Se crea Sasser, buscando sistemas Microsoft Windows 2000, 2003 y XP que aún no hayan parcheado una vulnerabilidad en el proceso LSASS (Local Security Authority Subsystem)
2005	Existe un fuerte incremento del uso de puertas traseras, rootkits y caballos de Troya [Kaspersky Labs., 2008]. Se hace uso de los caballos de Troya para el robo de información financiera almacenada en el equipo infectado. Continúan apareciendo variantes Bagle, Sober y Netsky. Hasta finales de año Zafi.D, y Sober.X se mantuvieron como las amenazas más latentes del 2005. Aparece el primer código malicioso para dispositivos móviles: Worm.SymbOS.Comwar.a que se propagaba mediante MMS, no siendo el único ya que también aparecieron códigos maliciosos para consolas de juego (PSP). Finalmente en diciembre de ese año se descubre el exploit WMF [Mary Landesman, 2006].

2006	El gusano Sober.X es utilizado para descargar nuevos códigos maliciosos. Surge además el gusano Nyxem también conocido como el gusano Kama Sutra debido a los provocativos asuntos que utilizaba al propagarse mediante correo electrónico. En febrero se hace público la existencia de un gusano para MacOS X. En general se puede mencionar que los gusanos Nyxem, Bagle y Warezov junto con las variantes Gpcode (un caballo de Troya que cifra los datos del usuario) fueron los principales códigos maliciosos de ese año [Kaspersky Labs., 2008].
2007	Este año sin duda alguna es posible bautizarlo como el año de Storm Worm. Este código malicioso apareció en enero y durante todo el año demostró un amplio rango de diferentes comportamientos: interacción entre sus variados componentes, métodos de propagación y tácticas de ingeniería social. Además incluía tecnologías de rootkit, código basura y botnets capaces de auto-protegerse contra el análisis y el estudio, interacción entre los equipos infectados mediante redes P2P sin ningún centro de control. El gusano utilizó todos los métodos de propagación existentes: sistemas de mensajería instantánea, correo electrónico, redes sociales, blogs, foros y fuentes de noticias (RSS) [Kaspersky Labs., 2008]. Casi todos los códigos maliciosos que aparecieron fueron de corta duración y limitados en regiones y países específicos.
2008	Durante la primera mitad del año se caracteriza por la proliferación de Storm Worm, continúa el auge del uso de los virus como amenazas combinadas de los códigos maliciosos. Los ataques a redes sociales fueron atacados de forma muy activa. La proliferación de códigos maliciosos a dispositivos moviles tal como el envío de mensajes SMS a números cortos.

Tabla 1.3: Semblanza de los códigos maliciosos desde 1986[Borghello, 2006].

Capítulo 2

Objetivos del análisis de códigos maliciosos

En el capítulo anterior se mencionaron las principales amenazas de códigos maliciosos existentes hasta la fecha y resulta evidente su alta peligrosidad. Además el promedio de pérdida económica según [Richardson, 2008] en el 2007 fue de US$350,424 por cada incidente de seguridad informática, de acuerdo con 494 encuestados en EUA de diversos sectores como compañías privadas, agencias de gobierno, instituciones financieras, instituciones médicas y universidades, un crecimiento considerable ya que en el año anterior la pérdida fue de US$168,000. Aunque no solamente se presentan daños económicos a las organizaciones y personas como consecuencia de algún código malicioso, es uno de los factores más representativos del daño sufrido por dichos códigos.

Sin embargo, medir las consecuencias de un ataque por un código malicioso se vuelve una tarea difícil ya que algunos de los reportes como [Richardson, 2008], [Microsoft, 2008], [Symantec, 2008] y [of Homeland Security et al., 2006] sólo realizan estimados de la información financiera acerca de la pérdida económica. Además, la mayoría de estos estudios son específicos del mercado estadounidense y no reflejan de manera global los resultados, a excepción de [Symantec, 2008]. Es también una tarea difícil, como en casi cualquier ámbito informático, determinar el costo de la información ya que es una cuestión subjetiva del analista y propia de cada organización y usuario. Ante estos problemas no se puede concluir más que estimados, ya que no existe una contabilidad estándar por las pérdidas debido al tiempo de inactividad de un equipo de cómputo, que resultan en una base y justificación para el posterior análisis de los códigos maliciosos.

Algunos datos interesantes que ayudan a determinar el porqué es necesario realizar un análisis de códigos maliciosos, de acuerdo con [Richardson, 2008] son:

- El porcentaje de presupuesto de TI gastado en seguridad es del 26 % en tan solo el 3 % de los encuestados.

Lo anterior nos indica que en algunas organizaciones no se ha tomado verdadera conciencia de lo que implica tener una mayor cantidad de presupuesto en el área de seguridad informática y también de la cantidad de organizaciones que están dispuestas a invertir una cantidad razonable del presupuesto de TI. Aunque un porcentaje específico de inversión en seguridad informática es incierto de determinar, sí es posible considerar que en un presupuesto de TI la seguridad se debe de evaluar en distintos sectores de TI de cualquier organización e incluso las necesidades y requerimientos de las organizaciones son distintas. El peor escenario para una organización es considerar a la seguridad informática como un gasto y/o considerar un presupuesto de seguridad informática después de ocurrir un incidente de seguridad, como dice un refrán mexicano:

> Una vez ahogado el niño, a tapar el pozo.

- Menos del 1 % de los encuestados gastan el 48 % de su presupuesto de seguridad, en la formación en materia de sensibilización.
- 52 % de los ataques sufridos por las organizaciones fueron causados por virus.
- 25 % fue negación de servicio.
- 21 % con bots en la organización.
- Las pérdidas económicas por virus, gusanos y spyware fue US$8,391,800.
- Las pérdidas económicas debido a bots en la organización fue US$2,869,600.

Se requiere prevención en la medida de lo posible, ya que en cuestiones de seguridad informática no hay certeza al 100 % debido a que existe el factor humano. Al estar implicado el factor humano se pueden tener errores de programación, de implementación de las tecnologías de seguridad informática, etcétera. El ser humano es un elemento estocástico.

Los códigos maliciosos han sido temidos por décadas por su capacidad de modificar archivos y/o procesos de los sistemas, inhabilitando las operaciones de los equipos por un tiempo corto o largo. Además los atacantes encontraron fama y reconocimiento como su único móvil para idear, codificar y difundir sus códigos maliciosos.

Ahora la situación ha cambiado, las amenazas y la fama han dejado de ser llamativas. Los creadores de códigos maliciosos han encontrado una forma más redituable para sus códigos maliciosos: el dinero.

En resumen el código malicioso es una amenaza y un problema crítico para las organizaciones, gobierno y usuarios. Ya que de acuerdo con [Symantec, 2008]:

- En el segundo semestre del 2007 499,811 nuevos códigos maliciosos fueron reportados, un incremento de 136 % desde el primer semestre del mismo año.
- Dentro de la lista de familias de códigos maliciosos detectados, de los primeros diez: 5 fueron caballos de Troya, 2 fueron gusanos, 2 fueron gusanos con un componente de puerta trasera y uno fue un gusano con un virus.

Crear códigos maliciosos no es tan difícil. De hecho, es tan simple como escribir unas cuantas líneas de código o descargar y configurar un conjunto de componentes fácilmente personalizables. Actualmente el código malicioso se inserta en aplicaciones inofensivas, incluyendo páginas web y correo electrónico. Esto hace difícil detectar y detener los códigos maliciosos antes de que puedan hacer daño.

En general, cuando un código (aplicación) llega a un equipo, éste tiene cuatro enfoques para protegerse de la infección y ataque del código malicioso:

- Analizar el código y rechazarlo si existe la posibilidad de daño hacia el equipo y/o aplicación.
- Reescribir el código antes de ejecutarlo de tal forma que no haga daño en el equipo y/o aplicación.
- Monitorear el código mientras se ejecuta y detenerlo antes de que haga daño.
- Auditar el código durante su ejecución y recabar información del daño realizado.

Las preguntas a las que debe de responder un análisis de códigos maliciosos son:

- ¿Cuál es el propósito del código malicioso?
- ¿Cómo llegó ahí?
- ¿Quién o qué es el responsable del código malicioso?
- ¿Cómo liberarse de él?
- ¿Cuál es su actividad?
- ¿Cuánto tiempo de existencia lleva?
- ¿Cuál es su forma y/o medio de propagación?
- ¿Cómo se puede prevenir, para que no vuelva a ocurrir?

La finalidad de prevenir y mitigar los códigos maliciosos es contener, en la medida de lo posible, los daños causados por éstos. En consecuencia, se busca que el análisis de códigos maliciosos dé soluciones que permitan reducir la capacidad de propagación de los códigos maliciosos, el mecanismo de conocer estas respuestas se encuentran dando solución a los siguientes cuestionamientos:

- ¿Cuáles son los indicadores basados en red que revelan la presencia y actividad del código malicioso?
- ¿Cuáles son los indicadores basados en equipos que revelan la presencia y actividad del código malicioso?
- ¿Es el código malicioso persistente?
- ¿Cuándo fue escrito, compilado e instalado el código malicioso?
- ¿Está basado en alguna otra herramienta ya conocida?
- ¿En qué lenguaje de programación fue escrito?
- ¿Está empaquetado el código malicioso?, si es afirmativo, ¿Qué programa fue usado para empaquetarlo?
- ¿El código malicioso tiene alguna funcionalidad que no permita su depuración?
- ¿Incluye alguna funcionalidad de rootkit?

Un análisis de código malicioso debe tomar en cuenta aspectos técnicos muy importantes los cuales se listan a continuación:

- Determinar si el código en cuestión es dañino examinándolo sin necesidad de utilizar herramientas extras, tales como motores antivirus.
- Simplificar el análisis de códigos maliciosos.
- Determinar el tiempo de análisis para que sea proporcional a la información poseída por el analista y las técnicas utilizadas por el código.
- Conseguir, en la medida de lo posible, el código fuente del software malicioso; esto ayuda a tener un mayor entendimiento de las técnicas utilizadas por los atacantes.
- Considerar la curva de aprendizaje que tendrá el analista al investigar los códigos maliciosos, no hay que olvidar que los atacantes siempre buscan no ser descubiertos y dificultar las actividades del analista.
- Obtener información relevante que ayude a entender los vectores de ataques del código malicioso, los daños ocasionados en las aplicaciones y equipos, la motivación del atacante y finalmente la estructura y el comportamiento del mismo.
- Desarrollar de un informe que permita restaurar a la aplicación y/o el equipo de cómputo al estado óptimo anterior a la infección.
- Elaborar propuestas de esquemas de solución y/o mitigación para la propagación y daño del código malicioso.

- Desarrollar modelos que representen el ecosistema del código malicioso, esto es necesario para futuras referencias en posteriores análisis.
- Desarrollar análisis automatizados basados en los patrones de comportamiento y técnicas de codificación usados por el código malicioso.

2.1. Alcances del análisis de códigos maliciosos

Se ha hablado sobre la importancia que tiene el análisis de códigos maliciosos, sin embargo, no se ha mencionado como se realiza dicho análisis. Para realizar dicha tarea surgen dos corrientes o métodos:

- Análisis estático o de código fuente.
- Análisis dinámico o de comportamiento.

En general ambas técnicas son poderosas y se complementan ambas. La distinción surge de la complejidad, el tiempo utilizado y la información obtenida.

El análisis estático consiste en obtener el código fuente del software malicioso para estudiarlo y determinar las actividades realizadas por dicho código.

Aunque resulta atractivo este enfoque, en la realidad no lo es tanto ya que se requiere de tener conocimientos de:

- Ingeniería inversa.
- Código ensamblador.
- Depuradores.

Además el tiempo que se invierte es mayor en comparación con el análisis de comportamiento. El tiempo utilizado en el análisis estático está muy ligado a la complejidad y longitud del código fuente del código malicioso aunado a factores como el cifrado de partes del código fuente y a la dificultad de analizar técnicas avanzadas de programación.

Las ventajas que presenta este tipo de análisis son la posibilidad de entender las técnicas utilizadas por los atacantes y como el código malicioso se aprovecha de las vulnerabilidades existentes en la aplicación o sistema.

Para realizar dicho análisis existe una gran diversidad de herramientas entre las que se puede mencionar:

- IdaPro[1]: es un desensamblador y depurador multi-procesador para Windows o Linux que ofrece un completo plugin de ambiente de programación. Explora programas binarios, para los cuales el código fuente no está siempre disponible, para crear mapas de su ejecución.
- Bintext[2]: es un extractor de texto de cualquier tipo de archivo e incluye la habilidad para encontrar texto ASCII plano, texto Unicode y cadenas de recursos.
- UPX[3]: es un empaquetador versátil de ejecutables que permite, en el caso del análisis estático poder desempaquetar el código malicioso.
- Gdb: es el depurador de GNU/Linux.
- File: utilería de GNU/Linux que permite identificar el tipo de archivo.
- LordPE: es una herramienta que permite editar y ver muchas partes de archivos ejecutables portátiles.
- OllyDbg[4]: es un depurador para Windows analizando a nivel ensamblador de 32 bits. Con un énfasis en análisis de código binario.
- Hexdump[5]: es una herramienta que muestra los contenidos de los archivos en formato hexadecimal.
- Strings: utilería de GNU/Linux que permite imprimir los caracteres legibles de un archivo. Es útil para determinar el contenido de archivos que no son de tipo texto.
- VMUnpacker[6]: permite analizar códigos maliciosos en ambientes virtuales.

Dentro del análisis de código se buscan los elementos que pueden ser usados para caracterizar e identificar un código malicioso. Estas características de detección están fuertemente basadas en la sintaxis, tales como:

- Secuencias de cadenas de byte.
- Secuencias de instrucciones.
- Expresiones regulares.
- Heurística:

  Utilizar técnicas avanzadas de reconocimiento de un bloque de memoria ejecutado y que fue previamente escrito.

  Valores sospechosos en el encabezado del archivo (PE).

[1] http://www.hex-rays.com/idapro/
[2] http://www.foundstone.com/us/resources/proddesc/bintext.htm
[3] http://upx.sourceforge.net/
[4] http://www.ollydbg.de/
[5] http://www.diamondcs.com.au/consoletools/hexdump.php
[6] http://www.sucop.com/download/20.html

- Ofuscación de código: código fuente que mantiene su funcionalidad pero que ha sido escrito de manera ininteligible.
- Transformaciones polimórficas cifran el cuerpo del código malicioso y agregan al inicio una rutina de descifrado.

  Diferente clave de cifrado usada para cada instancia.
- Representación sintáctica de código metamórfico cambia con cada instancia:

  Renombrar el registro.

  Inserción de código muerto (código que no se ejecuta).

  Reordenamiento en bloque.

  Sustitución de comandos.

El otro tipo de análisis consiste en observar el comportamiento del código malicioso y sus efectos al ejecutarse en su ambiente. Aquí también se desea obtener la estructura del código en relación a la operación del mismo, es decir, el flujo de operación del código malicioso.

Los objetivos del análisis de comportamiento son los siguientes:

- Monitorear la interacción del código malicioso con los distintos elementos del sistema operativo, aplicación, lenguaje de programación, etcétera; y obtener las funciones, llamadas y eventos surgidos de dicha interacción.
- Definir los elementos que hacen posible la ejecución del código malicioso.
- Determinar los efectos que surgen debido a la ejecución del código malicioso.

La desventaja que tiene este tipo de análisis radica en la necesidad de ejecutar el software malicioso, lo que supone la probabilidad de quedar infectado no sólo el equipo de pruebas sino también otros dispositivos y/o aplicaciones. Es por ello que se aconseja realizar el análisis de códigos maliciosos en ambientes controlados, es decir, lugares donde se tenga la certeza de no propagar el código malicioso.

Se sugiere al lector revisar los siguientes capítulos, en donde se expone a profundidad este tipo de análisis.

2.2. Objetivos de las organizaciones dedicadas a analizar y combatir el código malicioso

Como ya se mencionó en la introducción, la evolución de los códigos maliciosos ha traído como consecuencia que surjan organizaciones que tienen como objetivo generar soluciones a los problemas causados por el código malicioso. Dentro de este objetivo surgen diferentes aproximaciones

para alcanzar dicho objetivo tales como el obtener muestras del software malicioso, analizar el código malicioso y crear un reporte que indique la serie de acciones a tomar para evitar la infección y la propagación del mismo.

Ahora bien las organizaciones comerciales, más representativas, que se dedican a analizar y combatir el código malicioso, se encuentran (al momento de redactar este documento):

- Aladdin Knowledge Systems[7]
- AVG[8]
- Central Command, Inc.[9]
- Computer Associates International, Inc.[10]
- Frisk Software International[11]
- F-Secure Corporation[12]
- McAfee (a Network Associates company)[13]
- Network Associates, Inc.[14]
- Norman Data Defense Systems[15]
- Panda Software[16]
- Proland Software[17]
- Sophos[18]
- Symantec Corporation[19]
- Trend Micro, Inc.[20]
- Adaware[21]

[7] http://www.esafe.com/
[8] http://www.grisoft.com/
[9] http://www.centralcommand.com/
[10] http://www.cai.com
[11] http://www.f-prot.com/
[12] http://www.f-secure.com
[13] http://www.mcafee.com
[14] http://www.nai.com
[15] http://www.norman.com
[16] http://www.pandasoftware.com/
[17] http://www.pspl.com
[18] http://www.sophos.com
[19] http://www.symantec.com
[20] http://www.trendmicro.com
[21] http://www.lavasoft.com

- Aluria[22]
- Microsoft Anti-Spyware[23]
- Spy Sweeper[24]
- Spyware Detector[25]
- XoftSpy scan[26]

Existen además las organizaciones que también combaten los códigos maliciosos pero cuyo fin no es el lucro económico:

- AVAR (Association of Anti Virus Asia Reasearchers)[27] La misión de AVAR es prevenir la propagación y daño causado por software malicioso y desarrollar relaciones cooperativas entre expertos de software malicioso en Asia. Es una organización independiente y sin fin de lucro, que está orientada a la región de Asia pacífico. AVAR consiste de expertos prominentes en virus de computadora de varias áreas como Australia, China, Hong Kong, India, Japón, Corea, Filipinas, Singapur, Taiwán, Reino Unido y Estados Unidos de América.
- EICAR (European Institute for Computer Anti-Virus Research)[28] EICAR combina universidades, industria y medios más expertos en seguridad y derecho del gobierno, el ejército, autoridades judiciales y organizaciones de protección de la privacidad, cuyos objetivos son unir esfuerzos no comerciales contra la codificación y proliferación de códigos maliciosos y contra el crimen informático, fraude y el mal empleo de computadoras o redes, e inclusive explotación maliciosa de datos personales, basados en un código de conducta.
- Virus Bulletin[29] Es una publicación internacional en prevención, reconocimiento y eliminación de virus de computadora. Es una revista técnica especializada en desarrollos en el campo de virus de computadoras y productos antivirus.
- The WildList Organization International[30] Su misión es proveer de información precisa, oportuna y completa acerca de virus de computadora para usuarios y desarrolladores de productos. La WildList, una lista de virus de computadora encontrados en la red y reportados por un grupo diverso de aproximadamente 40 voluntarios calificados, está disponible gratuitamente por la organización.

[22] http://www.aluriasoftware.com
[23] http://www.microsoft.com/athome/security/protect/windowsxp/antispy.mspx
[24] http://www.webroot.com
[25] http://www.spywaredetector.us
[26] http://www.xoftspy.net/xoftspy/lp/17/
[27] http://www.aavar.org/
[28] http://www.eicar.com/
[29] http://www.virusbtn.com/
[30] http://www.wildlist.org/

- UNAM-CERT[31] Tiene un proyecto sobre malware cuyo propósito es identificar, analizar e informar sobre las amenazas de nuevos códigos maliciosos propagados en RedUNAM; colaboración e investigación de nuevas técnicas y estrategias empleadas por los intrusos y seguridad pro-activa para la red universitaria.

Existen muchas organizaciones enfocadas en los códigos maliciosos. La gran disyuntiva es ¿Por qué aún proliferan códigos maliciosos? no existe una respuesta acertada, ya que como se ha mencionado existen muchos atacantes que por mucho tiempo generaron códigos maliciosos por diversión y fama y ahora lo realizan por dinero.

Además de que es relativamente sencillo conseguir códigos maliciosos. Las organizaciones han tratado por años de unir esfuerzos y que a pesar de los errores y aciertos no se ha conseguido disminuir la proliferación de códigos maliciosos. Es necesario definir mecanismos que permitan mitigar a un nivel aceptable la existencia de dichos códigos.

La responsabilidad es de todos: empresas, organizaciones, gobierno, casas productoras de software, expertos en informática y usuarios. A veces sucede, sin embargo, que todos estos esfuerzos técnicos y organizacionales se ven comprometidos porque no se ha tenido el mismo celo en todos los eslabones de la -a veces muy larga- cadena de seguridad informática. Una cadena es tan fuerte como su eslabón más débil: aquel que primero se romperá ante un esfuerzo o estrés capaz de hacerlo vencer. Por eso, el eslabón más débil, el factor humano, compromete toda la seguridad de la cadena, y de nada sirve una seguridad reforzada en unos eslabones si ésta no se extiende por igual a todas y cada una de las piezas del sistema que son parte componente. El factor humano participa en varias etapas de esta cadena. Las vulnerabilidades asociadas a este factor podrían ir, desde una implementación incorrecta del software (aprovechado por los códigos maliciosos) hasta el descuido al utilizar los sistemas de cómputo (descargar software de dudosa procedencia, ejecutar software desconocido) y también la posibilidad de ser víctima de la ingeniería social.

[31] http://www.cert.org.mx
http://www.malware.unam.mx

Capítulo 3

Análisis de comportamiento de códigos maliciosos

Tal como se mencionó en el capítulo anterior, un análisis de comportamiento de códigos maliciosos consiste en ejecutarlo y recabar toda la información que pueda indicar los vectores de ataque utilizados, su actividad en el sistema, sus métodos de infección y propagación y finalmente obtener la forma de erradicarlo.

Las fases que componen un análisis de comportamiento son:

1. Captura del código malicioso.
2. Identificación del código malicioso.
3. Preparación del sistema y/o aplicación sin infectar, para una posterior comparación de resultados.
4. Ejecución del código malicioso.
5. Monitoreo del sistema y/o aplicación durante la ejecución del código malicioso.
6. Obtención de información del monitoreo en el sistema y/o aplicación y recolección de evidencias (rastros dejados por el código malicioso no detectado por el monitoreo).
7. Análisis del estado sin infección del sistema y/o aplicación con la información recolectada en el punto anterior.
8. Realizar un reporte ejecutivo y técnico de la actividad del código malicioso.

A continuación se explica cada una de las fases del análisis dinámico.

1. **Captura del código malicioso.** En esta etapa se realiza la obtención de códigos maliciosos, existen diversas técnicas y herramientas para realizar esto. Las organizaciones dedicadas a combatir los códigos maliciosos son quienes se encargan de recolectar una gran variedad de los mismos. Las técnicas mas comunes incluyen:

   - Nepenthes es un honeypot de baja interacción. Los honeypots de baja interacción emulan vulnerabilidades conocidas para reunir información de ataques potenciales. Está diseñado para emular vulnerabilidades que utilizan los códigos maliciosos para propagarse y poder capturarlos. Debido a las múltiples vulnerabilidades utilizadas por el software malicioso, Nepenthes se ha diseñado de forma modular. Hay módulos para:

     Resolver DNS asíncrono.

     Emular vulnerabilidades.

     Descargar archivos.

     Presentar los archivos descargados.

     Lanzar eventos.

     Manejador de shellcode.

     Es necesario tener conocimiento de las posibles vulnerabilidades que un código malicioso puede explotar para que el atacante y/o el código no reconozca que está atacando un honeypot. La desventaja al utilizar Nepenthes es la imperiosa necesidad de tener que escribir un nuevo módulo cada vez que aparece una nueva vulnerabilidad.

   - Honeytrap es una herramienta de seguridad escrita para observar ataques contra servicios de red. Reúne información en lo que se refiere a ataques basados en red conocidos y desconocidos. Honeytrap es muy utilizado cuando lo que se desea es cazar ataques desconocidos.

   - HoneyBow es un honeypot basado en el principio de honeypot de alta interacción. La mayor ventaja de esta herramienta es en reunir códigos maliciosos de día cero. Tiene tres componentes:

     MwWatcher es un programa que monitorea cambios en el sistema de archivos honeypot en tiempo real y atrapa potencial código malicioso. Se ejecuta en sistemas Windows de 32 bits.

     MwFetcher es un programa que extrae potencial código malicioso de un disco virtual de VMware o un disco físico comparando la lista de archivos infectados con la lista de archivos limpia. Actualmente sólo se ejecuta en GNU/Linux.

     MwSubmitter es un programa que envía muestras de código malicioso a la Alianza mwcollect usando el protocolo G.O.T.E.K. Solo se ejecuta en GNU/Linux.

   Honeyclients los cuales al contrario de las técnicas arriba mencionadas, son honeypots ejecutados desde el lado del cliente. Un honeyclient es una máquina virtual diseñada para ejecutar

una aplicación en específico hacia uno o más recursos remotos. Esto se lleva a cabo para verificar si el contenido obtenido de cada recurso remoto es de naturaleza maliciosa, volviendo al honeyclient comprometido debido al procesamiento del contenido malicioso. Algunos honeyclients son:

- Capture-HPC es un cliente honeypot de alta interacción, identifica servidores maliciosos interactuando con potenciales servidores maliciosos usando una máquina virtual dedicada y observando sus cambios de estado en el sistema. Si un cambio de estado en el sistema es detectado, ya que no ocurre otra actividad en el equipo, el servidor Capture con el que interactuó se clasifica como malicioso.
- HoneyClient está basado en un explorador de internet (IE/Firefox) y escrito en Perl. Está basado en estados y detecta ataques en clientes Windows monitoreando archivos específicos, directorios y entradas de registro. Se le asigna un conjunto de URLs en búsqueda de códigos maliciosos del lado del cliente.
- Strider HoneyMonkey está desarrollado por Microsoft para detectar y analizar sitios web almacenando códigos maliciosos. Su principal virtud es detectar códigos maliciosos que explotan vulnerabilidades de exploradores.
- SHELIA es un emulador de cliente que explora un directorio de correo especificado en la línea de comandos. En dicho directorio el emulador clientes es capaz de seguir cada URL y abrir cada adjunto. Monitorea los procesos y genera alertas cuando el proceso intenta ejecutar una operación inválida de un área de memoria que no se supone que sea de código ejecutable.
- UW Spycrawler utiliza el explorador Mozilla: está basado en estados y detecta ataques en clientes monitoreando archivos, procesos, registro y cierres inesperados del explorador. El mecanismo de detección está basado en eventos.
- Web Exploit Finder es un sistema que automáticamente detecta e identifica sitios web maliciosos. Se concentra en software malicioso que se instala sin ninguna intervención del usuario. Para detectar cambios en el equipo infectado utilizan técnicas de rootkit, es decir, monitoreo y evaluación de llamadas relevantes al sistema.
- HoneyC es un cliente honeypot de baja interacción, escrito en Ruby, Detecta servidores maliciosos examinando estáticamente la respuesta del servidor de cadenas maliciosas a través del uso de firmas de Snort.
- Monkey-Spider rastrea sitios web para exponer sus amenazas a clientes web. Utiliza soluciones antivirus para detectar malware.
- SpyBye funciona como un servidor proxy HTTP e intercepta todas las peticiones del explorador. Usa pocas reglas para determinar si los enlaces son dañinos, desconocidos e incluso peligrosos.

Incidentes de seguridad, éstos ocurren cuando surge, por ejemplo, una eventualidad debida a un posible código malicioso. Los incidentes de seguridad ocurren en cualquier organización que cuente con algún tipo de infraestructura informática. Un reporte de incidente de seguridad es el medio de comunicación que se establece entre usuarios y organizaciones finales con las organizaciones especializadas en combatir los códigos maliciosos.

2. **Identificación del código malicioso.** En esta fase se determina mediante una simple inspección el tipo de archivo y/o aplicación usado por el código malicioso para infectar y propagarse. La importancia de esta fase radica en el uso de herramientas de análisis a utilizar, ya que como se verá mas adelante no todas las herramientas soportan múltiples tipos de archivos.

   Un aspecto a cuidar al tratar de identificar el código malicioso es el observar solamente la extensión del archivo, ya que hay ocasiones en las que el archivo puede tener una extensión pero en realidad ser de otro tipo. La complejidad crece al tener un gran número de extensiones y tipos de archivos, por ello sólo se mencionarán algunas categorías relevantes:

   - Sin extensión.
   - Archivos de texto.
   - Archivos de datos.
   - Archivos de imagen.
   - Archivos de audio.
   - Archivos de vídeo.
   - Archivos web.
   - Archivos de fuente.
   - Archivos de sistema.
   - Archivos de configuración.
   - Archivos ejecutables.
   - Archivos comprimidos.

3. **Preparación del sistema y/o aplicación sin infectar, para un posterior análisis de resultados.** Esta fase es primordial, ya que es aquí donde se recaba la suficiente información que permite al analista realizar una comparación posterior. La complejidad radica en determinar lo que se quiere monitorear y como se va a monitorear. En resumen se busca tener un registro de la integridad de los archivos, directorios y memoria del sistema y/o aplicación.

4. **Ejecución del código malicioso.** La parte medular del análisis de comportamiento es ésta. Se requiere conocer el tipo de archivo del código malicioso para poder ejecutarlo sin problemas. Es importante mencionar que hasta el momento el analista sólo cuenta con información mínima del código malicioso tal como el nombre, tamaño y tipo de archivo del código malicioso.

   Es en esta fase en donde surge el concepto denominado correcta ejecución y no es otra cosa más que garantizar que se cuenta con los elementos requeridos por el código malicioso para su ejecución. La forma de llegar a una "correcta ejecución" depende del conocimiento y experiencia del analista, ya que como se menciona en las siguientes secciones algunos códigos no se ejecutan bajo determinadas condiciones.

   El no poder llegar a una "correcta ejecución" no siempre se traduce en un fracaso, al contrario, también nos arroja información valiosa acerca del código malicioso. El determinar los factores que no permitieron la infección del código malicioso se deben documentar ya que de esa manera se puede determinar qué versión del sistema y/o aplicación no es vulnerable al código malicioso, no olvidar que el objetivo es determinar qué o cuales son las condiciones que permiten la ejecución, infección y propagación del software malicioso.

5. **Monitoreo del sistema y/o aplicación durante la ejecución del código malicioso.** Cuando el analista llega a esta etapa, significa que se tiene información suficiente de los factores que posibilitan la infección de un sistema y/o aplicación. Además en esta etapa se requiere de la ejecución de distintas técnicas y herramientas que permitan tener conocimiento de la actividad del código malicioso. El seguir el flujo de ejecución de un código malicioso no es tarea fácil debido a que en ocasiones el atacante conoce de antemano que su código podría ser analizado y por ello dificulta su análisis. Como en prácticamente cualquier actividad humana, la experiencia es la mejor herramienta de análisis, ya que permite tener referencias de la actividad de otros códigos maliciosos y que posiblemente ya han sido analizados. Recordando que en el ambiente de los códigos maliciosos es común encontrar familias de códigos maliciosos, los cuales presentan actividad similar pero con cambios mínimos para evitar ser detectados por aplicaciones de protección contra códigos maliciosos. Al ejecutar un código malicioso el analista no siempre tiene conocimiento del tiempo requerido por un código malicioso para infectar un equipo. El considerar un tiempo prudente también afecta el tiempo invertido en el análisis del código, en ocasiones el analista se verá en la necesidad de responder de manera pronta y eficaz para obtener información del comportamiento del código malicioso analizado. En consecuencia, la decisión es del analista, aunque es aconsejable detener la ejecución del código malicioso una vez que se haya determinado tener información suficiente de la actividad del mismo.

El punto importante en esta fase es como finalizar la ejecución del código malicioso. Existen diferentes mecanismos para detenerlo, aunque algunos códigos son capaces de volver a ser ejecutados a pesar de ser cerrados. Entre ellos están:

- Matar los procesos generados por el código malicioso.
- Cerrar la aplicación infectada y/o usada por el código malicioso.
- Detener el tráfico de red generado por el código malicioso.
- Eliminar el o los archivo(s) que acompañan al código malicioso.
- Apagar el dispositivo donde reside el código malicioso.

6. **Obtención de información del monitoreo en el sistema y/o aplicación y recolección de evidencias (rastros dejados por el código malicioso y que no hayan sido detectados en la información obtenida por las herramientas de la fase del monitoreo).** Existen distintas técnicas para la obtención de información tales como:

   - Auditoría informática: utilizar los conceptos de la auditoría como configuración de acceso a recursos del sistema y/o aplicación, violación de directivas de seguridad configuradas por default, etcétera.
   - Análisis forense: recabar información relevante para el análisis de comportamiento mediante técnicas de análisis forense, por ejemplo reconstruir la actividad del código malicioso con el uso de herramientas de seguimiento de acceso al sistema de archivos.

   Ambos enfoques tienen sus ventajas y desventajas y al utilizarse ambas se puede complementar. Sin embargo la limitante en ambos casos es la inversión de tiempo requerido. Esta etapa es aconsejable en casos extremos en donde la etapa de monitoreo no arrojó la suficiente información de la actividad del código malicioso.

7. **Análisis del estado sin infección del sistema y/o aplicación con la información recolectada de la actividad del código malicioso.** En esta etapa se requiere tener de un profundo conocimiento del sistema y/o aplicación infectada, ya que ello ayudará a minimizar la generación de falsos positivos, es decir, eventos que surgieron durante la etapa de ejecución y monitoreo del código malicioso pero que no fueron generados por el software malicioso.

   La forma de llevar a cabo esta fase es mediante un análisis de los estados sin infectar e infectado del sistema y/o aplicación. Para conseguir dicha meta se requiere de:

   - Relación espacio-forma-tiempo de archivos, directorios y memoria del sistema y/o aplicación. Se deben responder preguntas como:

     ¿El archivo/directorio/memoria cambio de ubicación?

     ¿El archivo/directorio/memoria fue modificado en su estructura y/o contenido?

     ¿El archivo/directorio/memoria fue accedido o leído?

- Tráfico de red generado habitualmente por el sistema y/o aplicación comparado con el tráfico generado debido a la ejecución del código malicioso.
- Procesos e hilos que muestren diferencias sustanciales en ambos estados.
- Uso de los recursos físicos.

8. **Realizar un reporte ejecutivo y técnico de la actividad del código malicioso.** En el resumen ejecutivo se incluye una reseña breve de la actividad del código malicioso. Se debe escribir sin cuestiones técnicas, ya que quienes lo leerán son usuarios comunes, ejecutivos y/o jefes. El objetivo que se persigue en el resumen ejecutivo es comunicar a los lectores de la peligrosidad, actividad y posible erradicación del código malicioso. Debe ser corto y mantener un contexto de cómo el código malicioso impacta en las actividades diarias del usuario.

   En el reporte técnico puede variar su longitud, dependiendo del alcance y detalle del análisis realizado. Debe contener los informes de las distintas etapas del análisis de comportamiento.

Como ya se ha mencionado en un análisis de comportamiento, el aspecto fundamental es ejecutar el código malicioso. En este caso existen dos vías para realizar el análisis.

3.1. Análisis en ambientes reales

Cuando se utiliza un análisis en tiempo real se refiere a la ejecución del código malicioso en un dispositivo físico (Hardware). La máxima ventaja de ejecutar un código malicioso de esta forma es la posibilidad de evitar los problemas tales como la detección de ambientes virtuales y que se explicará en la siguiente sección. Una desventaja evidente es el gasto en dispositivos tales como equipos de cómputo, hub's, switches, etcétera Además del consumo en energía eléctrica.

En el ámbito del análisis de comportamiento, han surgido distintos enfoques para realizar el análisis de comportamiento. Estos enfoques han surgido como una respuesta a las necesidades del análisis ya que en ocasiones se requiere de un solo sistema de cómputo y en otras se requiere una red de datos. Estos enfoques que se han propuesto se encuentran divididos en dos ramas:

- *SandNet*: se refiere a un ambiente controlado en el cual se delimita lógicamente y físicamente el acceso a la red corporativa. Una ventaja de este enfoque es la posibilidad de integrar más de un dispositivo de red (computadora, hub, switch, etcétera) con esto se logra tener un ambiente lo más similar y probable a un escenario de ataque real, ver figura 3.1 en la página siguiente.
- *SandBox*: se refiere a un ambiente controlado, conformado por un equipo de cómputo ya sea físico y/o virtual. La ventaja en este enfoque radica en evitar un gasto en equipos de red extra.

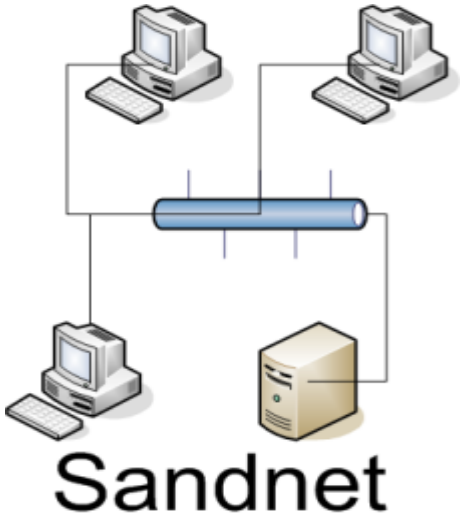

Figura 3.1: Esquema de una Sandnet.

La gran limitante de este enfoque es el alcance del análisis, ya que no se puede determinar cómo se comporta el código malicioso en una red, ver figura 3.2 en la página siguiente.

Una técnica propuesta por [Jensen, 2008] establece la instalación de un laboratorio físico con 5 equipos instalados con distinto software resaltando que todos ellos están conectados a un hub, además de contar con software para verificar la integridad de archivos. La importante aportación de esta técnica es la posibilidad de contar con software para detectar vulnerabilidades como es el caso de Nessus y de software para verificar la integridad del sistema de archivos en el equipo con Windows, ver figura 3.3. Para el análisis de comportamiento de códigos maliciosos existe un conjunto de herramientas (software y hardware) así como herramientas de análisis (basados en métodos probabilísticos-estadísticos y de software) de reciente creación en entornos académicos con ayuda, en algunos casos, de software y conceptos ya diseñados. Aunque estas herramientas no cubren toda la gama de códigos maliciosos son de utilidad en aquellos en los cuales han sido probados.

Al analizar el comportamiento de un código malicioso es necesario mitigar la propagación del mismo, es decir, tener un ambiente controlado en el cual el código malicioso no infecte al sistema que hospeda la sandbox y/o a los nodos de la red en una sandnet/sandbox. Además de que una de las metas del análisis de códigos maliciosos es el de minimizar la propagación del software malicioso, y es en esta fase donde [Llewellyn-Jones et al., 2005] proponen un método para evitar la propagación del código malicioso una vez que ha sido identificado. Su método, que hace uso de autómatas celulares, es una correlación entre los nodos de una red de cómputo ubicua y células de un autómata celular usando la propagación de confianza entre los dispositivos, ver figura 3.4 en la página siguiente. En ocasiones un código malicioso puede tener múltiples ejecuciones, este fue un problema que [Moser et al., 2007] detectaron al analizar un código malicioso el analista

Figura 3.2: Esquema de una Sandbox.

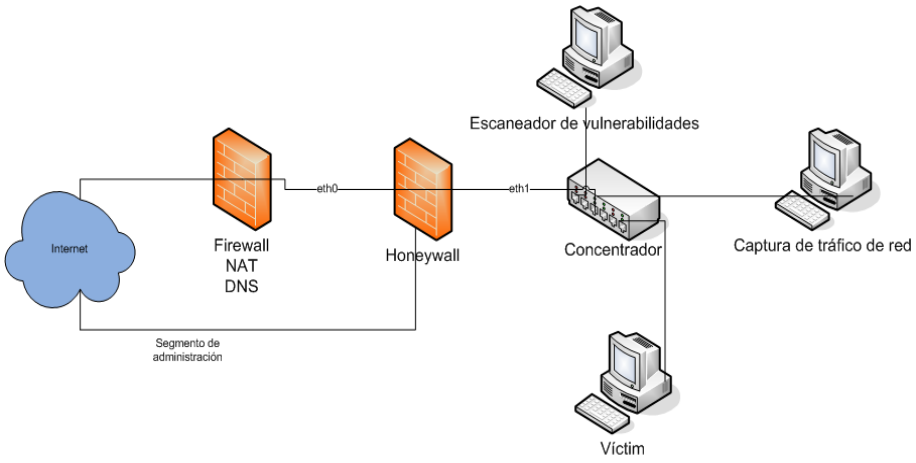

Figura 3.3: Banco de pruebas por [Jensen, 2008].

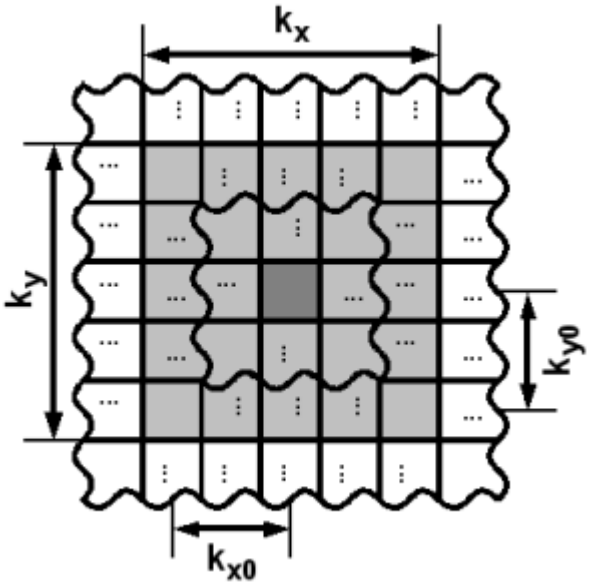

Figura 3.4: Uso de autómatas celulares por [Llewellyn-Jones et al., 2005].

solamente es capaz de observar una sola ejecución del código. Esto se debe a que ciertas acciones maliciosas son lanzadas bajo ciertas circunstancias, en consecuencia ellos proponen un sistema que permite explorar múltiples trayectorias de ejecución e identificar las acciones maliciosas que son ejecutadas solo cuando ciertas condiciones son conocidas.

Así mismo algunos analistas han intentado determinar el comportamiento de un código malicioso mediante el uso de conceptos propios de las ciencias de la computación. El análisis dinámico de contaminación propuesto por [Yin et al., 2008] es interesante ya que mediante el uso de una herramienta llamada Panorama, creada por ellos, permite obtener un grafo en donde se muestra los procesos y archivos usados por la muestra del código malicioso a analizar. El enfoque usado en Panorama permite además identificar las conexiones de red generadas durante la ejecución de la aplicación maliciosa, ver figura 3.5 en la página siguiente. Un aspecto que en ocasiones se da por hecho en el análisis de comportamiento de códigos maliciosos es la fase de restauración del sistema y/o aplicación, es decir, recuperarlo de tal forma que sus elementos se encuentren en el estado inicial sin la infección del código malicioso. En [Hsu et al., 2006] se encuentra una técnica para la eliminación del código malicioso y la reparación del sistema. Esta técnica monitorea y reporta operaciones de los programas no confiables. Mediante el uso de las bitácoras remueve los códigos maliciosos y sus efectos en el sistema.

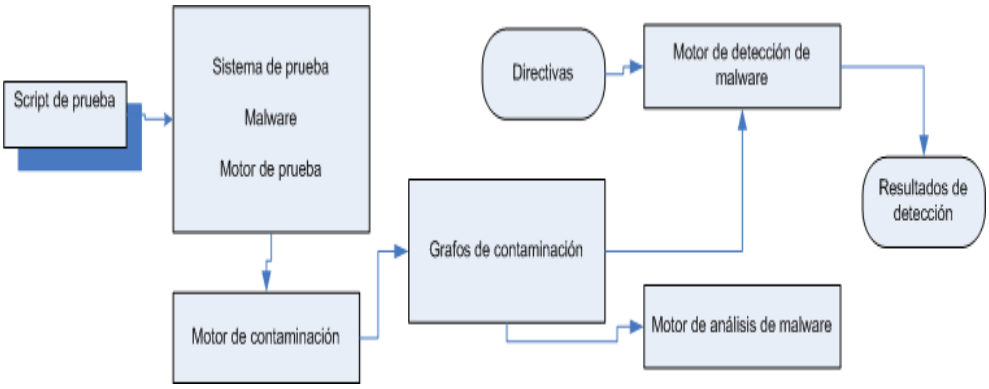

Figura 3.5: Esquema de herramienta Panorama por [Yin et al., 2008].

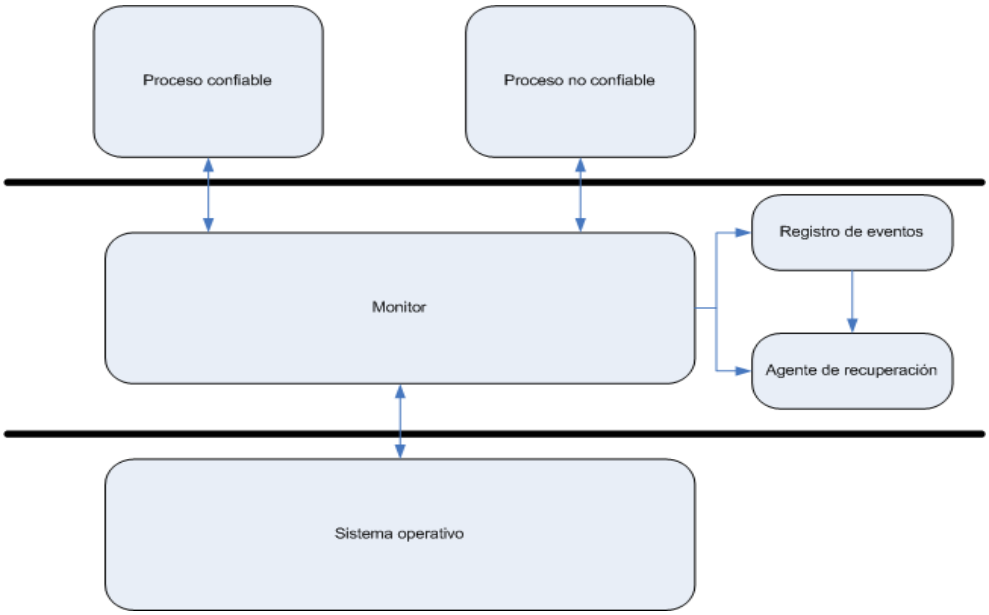

Figura 3.6: Esquema de restauración por [Hsu et al., 2006].

Una herramienta que es muy útil dentro del enfoque de *SandNet* es TRUMAN[1]. TRUMAN es una herramienta para analizar códigos maliciosos (sólo ejecutables) en un ambiente que está aislado y con la posibilidad de acceso "virtual" a Internet con el cual el código malicioso puede interactuar. Se ejecuta en hardware nativo evitando así un problema con los ambientes virtuales que se explicará en la siguiente sección. Una problemática que surge con el enfoque de análisis en ambientes reales es la necesidad de restaurar los dispositivos a un estado limpio; para resolver este problema TRUMAN automatiza este proceso, dejando al analista con sólo un mínimo de trabajo que hacer para obtener un análisis inicial del código malicioso.

TRUMAN se compone de una imagen de arranque Linux y un conjunto de scripts escritos en Perl. Además incluye la utilidad pmodump, una herramienta que reconstruye el espacio de memoria virtual de un proceso de un volcado de Memoria física.

Existen en el mercado una gran variedad de software que es utilizado en el análisis de comportamiento de códigos maliciosos y en las siguientes tablas se muestran algunas herramientas mas representativas y que se aconseja su uso, ver tabla 3.1 en la página siguiente, tabla 3.2 en la página 52, tabla 3.3 en la página 53 y tabla 3.4 en la página 53.

Nombre	Descripción
MANDIANT Red Curtain[2]	Es una herramienta, liberada como software libre que examina archivos ejecutables para determinar qué tan sospechosos son basados en un conjunto de criterios. Examina múltiples aspectos de un ejecutable: entropía, indicaciones de empaquetado y firmas de compilador, la presencia de firmas digitales y otras características para generar una puntuación de amenaza.
Process Explorer[3]	Muestra una lista de los procesos actualmente activos, incluye los nombres de las cuentas de los dueños de dichos procesos, etcétera.
RAPIER[4]	Es una herramienta de seguridad que permite adquirir información relevante (procesos en ejecución, ubicación de esos procesos en el disco duro, puertos que esos procesos usan, archivos abiertos, usuarios con sesión iniciada, lista de todas las conexiones de red, etcétera) del código malicioso de forma automatizada. Tiene dos versiones: servidor y cliente, permite seleccionar los módulos que se desean incluir en el reporte (entre los módulos que tiene se encuentran SecCheck y el módulo de red con fport).

[1] http://www.secureworks.com/research/tools/truman.html
[2] http://www.mandiant.com/redcurtain.htm
[3] http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx
[4] http://code.google.com/p/rapier/

Process Monitor[5]	Es una herramienta de monitoreo para la plataforma Windows que muestra en tiempo real el sistema de archivos, el registro de Windows y la actividad de procesos e hilos.
SysAnalyzer[6]	Es una herramienta de análisis de códigos maliciosos de manera automatizada, monitorea varios aspectos del sistema y estados de procesos. Monitorea y compara: ▪ Procesos en ejecución. ▪ Puertos abiertos. ▪ Controladores cargados. ▪ Bibliotecas infectadas. ▪ Cambios en las llaves del registro. ▪ APIs llamadas por un proceso específico. ▪ Modificaciones de archivo. ▪ Tráfico HTTP, IRC y DNS.
Autoruns[7]	Como se mencionó en el primer capítulo, algunos códigos maliciosos garantizan su ejecución utilizando algunas llaves del registro, en este caso este programa permite de manera gráfica obtener todas las aplicaciones, servicios y proceso que se ejecutan al iniciar el equipo.

Tabla 3.1: Software relacionado a listar procesos.

[5] http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx?PHPSESSID=d926
[6] http://labs.idefense.com/software/malcode.php
[7] http://technet.microsoft.com/en-us/sysinternals/bb963902.aspx

Virus total[8]	Es un portal de internet en el cual se puede proporcionar la muestra de código malicioso el cual será analizado por distintos motores antivirus. La utilidad de virus total radica en la fácil identificación de los códigos maliciosos, es decir, se puede conocer qué motores antivirus ya han identificado el código malicioso y cuáles no.
VirSCAN[9]	Es un servicio de escaneo en línea, de operación similar a Virus total.
Jotti[10]	Es otro portal de internet que funciona de manera similar a virus total.
Sunbelt[11]	Permite analizar un código malicioso y regresa un reporte web o un reporte en texto al correo electrónico ingresado. Este reporte incluye información como procesos utilizados, puertos abiertos, actividad de red, actividad del registro de Windows; este reporte da un panorama general acerca de la actividad del código malicioso sin la necesidad de analizarlo.
SecCheck SCU[12]	Es una herramienta que ayuda en la detección y eliminación de aplicaciones maliciosas.

Tabla 3.2: Software en Internet para el análisis de códigos maliciosos.

[8] http://www.virustotal.com
[9] http://www.virscan.org/
[10] http://virusscan.jotti.org/
[11] http://research.sunbelt-software.com/Submit.aspx
[12] http://mynetwatchman.com/tools/sc/

BgInfo[13]	Es una aplicación que arroja información importante del sistema como su nombre, dirección IP, versión del sistema operativo, CPU, memoria, etc.
PsFile[14]	Es una aplicación que muestra una lista de archivos en un sistema que son abiertos de forma remota, de gran utilidad para determinar qué nodo en la red ha abierto un archivo. Muestra el recurso abierto, el usuario que lo tiene abierto y el tipo de acceso que tiene sobre el recurso (lectura, escritura).
RootkitRevealer[15]	Es una utilidad para la detección avanzada de rootkits.
TCPView[16]	Aplicación que muestra información de conexiones TCP y UDP, incluyendo las direcciones locales y remotas y el estado de la conexión TCP.
Fport[17]	Identifica puertos desconocidos y sus aplicaciones asociadas a dichos puertos.
Sniffers(tcpdump, wireshark, snort, etcétera)	Dentro de esta categoría se engloba el software, elegido por el analista, para recolectar información de red realizada por el código malicioso.

Tabla 3.3: Software para análisis de tráfico de red.

BinText[18]	Es un extractor de texto de cualquier archivo y de análisis de cadenas.
Hfind[19]	Busca archivos ocultos en el disco duro. Busca aquellos archivos que tienen el atributo de oculto o una combinación de atributos del sistema/directorio.
Regshot[20]	Es una utilidad para comparar el registro de Windows entre dos estados, por ejemplo el estado sin infección y el estado infectado.

Tabla 3.4: Herramientas varias para el análisis de comportamiento de códigos maliciosos.

[13] http://technet.microsoft.com/es-es/sysinternals/bb897557(en-us).aspx
[14] http://technet.microsoft.com/en-us/sysinternals/bb897552.aspx
[15] http://technet.microsoft.com/es-es/sysinternals/bb897445(en-us).aspx
[16] http://technet.microsoft.com/es-es/sysinternals/bb897437(en-us).aspx
[17] http://www.foundstone.com/us/resources/proddesc/fport.htm
[18] http://www.foundstone.com/us/resources/proddesc/bintext.htm
[19] http://www.foundstone.com/us/resources/proddesc/forensictoolkit.htm
[20] http://sourceforge.net/projects/regshot

3.2. Análisis en ambientes virtuales

Los ambientes virtuales implementados en software permiten compartir hardware entre múltiples sistemas operativos ejecutándose al mismo tiempo. La flexibilidad del uso de ambientes virtuales se debe a la reducción en la inversión de equipo de cómputo. Al igual que el análisis en ambiente real al ejecutar un código malicioso se debe evitar que el software salga del ambiente controlado, es decir, que infecte a otros equipos, sistemas y/o aplicaciones, como lo es el equipo nativo en el cual reside el ambiente virtual, e incluso al equipo que hospeda al ambiente virtual. La otra desventaja y que ocasiona dolores de cabeza a los analistas de códigos maliciosos es la posibilidad de tener código malicioso que reconoce el ambiente virtual y con ello modifica su ejecución: desde no ejecutarse en dicho ambiente hasta la ejecución del mismo pero con una actividad (comportamiento) distinto al que en realidad realiza.

Dentro de la gran gama de productos de software de virtualización, resaltan algunos de ellos que por su popularidad, facilidad de uso o licencia se han convertido en los preferidos por los analistas de códigos maliciosos.

VMware[21]

Es uno de los productos virtualizadores mas utilizados. Existen diferentes productos realizados por esta compañía, aunque se aconseja utilizar su versión Server la cual requiere únicamente de un registro. VMware permite la simulación de múltiples equipos (sistemas operativos) de forma simultánea en un solo equipo. Existen varias ventajas al usar este producto se obtienen los siguientes beneficios:

- Es benéfico tener múltiples sistemas en el laboratorio de análisis, de esta forma es posible que el código malicioso pueda interactuar con componentes de internet simulado.
- Es posible tomar una instantánea del estado del sistema antes de infectarlo y tomando instantáneas periódicas a lo largo del análisis ahorra tiempo[22]. Esta funcionalidad permite revertir el sistema al estado deseado de forma casi instantánea.
- La opción de red *host only* es conveniente para interconectar sistemas virtuales usando una red simulada sin la necesidad de hardware adicional. Esto permite, además evitar que el laboratorio de análisis esté conectado a una red de producción. La opción de *host only* permite a cualquier sistema ver todo el tráfico en la red simulada.

Para configurar un laboratorio de análisis con VMware es sencillo. Se requiere de suficiente memoria RAM (no menor a 512 MB) y espacio en disco donde se albergarán los equipos virtuales.

[21] http://www.vmware.com/products/server/

[22] Esta característica no está disponible en la versión Server pero sí en Workstation, ACE, Fussion, etcétera

Se requiere además del software necesario: VMware Workstation o Server y los medios de instalación para el o los sistemas operativos a utilizar.

- VMware emula el hardware de una computadora, de manera que es posible instalar el sistema operativo en la máquina virtual usando el asistente de VMware.
- Una vez instalado el sistema operativo se aconseja al lector instalar el paquete denominado *VMware Tools*, que optimiza el sistema para una mejor funcionalidad con VMware.
- Instalar el software de análisis de códigos maliciosos.

Es recomendable tener varias máquinas virtuales con diferentes sistemas operativos en el laboratorio, cada uno representando el sistema operativo que el código malicioso ataca. Esto permite la observación de los programas maliciosos en sus ambientes nativos.

Al manipular códigos maliciosos se deben tomar las precauciones para no infectar sistemas en producción. Esto puede suceder cuando se maneja el software malicioso de forma inapropiada o cuando la muestra explota una vulnerabilidad en la configuración de VMware y escapa del concepto de *sandbox*.

Se sugiere seguir los siguientes lineamientos para mitigar estos riesgos:

- Mantener VMware con los parches de seguridad liberados por el fabricante.
- Dedicar el equipo huésped unicamente al laboratorio basado en VMware, no usarlo para otros propósitos.
- No conectar el laboratorio a una red de producción.
- Monitorear el equipo huésped con software de detección de intrusos basado en *host*, además de verificadores de integridad de archivos.
- Periódicamente restaurar el sistema operativo y sus aplicaciones huésped a su estado inicial, es decir, reinstalar el software.

QEMU[23]

Es un virtualizador muy utilizado normalmente en distribuciones Linux, aunque también existe soporte para otros sistemas operativos. Para ejemplificar la implementación de un laboratorio de análisis con QEMU se detallarán los pasos necesarios bajo la distribución Debian.

El primer paso es instalar qemu y su acelerador kqemu-common.

[23] http://http://bellard.org/qemu/

```
#apt-get install qemu kqemu-common kqemu-source
```

Con esto se logra tener instalado el virtualizador, el siguiente paso es instalar las máquinas virtuales que serán las que componen el laboratorio.
El primer paso para instalar una máquina virtual es crear el disco duro donde reside la máquina:

```
qemu-img create -f qcow xp.qcow 2G
```

Se crea un archivo de respaldo para ese disco duro,

```
qemu-img create -b xp.qcow -f qcow xp2.qcow
```

Se procede a instalar el sistema operativo windows

```
qemu -cdrom xp.iso -boot d xp2.qcow
```

Para instalar el servidor Linux se procede de la siguiente manera:

```
qemu-img create -f qcow deb.qcow 1G
qemu-img create -b deb.qcow -f qcow deb2.qcow
qemu -cdrom debian-40r3-i386-netinst.iso -boot d xp2.qcow
```

Una vez que se tengan instalados ambos sistemas operativos, es necesario instalar las herramientas para el análisis de malware. Para arrancar nuevamente las máquinas virtuales únicamente se necesita ejecutar lo siguiente:

```
qemu -hda xp2.qcow -m 96 -localtime
qemu -hda deb2.qcow -m 64 -localtime
```

Es necesario aclarar que por default las máquinas virtuales tienen la posibilidad de salir a internet mediante qemu y que los cambios hechos en la máquina virtual no son reversibles, para evitar escribir los cambios hechos durante el análisis de códigos maliciosos es agregar la opción de -snapshot.
Una vez instaladas las herramientas se configura la red para la comunicación entre ambas máquinas.

VirtualBox[24]

El programa Sun xVM VirtualBox es una aplicación de virtualización de diferentes sistemas operativos, puede ejecutarse en distintos sistemas operativos. A diferencia de su competidor VMware, éste es de distribución gratuita.

El proceso de instalación y configuración de un laboratorio de análisis de códigos maliciosos es bastante sencillo.

[24] http://www.sun.com/software/products/virtualbox/get.jsp

- Primero se debe descargar la aplicación del sitio web: http://www.sun.com/software/products/virtualbox/get.jsp.
- Una vez descargada la aplicación se procede a ejecutar el instalador.
- Se ejecuta la aplicación, donde se presenta una interfaz de administración de todas las máquinas virtuales hospedadas en el equipo.
- Para crear una máquina virtual se presiona el botón de Nueva, se ejecutará un asistente.
- Se le asigna el tamaño de memoria RAM que se le desea dar a la máquina virtual.
- Se crea un nuevo disco duro virtual, presionando el botón de nuevo. Se ejecutará un asistente para la configuración del disco duro virtual.

  Se selecciona una expansión dinámica.

  Se asigna un tamaño al disco duro y la ubicación del mismo en el sistema de archivos del equipo huésped.

  Se presenta un resumen con los datos proporcionados.

  Existe la posibilidad de asignar un disco duro ya configurado.
- En el asistente se ingresa el nombre que tendrá la máquina virtual y se especifica el tipo de sistema operativo que va a hospedar.
- Una vez finalizado el asistente se muestra una interfaz de administración de todas las máquinas virtuales hospedadas en el equipo.
- Al presionar el botón de configuración se muestra una ventana con los parámetros de configuración de la máquina virtual; en la sección de red se puede especificar el tipo de tarjeta de red a utilizar (tiene 3 tipos de interfaces de red) y además se puede configurar el tipo de conexión que tendrá esa tarjeta de red. Es necesario agregar que para los requerimientos de este laboratorio se escogerá una red privada, es decir, sólo habrá comunicación entre los equipos virtuales que integran el laboratorio de análisis. Para lograr esto únicamente se configura el nombre de red que compartirán estos equipos.
- Una vez configurado los parámetros necesarios se da clic en el botón de Iniciar. Se instala el sistema operativo junto con las herramientas de análisis y el complemento de VBoxGuest Additions (propio de VirtualBox, para un mejor manejo de la máquina virtual). Una vez realizado esto se procede a crear una instantánea del equipo limpio, esto se logra desde la consola de administración en la segunda pestaña del lado derecho y dando clic en el botón tomar instantánea.

Virtual PC[25] Virtual PC es un software virtualizador creado por Microsoft. Aunque su principal desventaja, y muy obvia, es que únicamente se puede instalar en sistemas operativos Windows. Es posible usarlo de manera gratuita.

El proceso de instalación y configuración es igualmente sencillo, es necesario aclarar que se debe seleccionar la opción de red local en todas las máquinas del laboratorio.

Norman SandBox [Norman, 2008] es una utilidad que automatiza, simplifica y acelera el proceso de obtención de información al analizar código malicioso. Norman provee un análisis comprensible de la acción de cualquier archivo ejecutable. Después de que un archivo ha sido procesado genera reportes con una descripción de las acciones del archivo en una vista de bitácora API y un resumen del reporte.

El resumen del reporte incluye los siguientes bloques de información:

- Categorías Archivo/Código malicioso.
- Cambios en el sistema de archivos.
- Cambios en el registro y en la configuración del sistema.
- Detalles de los servicios de red.
- Información del procesador.

En ocasiones al analizar un código malicioso puede ocurrir que se creen nuevos proceso que podrían ser detectados mediante las herramientas del sistema operativo, sin embargo, algunos códigos maliciosos ocultan sus procesos y en consecuencia hace más difícil su erradicación. En [Wen et al., 2008] se hace uso de una herramienta llamada Libra, creada por los autores, que permite observar todos los procesos del sistema operativo mediante una vista de árbol de la lista de procesos.

Como ya se mencionó al inicio uno de los problemas de los ambientes virtuales es la posibilidad de que el código malicioso sea capaz de detectar el ambiente virtual, con las mencionadas consecuencias. Es por ello que [Carpenter et al., 2007] han explicado los mecanismos para evitar que el código malicioso detecte el ambiente virtual (específico para el software virtualizador VMware), es importante mencionar que las modificaciones propuestas tienen como consecuencia un cambio sustancial en la interacción usuario-software. En el mismo documento los autores hacen mención de una herramienta, VMmutate, que realiza los cambios aconsejados en dicha investigación.

[25] http://www.microsoft.com/downloads/details.aspx?displaylang=es&FamilyID=28c97d22-6eb8-4a09-a7f7-f6c7a1f000b5

Capítulo 4

Diseño de una metodología para el análisis de comportamiento de códigos maliciosos

A lo largo de este documento se ha hecho énfasis en la importancia del análisis de códigos maliciosos y en los resultados que dicho análisis arroje. El análisis de comportamiento de software malicioso tiene una serie de pasos que se han esbozado en el capítulo anterior. La inmensidad y complejidad de algunos códigos maliciosos (y sus variantes) hacen de este tipo de análisis una tarea compleja.

Se hace necesario el diseño de una metodología que agrupe, en lo posible, los diferentes enfoques que se han presentado a lo largo del documento y que en las referencias se explican a detalle. El propósito de esta metodología es permitir que aquellos analistas de códigos maliciosos que la sigan puedan determinar de forma clara la actividad de un código malicioso.

Para poder desarrollar una metodología para el análisis de códigos maliciosos, ya sea dinámico o estático, es necesario entender el ciclo de vida de un código malicioso. Los códigos maliciosos tienen toda una fase de desarrollo la cual de acuerdo con [Vixie and Dragon, 2008] se compone de cuatro etapas las cuales reflejan los diferente momentos en los cuales el código malicioso podría encontrarse analizado, ver figura 4.1 en la página siguiente.

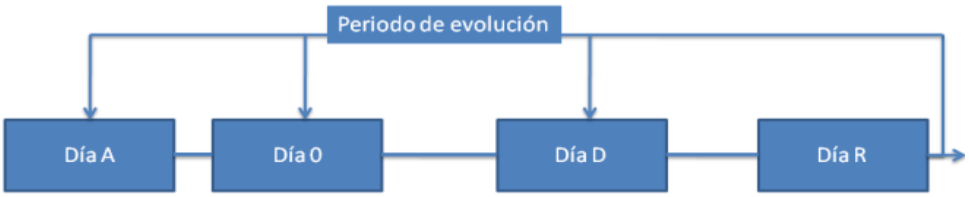

Figura 4.1: Ciclo de vida de un código malicioso.

- Día A: código malicioso con autoría (es conocido el código malicioso).
- Día 0: el código malicioso se libera (no es reconocido el código malicioso).
- Día D: primera oportunidad para la detección del código malicioso.
- Día R: respuesta ante la aparición del código malicioso (por ejemplo actualización de firmas de virus).
- Día E: cuando el código malicioso evoluciona hacia una nueva versión basado en código y patrones de comportamiento de códigos maliciosos anteriores.

El entender el ciclo de vida de un código malicioso permite que las actividades de análisis de comportamiento de códigos maliciosos se realice tomando en cuenta el proceso de evolución del código que se analiza. Es un factor a tomar en consideración debido a que la información a obtener del comportamiento del código malicioso está determinado por la etapa en la que se encuentre la muestra analizada.

Para el diseño de esta metodología se proponen los siguientes lineamientos:

- Se hace uso del análisis DAFO o FODA[1] en cada etapa de la metodología.

[1]Es una metodología de estudio de la situación competitiva de una empresa dentro de su mercado y de las características internas de la misma, a efectos de determinar sus Debilidades, Amenazas, Fortalezas y Oportunidades. Las

- Está basada para códigos maliciosos que atacan la plataforma Windows[2]
- Se propone una combinación de varios esquemas del área de seguridad informática:

  Proceso de administración del riesgo.

  Auditoría informática.

  Análisis forense.

  Manejo y respuesta a incidentes.

Para el desarrollo de esta metodología es necesario explicar los lineamientos anteriormente descritos.

4.1. Análisis DAFO o FODA

El análisis FODA, aunque enfocado a empresas y negocios, es una herramienta de gran utilidad para entender y tomar decisiones en toda clase de situaciones. FODA es el acrónimo de Fortalezas, Oportunidades, Debilidades y Amenazas que se concibe como una matriz. Los renglones y las columnas de la matriz proveen un buen marco de referencia para revisar la estrategia, posición y dirección de una idea o proceso. La composición de una matriz FODA se muestra en la tabla 4.1. Lo que indica la matriz FODA es el grado de relación que existe entre cada uno de los factores.

	Amenazas	Oportunidades
Debilidades		
Fortalezas		

Tabla 4.1: Matriz FODA.

En ella se identifican de forma decreciente el grado de influencia de cada factor en cada una de las intersecciones. Esta flexibilidad permite de forma eficiente determinar la factibilidad de un proceso o idea, lo cual ayuda en el desarrollo de la metodología para el análisis de comportamiento de códigos maliciosos. Una vez identificado todos los factores, la interpretación de la matriz FODA está en la tabla 4.2 en la página siguiente.

4.2. Proceso de administración del riesgo

El incluir conceptos del proceso de administración del riesgo en la metodología permite tener certeza de cómo incluir el análisis de códigos maliciosos y principalmente el reporte final de dicho análisis en cualquier programa de seguridad en cómputo de cualquier organización. El proceso de

debilidades y fortalezas son internas a la empresa; las amenazas y oportunidades se presentan en el entorno de la misma. Para la metodología se aplica este análisis enfocado en los códigos maliciosos.

[2]Esto se debe a que existen una mayor proporción de códigos maliciosos programados para esta plataforma que para otros (GNU/Linux, BSD, MacOS, etcétera). Además la plataforma Windows es de las más utilizadas en todo el mundo.

	Amenazas	Oportunidades
Debilidades	**Defender**: Es el peor escenario, se deben tratar de erradicar los factores que perjudican el proceso.	**Atacar**: Existen soluciones para el proceso en cuestión.
Fortalezas	**Sobrevivir**: Las amenazas son eliminadas por las fortalezas del proceso en cuestión.	**Reorientar**: El proceso en cuestión puede concebirse de una manera mas exitosa.

Tabla 4.2: Interpretación de la matriz FODA.

administración del riesgo permite proteger a la organización y su habilidad para cumplir la misión de la organización, no solo sus recursos de TI[3], [Stoneburneri et al., 2002]. Este concepto permite explicar a los administradores de la organización la necesidad de integrar este proceso como una función esencial de administración de la organización.

El riesgo es el impacto negativo neto de una vulnerabilidad, considerando la probabilidad e impacto de ocurrencia[Stoneburneri et al., 2002]. La administración del riesgo es el proceso de identificación de riesgos, evaluación de riesgos y reducción de riesgos a un nivel aceptable[Stoneburneri et al., 2002]; en este enfoque los riesgos que se quieren encontrar son precisamente aquellos relacionados con los códigos maliciosos, esos riesgos podrán ser identificados mediante esta metodología.

- Identificación de riesgos: determinar que vulnerabilidades utiliza el código malicioso para la infección del sistema y/o aplicación.
- Evaluación de riesgos: el código malicioso puede tener un gran impacto, número de sistemas infectados y/o daños causados, en las organizaciones.
- Reducción de riesgos: aunque no es parte del análisis de comportamiento de códigos maliciosos, la información obtenida de dicho análisis es el punto de partida que las organizaciones dedicadas a combatir el software malicioso pueden tomar en cuenta.

4.3. Auditoría informática

La auditoría informática es un proceso muy importante en la seguridad informática, en general se aplica en organizaciones para verificar distintos aspectos establecidos en normas, estándares, buenas prácticas, etcétera, además de que algunos conceptos y tópicos de la auditoría informática serán aplicados en la elaboración de la presente metodología.

[3] Acrónimo para Tecnologías de la Información, concepto que engloba al conjunto de servicios, redes, software, aparatos que tienen como fin el mejoramiento de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.

Los conceptos que son de gran utilidad para esta metodología son:

- Administración de contraseñas: el código malicioso puede hacer uso de contraseñas débiles e incluso el acceso a recursos no protegidos por contraseña. La detección de este tipo de comportamiento depende en gran medida en la configuración del laboratorio de análisis de malware.
- Acceso a la información: el código malicioso busca, basado en su comportamiento, la obtención y/o divulgación de la información sensible del sistema infectado. Para la obtención de esta información el software malicioso puede violar los distintos niveles de acceso establecidos en el sistema, el borrado y/o modificación de bitácoras de acceso a información confidencial.
- Violaciones a controles de acceso: los controles que se tengan implementados en el laboratorio de análisis deben ser, en la medida de lo posible, semejantes a los ambientes en donde el código malicioso se propaga.

4.4. Análisis forense

El análisis forense tiene ciertos mecanismos que pueden ser aplicables para el análisis de comportamiento de códigos maliciosos. El análisis forense consiste en la identificación, recolección, examinación y análisis de datos mientras se preserva la integridad de la información, [Kent et al., 2006]. El propósito del análisis forense puede ser variado e incluye:

- Investigar crímenes.
- Violaciones a las políticas de seguridad de la organización.
- Reconstruir incidentes de seguridad.
- Solución de problemas operacionales.
- Recuperación de daño accidental a un sistema.

El proceso general del análisis forense es:

- **Colección**: identificar, etiquetar, grabar y adquirir datos de las posibles fuentes de datos relevantes, mientras los procedimientos subsecuentes preservan la integridad de los datos.
- **Examinación**: procesar los datos recolectados usando una combinación de métodos manuales y automáticos, y evaluando y extrayendo los datos de interés particular, mientras se preserva la integridad de los datos.
- **Análisis**: analizar los resultados de la fase anterior, usando métodos y técnicas legalmente justificables, para derivar en información útil que responde a los cuestionamientos que son los motivos que generaron la colección y examinación.

- **Reportar**: reportar los resultados del análisis, que podrían incluir una descripción de las acciones usadas, explicar cómo fueron seleccionadas las herramientas y procedimientos, determinar que otras acciones necesitan ser realizadas y proporcionar recomendaciones para mejorar las directivas, herramientas y otros aspectos del análisis forense.

Las técnicas de análisis forense permiten a esta metodología obtener información extra la cual no es posible obtener mediante las herramientas descritas en el capítulo anterior. La importancia del proceso del análisis forense recae en su capacidad para obtener datos extras basados en la colección de evidencia del laboratorio de análisis, la examinación de datos importantes del sistema y finalmente el análisis de esa información extraída. El enfoque del análisis forense de preservar la integridad de los datos es la premisa que se incluye en esta metodología.

4.5. Propuesta de metodología para el análisis de comportamiento de códigos maliciosos

En el capítulo anterior se menciona un esbozo del proceso de análisis de comportamiento de códigos maliciosos.

1. Captura del código malicioso.
2. Identificación del código malicioso.
3. Monitoreo del sistema y/o aplicación sin infectar, para un posterior análisis de resultados.
4. Ejecución del código malicioso.
5. Monitoreo del sistema y/o aplicación durante la ejecución del código malicioso.
6. Obtención de información del monitoreo en el sistema y/o aplicación y recolección de evidencias (rastros dejados por el código malicioso no detectado por el monitoreo).
7. Análisis del estado sin infección del sistema y/o aplicación con la información recolectada en el punto anterior.
8. Realizar un reporte ejecutivo y técnico de la actividad del código malicioso.

Esta metodología pretende mejorar los pasos descritos anteriormente agregando algunos conceptos que se han mencionado en el presente capítulo. En [Mell et al., 2005] se encuentra una excelente referencia de cómo se debe procesar un incidente derivado de algún código malicioso, desafortunadamente no menciona las soluciones posibles a ese incidente. La presente metodología permite aportar información relevante sobre el comportamiento de un código malicioso.

La primera fase es la captura del código malicioso de la cual, su análisis FODA de esta etapa está en la tabla 4.3 en la página siguiente.

	Amenazas	Oportunidades
Debilidades	▪ Se requieren de sensores. ▪ La ubicación de los sensores. ▪ Se depende altamente de las tecnologías honeypot.	▪ Existe un conocimiento pobre y/o erróneo de los códigos maliciosos por parte de los usuarios. ▪ Los usuarios no acostumbran a enviar códigos maliciosos para su análisis.
Fortalezas	▪ Capacidad de programar módulos para honeypot. ▪ Ampliar la ubicación de sensores. ▪ Clasificación pobre de códigos maliciosos.	▪ Determinar variantes de códigos maliciosos. ▪ Comunicación entre las organizaciones dedicadas a combatirlo.

Tabla 4.3: Análisis FODA de la fase de captura del código malicioso.

Es por ello que la fase de captura del código malicioso no corresponde a un análisis de comportamiento de código malicioso. Los distintos medios de obtención del malware no competen al analista y sólo retrasan su actividad principal. Esta fase debe de ser llevada a cabo pero por instancias alternas al propio analista de códigos maliciosos.

La segunda fase comprende la identificación del código malicioso, es un paso primordial del análisis de comportamiento de códigos maliciosos, el identificar la muestra maliciosa permite al analista determinar el proceso de análisis y las herramientas y técnicas a utilizar. Para esta metodología, está compuesta por:

- Determinar el tipo de archivo del código malicioso.

  Verificar la verdadera extensión del código malicioso.

  Inspeccionar la estructura interna del código malicioso, los primeros bytes indican el tipo de archivo.
  En Windows es posible verificar la extensión del archivo deshabilitando la opción en Windows Explorer para mostrar la extensión del archivo para tipos de archivos conocidos.
  En Linux existe una utilería llamada file la cual permite determinar el tipo de archivo de acuerdo al tipo de contenido del archivo y de la llamada al sistema stat.

- Una vez determinado el tipo de código malicioso, se aplica la fase de examinación del aná-

lisis forense, procesarlo usando una combinación de métodos manuales y automáticos, y evaluando y extrayendo los datos de interés particular, mientras se preserva la integridad de los datos.

Determinar la plataforma del tipo de archivo: Windows, Linux, Mac OS, BSD, etcétera.

Determinar la aplicación a la que pertenece el tipo de archivo.

Calcular la firma hash md5, sha, etcétera.

Verificar información disponible en internet para la firma hash, algunos de los sitios comentados en el capítulo anterior como virustotal.com mantienen un registro de códigos maliciosos analizados basados en la firma hash de dicho código. ¿Existen análisis previos realizados a esa muestra?.

Calcular el tamaño del código malicioso.

Resguardar el nombre y tipo de archivo utilizado por el código malicioso de la fase anterior.

- Contar con analizadores de estructuras de tipos de archivos más comunes. Existen herramientas de análisis de códigos maliciosos que trabajan sobre un tipo de archivo en especial. Esta fase determina el conjunto de herramientas que será necesario utilizar para obtener información del comportamiento del código malicioso.

La tercera fase comprende el monitoreo del sistema y/o aplicación sin infectar, para un posterior análisis de resultados. Está compuesta por:

- Habilitar, en la medida de lo posible, con distintos laboratorios de acuerdo con él tipo de archivo y de código malicioso a analizar. La capacidad de contar con laboratorios de análisis de comportamiento de códigos maliciosos acordes a la plataforma(s) a la(s) cual(es) definen su ataque.

- Contar con distintas versiones de las aplicaciones afectadas por un código malicioso. Un código malicioso explota una vulnerabilidad conocida en un sistema para lograr su infección en el mismo, aunque esa vulnerabilidad puede estar restringida a cierta versión del sistema. El contar con múltiples versiones permite al analista identificar y descartar las versiones que no son afectadas por el código malicioso.

- Contar con un inventario de procesos, sockets abiertos, hilos y aplicaciones legalmente instaladas y ejecutadas. La capacidad de contar con información confiable en el laboratorio de análisis le permite al analista filtrar información relevante al momento de la ejecución del código malicioso. El inventario del laboratorio de análisis debe de estar actualizado y en revisión constante debido a las múltiples ejecuciones de códigos maliciosos en el mismo.

En Windows se puede hacer uso de alguno de los programas reseñados en el capítulo anterior por ejemplo process explorer. Para la obtención de la lista de programas instalados se puede verificar el contenido de la llave del registro HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall. El uso del comando netstat permite obtener qué sockets y puertos están abiertos.

En Linux existen diferentes utilerías que permiten obtener los procesos que se están ejecutando como pueden ser ps y top, entre otros. El rango de aplicaciones en este tipo de plataformas es muy extenso, es por ello que el uso de la utilería find es de gran utilidad para determinar qué archivos son ejecutables. El uso del comando netstat es igualmente útil en Linux para determinar qué sockets están abiertos.

- Realizar un inventario del sistema de archivos, es decir, recolectar información relevante del sistema.

  Tiempos MAC. Este tipo de información es de gran utilidad debido a que al mantener un registro de la fecha y hora de la última modificación, el último acceso y cuándo se creó un archivo. En ocasiones el código malicioso hará uso del sistema de archivos para realizar su infección y al realizarlo podrían de manera intencional o no modificar los tiempos MAC de un archivo.

  Tamaño de los archivos.

  Ubicación de los archivos.

  Integridad del sistema de archivos. El mantener un inventario de la integridad del sistema de archivos del laboratorio de análisis de malware, permite analizar en fases posteriores de esta metodología si el código malicioso realizó modificaciones al sistema de archivos. Este tipo de inventarios es posible realizarlo mediante el uso del lenguaje Perl y su función stat, la cual de acuerdo con la plataforma donde se ejecute podrá determinar la información que esta metodología requiere.

La cuarta fase comprende la ejecución del código malicioso, debe integrar los siguientes factores:

- Documentar los requerimientos de ejecución del código malicioso. Ésta es una fase difícil debido a que el código malicioso podría cambiar la forma de comportarse cada vez que se ejecute. Es decir, los requerimientos de ejecución serán variables y no siempre serán posibles de cumplir en el laboratorio de análisis de comportamiento. El código podría no ejecutarse en ambientes virtuales o incluso podría requerir del acceso a información externa para su completa ejecución.

- Garantizar la ejecución del código malicioso. Es difícil determinar si el código malicioso se ejecuta de la forma como lo haría en un equipo víctima. Sin duda el comportamiento del sistema será distinto al normal y las fases posteriores podrían dar un veredicto de la ejecución del código malicioso. Debido a esto el analista debe de al menos cumplir con

aquellos requerimientos, que hasta el momento conoce, que podrían permitir la ejecución del código malicioso.

- Mantener la integridad del código malicioso, para pruebas posteriores. El análisis de comportamiento requiere de múltiples ejecuciones del código malicioso para recopilar suficiente información sobre su actividad. El resguardar la muestra del mismo permite al analista realizar varios análisis sobre el código malicioso.

La quinta fase es el monitoreo del sistema y/o aplicación durante la ejecución del código malicioso, el análisis FODA de esta etapa está en la tabla 4.4.

	Amenazas	Oportunidades
Debilidades	▪ Falta de recursos físicos, técnicos y/o lógicos para el monitoreo de la actividad del código malicioso. ▪ Resultados de monitoreo confusos.	▪ Bitácoras con información no relevante. ▪ Modificaciones de los resultados del monitoreo por parte del código malicioso. ▪ Ausencia de herramientas para el monitoreo debido a la plataforma.
Fortalezas	▪ Herramientas de monitoreo genéricas y específicas.	▪ Se obtiene información documental. ▪ Se conoce la forma de actuar del código malicioso.

Tabla 4.4: Análisis FODA de la fase de monitoreo del sistema y/o aplicación durante la ejecución del código malicioso.

De acuerdo con él análisis FODA se determina que la fase de monitoreo del sistema y/o aplicación durante la ejecución del código malicioso debe integrar los siguientes factores:

- Monitoreo del tráfico de red generado. Mediante el uso de sniffers tales como tcpdump.
- Monitoreo de puertos en el equipo infectado y detección de los procesos y/o servicios que los generan. Herramientas como netstat muestran cuáles puertos están abiertos y qué PID los está ejecutando.
- Medios de conexión permitidos para los puertos abiertos. Realizar pruebas de conexión mediante telnet hacia los puertos abiertos detectados.

- Cuando se analiza un código malicioso para dispositivos maliciosos:
- Es aún prematuro determinar las condiciones y características genéricas para un laboratorio de análisis.
- Se aconseja reproducir el ambiente que permite la infección del dispositivo móvil.
- La recopilación de información depende en gran medida de las habilidades del analista y de la disponibilidad y efectividad de aplicaciones para el análisis de códigos maliciosos para dispositivos móviles.
- Modificaciones al sistema de archivos. El uso de HIDS (por ejemplo tripwire, OSSEC) y NIDS (por ejemplo snort) ayudan a detectar cambios realizados en el sistema de archivos.
- Procesos creados. El monitoreo de los procesos creados en la memoria RAM es fundamental debido a que muchos códigos maliciosos realizan toda su actividad desde la memoria RAM sin utilizar el sistema de archivos, por lo cual la obtención de dichos procesos se vuelve una tarea complicada pero que mediante el uso de herramientas como process explorer, process monitor, top, ps ayudan a determinar qué procesos se ejecutan en ese momento.
- Uso excesivo de recursos compartidos. Por ejemplo el uso del protocolo SMB ampliamente utilizado en redes Windows en donde existan demasiadas peticiones desde el sistema donde se encuentra ejecutando el código malicioso. El uso de sniffers puede ayudar a la detección de este tipo de actividad.
- Identificar procesos nuevos. Al identificar procesos nuevos permite determinar si es generado por el código malicioso y en consecuencia documentarlo. Identificar archivos ejecutables nuevos en el sistema de archivos. Como ya se ha mencionado, un código malicioso tendrá siempre la posibilidad de llevar consigo más de un código malicioso, lo que podría traducirse en nuevos archivos ejecutables con procesos maliciosos.
- Identificar archivos ejecutables nuevos en el sistema de archivos.
- Identificar acceso a archivos que almacenan los hábitos de uso del usuario en el sistema.

  Cookies.

  Favoritos del explorador de internet.

  Aplicaciones instaladas en el equipo.
- Acceso a otros archivos, modificación de tiempos MAC.
- Tal y como se realiza en el análisis forense, se debe contar con una línea del tiempo que establezca la actividad realizada por el código malicioso en el sistema de archivos.

- Determinar, con base en la experiencia, la autenticidad y utilidad de los reportes del monitoreo, debido a que en ocasiones el código malicioso puede presentar un comportamiento incierto o inofensivo. Inclusive existe código malicioso que al detectar ambientes virtuales permite su ejecución pero su comportamiento difiere al que tiene en los sistemas normales de sus víctimas.

La siguiente fase es la obtención de información del monitoreo en el sistema y/o aplicación y recolección de evidencias (rastros dejados por el código malicioso no detectado por el monitoreo), debe constituirse por:

- De acuerdo con la decisión del analista recabar la información del monitoreo y sintetizarla.
- En caso de existir discrepancias en la información del monitoreo, realizar actividades específicas de la auditoría informática y el análisis forense.

  Auditoría informática:

  Determinar la forma de infección del sistema, es decir, identificar la vulnerabilidad aprovechada por el código malicioso.

  Determinar el usuario que utiliza el código malicioso, por ejemplo es el usuario SYSTEM (en plataformas Windows) o el usuario con uid 0 (en plataformas UNIX).

  En el caso de dispositivos e infecciones de red, determinar la ubicación lógica de los diferentes dispositivos de red.

  Determinar si el código malicioso se aprovecha o intenta hacer uso de contraseñas débiles.

  El código malicioso deshabilita o modifica el monitoreo propio del sistema, por ejemplo visor de eventos (en plataformas Windows) o syslog (en plataformas UNIX).

  El código malicioso viola la estratificación de niveles de acceso implementados en el sistema.

  En general cuáles controles implementados en el sistema son violados por el código malicioso.

  Análisis forense:

  Identificar el sistema de archivos: es un aspecto que en muchas ocasiones se pasa por alto, en el caso del análisis forense y de códigos maliciosos. Un sistema de archivos define la forma en que los archivos son nombrados, almacenados, organizados y accedidos en volúmenes lógicos. Aunque los sistemas de archivos tienen diferentes características pero también comparten otras en común. Por principio de cuentas cualquier sistema de archivos usa los conceptos de directorios y archivos para organizar y almacenar los datos. Además cualquier sistema de archivos utiliza alguna estructura de datos para apuntar la ubicación de archivos en el medio de almacenamiento. Por último almacenan cada archivo de datos escrito

al medio de almacenamiento en una o más unidades de asignación de archivo, denominadas comúnmente como clústeres o bloques. Una unidad de asignación de archivo es simplemente un grupo de sectores, que son las unidades mas pequeñas que pueden accederse en el medio de almacenamiento.

Formas de almacenamiento de los archivos: en un sistema de archivos conviven distintos tipos de archivos: los archivos borrados, archivos escondidos, espacio de relleno (cuando el tamaño de un archivo o alguna de sus partes es menor a la unidad de asignación de archivo del sistema de archivos) y el espacio libre.

Tiempos de modificación, acceso y creación de los archivos: comúnmente conocido como tiempos MAC. El tiempo de modificación es el último tiempo que un archivo fue cambiado en cualquier forma, incluyendo cuando un archivo es escrito y cuando es cambiado por otro programa. El tiempo de acceso es el último tiempo en que ocurrió un acceso al archivo (fue abierto, visto, impreso, etcétera). El tiempo de creación, es generalmente el tiempo y fecha en que el archivo fue creado. Sin embargo, cuando un archivo es copiado a un sistema, el tiempo de creación se convertirá al tiempo en que el archivo fue copiado al nuevo sistema. El tiempo de modificación se mantiene intacto. Diferentes tipos de sistemas de archivos pueden almacenar diferentes tipos de tiempos. Por ejemplo, en la plataforma Windows se mantiene el último tiempo de modificación, acceso y creación. En la plataforma UNIX se mantiene el tiempo de modificación, el último cambio de inodo y el último tiempo de acceso.

Localizar y extraer datos de los archivos: encontrar archivos y su contenido es una tarea esencial en el análisis forense y de códigos malicioso, ya que son una prueba fundamental del incidente de seguridad que obligó a realizar el análisis.

Examinar copias de archivos, no los archivos originales: al trabajar de este modo se evita afectar los archivos originales lo que podría llevar a resultados erróneos.

Preservar y verificar la integridad de archivos: al trabajar con copias de archivos se debe tener la certeza de que los archivos (originales y copias) no han perdido su integridad.

Encabezados de archivos y extensiones de archivos: como se mencionó en el capítulo 1 en ocasiones los códigos maliciosos cambian las extensiones de los archivos para burlar al usuario; sin embargo el analista, como ocurre en el análisis forense, no debería asumir que las extensiones de un archivo son precisas. Los analistas pueden identificar el tipo de dato almacenado en muchos archivos observando los encabezados de los archivos, aunque hay que alertar que es posible modificar estos encabezados pero no es tan común.

Contar con un conjunto de herramientas: ese conjunto de herramientas debe tener aplicaciones que permitan realizar inspecciones rápidas de datos además de un análisis en profundidad.

- Del análisis forense se utilizan algunos conceptos que se agregan para esta metodología:

Observar los archivos de configuración del sistema operativo.

System.ini: describe el estado actual del sistema. Este archivo es leído al inicio del sistema, contiene cientos de parámetros de Windows como datos acerca del hardware.

Win.ini: contiene datos acerca del entorno actual (escritorio, fuentes, sonidos, etcétera) y aplicaciones individuales. Con frecuencia, se actualiza mediante un programa de instalación para proveer información para la aplicación cuando se ejecuta.

Boot.ini: contiene opciones de configuración para el arranque de un sistema Windows.

Registro de Windows: es una base de datos que contiene información y configuraciones de todo el hardware, software, usuarios y preferencias del sistema.

Usuarios y grupos: Un grupo de usuarios es un conjunto de cuentas de usuario que tienen en común los mismos derechos de seguridad. Una cuenta de usuario puede ser miembro de más de un grupo. Los dos grupos de usuarios más comunes son el grupo de usuarios estándar y el grupo de administradores, pero hay otros. Una cuenta de usuario a veces se describe de acuerdo con el grupo de usuarios al que pertenece.

Archivos de contraseñas: también conocido como archivo SAM, dicho archivo almacena los nombres de usuarios y hashes de las contraseñas para cada cuenta en el equipo o el dominio si es un controlador de dominio. Este archivo es solo accesible por el usuario SYSTEM y se encuentra en %systemroot%system32\config

Tareas programadas: son aplicaciones que se ejecutan de acuerdo con cierta fecha y hora una o varias veces. Para acceder a la consola de administración de tareas programadas se puede ejecutar taskschd.msc.

Bitácoras: eventos del sistema, auditorías de acceso, eventos de aplicaciones, historia de comandos, archivos accedidos recientemente. Es posible acceder a dichas bitácoras ejecutando eventvwr.msc.

Archivos de aplicación: son aquellos que guardan los parámetros de inicio y uso de un programa.

Archivos de datos: son utilizados para la transacción de información de una aplicación y/o sistema.

Archivos de intercambio: en Windows se usa el archivo pagefile.sys y se usa para almacenar temporalmente datos los cuales son intercambiados entre la memoria RAM y éste, con el fin de disponer de un bloque mas grande de memoria.

Archivos de error (conocidos también como archivos dump): describen los servicios en modo kernel (sistema operativo) y modo usuario (aplicación) al momento de un error, incluye información de controladores y aplicaciones, así como módulos (controles y plug-ins) ejecutándose al momento del fallo.

Archivos de hibernación: es el archivo Hiberfil.sys y que se utiliza para almacenar el contenido de la memoria RAM del sistema al momento de realizar la hibernación.

Archivos temporales: se utilizan para almacenar información de una aplicación, como medio de respaldo o por insuficiencia de espacio en memoria RAM. Dichos archivos pueden aparecer en el disco duro en varios directorios a partir de un carácter de tilde $\sim$ y terminando con una extensión TMP.

Memoria RAM.

Configuración de red.

Conexiones de red.

Procesos en ejecución: se pueden obtener los procesos en ejecución mediante el comando tasklist.

Archivos abiertos: se pueden obtener los archivos abiertos mediante el comando openfiles.

Tiempo del sistema operativo.

La séptima fase es el análisis del estado sin infección del sistema y/o aplicación con la información recolectada de la actividad del código malicioso, el análisis FODA de esta etapa está en la tabla 4.5. De acuerdo al análisis FODA se determina que la fase de análisis del estado sin infección del

	Amenazas	Oportunidades
Debilidades	▪ Abundancia de información de ambos estados. ▪ Redundancia de información. ▪ Información equivoca y/o desactualizada. ▪ Consumo de tiempo considerable.	▪ Conocimiento profundo del sistema y/o aplicación. ▪ Aparición de falsos positivos.
Fortalezas	▪ Identificar diferencias útiles para el análisis. ▪ Desechar información propia de la aplicación.	▪ Análisis consecutivos facilitan la comparación de ambos estados.

Tabla 4.5: Análisis FODA de la fase de análisis de los estados sin infección e infección del sistema y/o aplicación.

sistema y/o aplicación con la información recolectada de la actividad del código malicioso, debe de componerse de:

- La experiencia es la única solución para minimizar los problemas detallados en la matriz FODA.
- Poner gran énfasis en esta etapa para obtener información eficaz que permita la contención y erradicación del código malicioso. La información eficaz es aquella que permite obtener datos sobre:

  Procesos creados por el código malicioso.

  Ciclo de ejecución del código malicioso (qué acciones realizó).

  Conexiones foráneas generadas por el código malicioso.

  Comunicación iniciada por el sistema que alberga al código malicioso hacia la red interna.

  Patrones de infección definidas por el código malicioso, vulnerabilidades aprovechadas por el código.

  Determinar el tipo de plataforma y ambientes que requiere el código malicioso para su infección.

  Información sensible recolectada por el código malicioso.

  Credenciales utilizadas por el código malicioso para conectarse a otros servicios (procesos creados por el software malicioso, conexiones a otros servidores, descarga de nuevo código malicioso).
- El obtener información mínima y/o insuficiente sirve para alertar a otras organizaciones de la peligrosidad del código malicioso, por supuesto esta afirmación debería de ser manejada con cuidado ya que con base en la información generada se actuará en consecuencia.

La octava fase es realizar un reporte ejecutivo y técnico de la actividad del código malicioso, el análisis FODA de esta etapa está en la tabla 4.6 en la página siguiente.

De acuerdo con el análisis FODA se determina que la fase de realización de un reporte ejecutivo y técnico de la actividad del código malicioso, debe componerse de:

- La necesidad de información técnica en el reporte ejecutivo debe de acompañarse por un glosario de términos.
- Evitar en la medida de lo posible la mezcla de términos técnicos, es decir, unificar aquellos que causan confusión en el lector.
- La longitud del reporte ejecutivo no debería de exceder dos cuartillas.

	Amenazas	Oportunidades
Debilidades	▪ El reporte ejecutivo requiere de información técnica esencial. ▪ El reporte ejecutivo debe ser conciso. ▪ Algunos usuarios normales tienden a asociar *cualquier* código malicioso con un virus.	▪ El receptor de los reportes difiere en cada organización. ▪ Los usuarios normales no tienden a leer los reportes ejecutivos.
Fortalezas	▪ El lenguaje utilizado en los reportes puede ser confuso.	▪ Los reportes permiten tomar acciones para la contención y erradicación del código malicioso. ▪ Los reportes son el medio de comunicación entre las organizaciones dedicadas a combatir los códigos maliciosos y la comunidad en general.

Tabla 4.6: Análisis FODA de la fase de reporte ejecutivo y técnico de la actividad del código malicioso.

- La longitud del reporte técnico debe de ser lo mas completo sin obviar información.
- Determinar la audiencia objetivo del reporte ejecutivo.
- Elaborar reportes ejecutivos que muestren los riesgos para la organización y/o usuarios.
- El reporte ejecutivo debe mostrar hechos cuantitativos, es decir, incluir capacidad de infección y propagación del código malicioso.
- El reporte ejecutivo debe de incluir medidas para la prevención del código malicioso.
- La veracidad del reporte ejecutivo y técnico radica en su efectividad para la contención y erradicación del código malicioso.

Finalmente la presente metodología se encuentra en la figura 4.2 en la página siguiente.

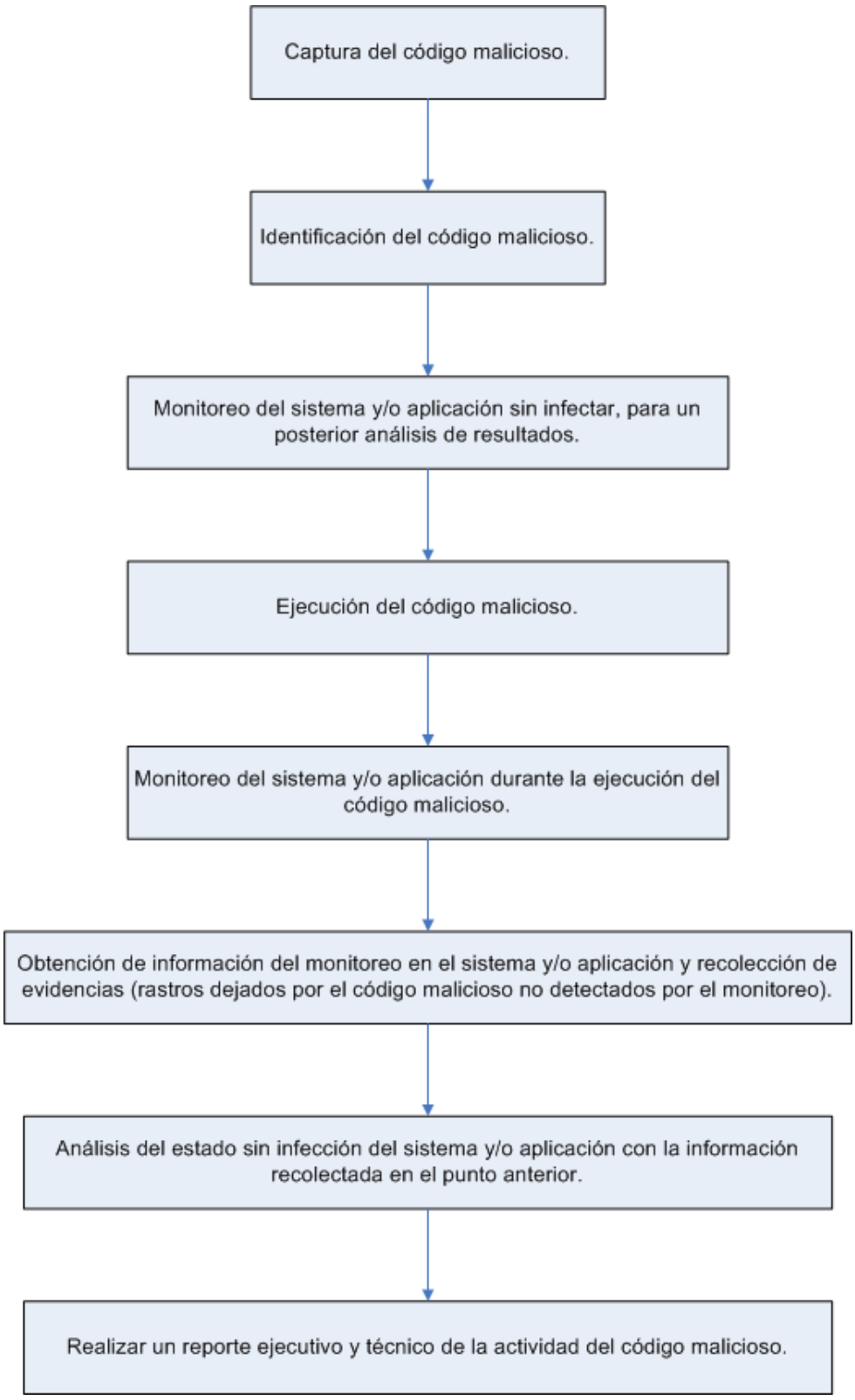

Figura 4.2: Metodología para el análisis de comportamiento de códigos maliciosos.

Conclusiones

En el campo de la seguridad informática existen distintas áreas de desarrollo e investigación. El análisis de códigos maliciosos es una actividad importante en el proceso de atención y manejo de incidentes. Como en cualquier actividad informática se requiere de esquematizar y/o documentar las tareas comunes de dicha actividad. Al analizar un código malicioso surge la necesidad de generar y acumular las distintas técnicas y formas de análisis de tal forma que permita llevar a cabo la importante tarea del análisis de software malicioso.

Debido a la existencia de los códigos maliciosos se ha hecho necesaria la creación de técnicas y herramientas que permitan conocer la forma de actuar y la composición de los códigos maliciosos. Además, algunos códigos cambian en tiempo real su forma de actuar e incluso agregan nuevas funcionalidades, por esa razón el uso del análisis estático no siempre arroja resultados confiables e incluso el tiempo invertido en dicha actividad lo hace en ocasiones poco efectivo.La segunda manera de analizar software malicioso, el análisis dinámico es la forma sucia y fácil del análisis de códigos maliciosos. Aunque el análisis dinámico tiene un inconveniente: analiza sólo una muestra de código malicioso a la vez.

El análisis de comportamiento de códigos maliciosos permite, en un alto porcentaje, determinar la forma de actuar del software malicioso y de cómo generar medidas de contención y erradicación. El analista podrá al finalizar el proceso de análisis obtener una respuesta que determine las pautas de comportamiento del código malicioso analizado. La respuesta ante la aparición de un código malicioso debe ser pronta, efectiva y puntual. Una respuesta pronta depende en gran medida de la complejidad del código malicioso y de la información que exista de él, tal y como sucede en el área de respuesta y manejo de incidentes, se debe contar con procedimientos que permitan mitigar el daño potencial que ocasionaría la muestra de código malicioso. La respuesta efectiva se deriva de las consecuencias benéficas que se tiene de alertar a la organización y/o usuario del potencial daño del código malicioso. Una respuesta puntual surge cuando las advertencias, lineamientos y soluciones surgidas del análisis de códigos maliciosos ayudan a reducir el número de sistemas infectados.

Es por ello que se propone el uso de herramientas de análisis de códigos maliciosos en conjunto con una metodología que permita obtener información precisa de la actividad del código malicioso

así como de la forma de erradicarlo. El siguiente paso es elaborar un reporte detallado de la muestra de código malicioso analizado de forma *rápida* y concisa. Los analistas podrían usar esos reportes para iniciar una respuesta al incidente de seguridad ocasionado por el código malicioso.

Una metodología para el análisis de comportamiento de códigos maliciosos debe obtener el comportamiento relevante del software malicioso, no debería pasar por alto ninguna funcionalidad ejecutada debido a que los analistas usarán dicha información para combatir la amenaza del código. Finalmente la metodología debe permitir realizar un correcto análisis del código malicioso y con la principal directriz de evitar en lo posible los falsos positivos.

Una metodología que abarque el proceso de análisis dinámico de software malicioso es ambiciosa, el cubrir el amplio espectro de plataformas de hardware y de software disponibles actualmente es difícil de llevar a cabo. Esta metodología trata de dar los lineamientos básicos y que sirven de punto de partida para el análisis de un código malicioso.

La complejidad del análisis de comportamiento tiene, como ya se mencionó en este documento, sus ventajas y desventajas; aunque el motivo principal de elaborar una metodología para este tipo de análisis resulta de la capacidad para poder ser llevada a cabo por expertos en el área informática con un mínimo de experiencia en el tratamiento/erradicación de software malicioso. Así mismo la presente metodología se encuentra basada en la experiencia adquirida en el Proyecto Malware[4] del UNAM-CERT[5] durante mi estancia como prestador de servicio social. Además, la metodología expuesta es usada actualmente en el Proyecto Malware.

Esta metodología cubre aspectos generales que pueden utilizarse en cualquier plataforma, como punto de partida; sin embargo se optó por realizarla con un ligero sesgo hacia la plataforma Windows. La razón de esto se debe a un mayor índice de proliferación de códigos maliciosos para dicha plataforma. El gran abanico de aplicaciones que son desarrolladas sobre esa plataforma, hacen que esta metodología tenga una mayor probabilidad de aceptación entre los interesados en el análisis de software malicioso. Del mismo modo que esta metodología da pautas para el análisis dinámico en sistemas Windows, se excluyen plataformas con menor porcentaje de uso comparado con Windows; es esa una de sus grandes debilidades de la metodología elaborada en este documento.

Como todo proceso informático, los cambios tecnológicos pueden hacer que la presente metodología sea perfectible y adaptable a las nuevas funcionalidades de los códigos maliciosos. Para reducir la probabilidad de este hecho, la metodología mantiene aspectos que han sido característicos de los códigos maliciosos a lo largo de su historia, además de considerar las tendencias actuales como

[4] www.malware.unam.mx
[5] www.cert.org.mx

son los códigos maliciosos combinados por ejemplo stormworm.

Un factor que la metodología no cubre es el análisis de comportamiento en dispositivos móviles. En un futuro no muy lejano se tendrán mayores amenazas de este tipo debido a su uso masivo. Es un área con un gran potencial de desarrollo en la investigación de software malicioso para dispositivos móviles e incluso con repercusión a nivel nacional con la mucho mayor penetración que tienen estos dispositivos entre los consumidores mexicanos.

La gran proliferación de códigos maliciosos, con sus variantes, obliga a entender la forma de actuar de dicho software así como también el definir cómo realiza su infección y propagación. Con la presente metodología se da respuesta a dichas interrogantes y que permiten, basados en el reporte del último paso de la metodología, tener un punto de partida para la contención y erradicación del código malicioso.

Anexo 1. Proceso de administración del riesgo

El incluir conceptos del proceso de administración del riesgo en la metodología permite tener certeza de cómo incluir el análisis de códigos maliciosos y principalmente el reporte final de dicho análisis en cualquier programa de seguridad en cómputo de cualquier organización. El proceso de administración del riesgo permite proteger a la organización y su habilidad para cumplir la misión de la organización, no sólo sus recursos de TI[6], [Stoneburneri et al., 2002]. Este concepto permite explicar a los administradores de la organización la necesidad de integrar este proceso como una función esencial de administración de la organización.

El riesgo es el impacto negativo neto de una vulnerabilidad, considerando la probabilidad e impacto de ocurrencia[Stoneburneri et al., 2002]. La administración del riesgo es el proceso de identificación de riesgos, evaluación de riesgos y reducción de riesgos a un nivel aceptable[Stoneburneri et al., 2002]; en este enfoque los riesgos que se quieren mitigar son precisamente aquellos relacionados con los códigos maliciosos, esos riesgos podrán ser identificados mediante esta metodología. La administración del riesgo se compone de tres procesos:

- Cálculo del riesgo.
- Reducción del riesgo.
- Evaluación y cálculo.

En cada etapa de la administración del riesgo se refleja la relación con el análisis de códigos maliciosos. El mismo concepto de administración del riesgo nos permite entender esta relación; la administración del riesgo es el proceso que permite a los administradores de TI balancear los costos operativos y económicos de medidas de protección y lograr ganancias en capacidad de misión protegiendo los sistemas de TI y los datos que apoyan las misiones de las organizaciones.

[6] Acrónimo para Tecnologías de la Información, concepto que engloba al conjunto de servicios, redes, software, aparatos que tienen como fin el mejoramiento de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario.

En el cálculo del riesgo se determina la probabilidad de un evento futuro adverso, se deben de analizar las amenazas de un sistema TI en conjunto con las potenciales vulnerabilidades y los controles existentes en el sistema TI. Además se toma en cuenta el impacto, es decir, la magnitud de daño que pudo ser causado por una amenaza debida a una o varias vulnerabilidades.

La metodología del cálculo del riesgo está compuesta por nueve pasos:

- **Caracterización del sistema.**
  Como su nombre lo indica, en esta fase se identifica al sistema TI, los recursos y la información que constituye el sistema. Además en esta fase se conoce el alcance del cálculo del riesgo, delinea los límites de autorización operacional y provee información esencial para definir el riesgo.

- **Identificación de la amenaza.**
  Una amenaza es el potencial de una particular fuente de amenaza de ejecutar exitosamente una vulnerabilidad particular. Una vulnerabilidad es una debilidad que puede ser accidentalmente lanzada o intencionalmente explotada. Es importante mencionar que una fuente de amenaza no representa un riesgo cuando no hay vulnerabilidad que pueda ser ejecutada. Se identifican las fuentes de amenazas potenciales del sistema TI que se evalúa.

- **Identificación de la vulnerabilidad.**
  Se desarrolla una lista de las vulnerabilidades del sistema que puedan ser explotadas por potenciales fuentes de amenazas. Una vulnerabilidad es un defecto o debilidad en los procedimientos de seguridad, diseño, implementación o controles internos de un sistema que podrían ser ejecutadas (accidentalmente lanzadas o intencionalmente explotadas) y resulta en una ruptura de seguridad o violación de la directiva de seguridad del sistema. Para identificar vulnerabilidades se pueden utilizar:

    Fuentes de vulnerabilidades:

    Documentación previa de cálculos de riesgo realizados al sistema.

    Reportes de: auditoría del sistema, anomalías del sistema, revisiones de seguridad y pruebas y evaluaciones del sistema.

    Listas de vulnerabilidad.

    Consejos de seguridad de las compañías de TI.

    CERTs y listas de correo y/o posts (SecurityFocus.com, secunia, etcétera)

    Análisis de software de seguridad.

    Ejecutar pruebas de seguridad al sistema en cuestión:

    Herramientas de escaneo de vulnerabilidades automatizado.

    Pruebas y evaluación de seguridad.

Pruebas de penetración.

Desarrollo de una lista de control de los requerimientos de seguridad.

- **Análisis de control.**
  Se analizan los controles que han sido implementados o están planeados para su implementación por la organización para minimizar o eliminar la probabilidad de una amenaza atacando una vulnerabilidad del sistema.

- **Determinación de la probabilidad.**
  Se consideran los siguientes factores:

  Motivación y capacidad de la fuente de amenaza.

  Naturaleza de la vulnerabilidad.

  Existencia y efectividad de controles actuales. La probabilidad de que una potencial vulnerabilidad podría ser ejecutada por una fuente de amenaza dada puede describirse como:

  Alta: la fuente de amenaza es altamente motivada y suficientemente capaz y los controles para prevenir la vulnerabilidad de ser ejecutada son inefectivos.

  Media: la fuente de amenaza es motivada y capaz, pero los controles están en el lugar que pueden impedir la ejecución exitosa de la vulnerabilidad.

  Baja: la fuente de amenaza carece de motivación o capacidad, o los controles en el lugar previenen o al menos impiden significativamente de ser ejecutada.

- **Determinación del riesgo.**
  Para garantizar una efectiva determinación del riesgo es necesario contar con la siguiente información:

  La misión del sistema, es decir, cuál era su rol en la organización.

  Los datos críticos del sistema.

  La sensibilidad de los datos del sistema. En consecuencia el impacto adverso de un evento de seguridad puede describirse en términos de la pérdida o degradación de una o varias de las metas de la seguridad: integridad, disponibilidad y confidencialidad.

Pérdida de integridad. La integridad del sistema y los datos se refiere al requerimiento de que la información sea protegida de modificaciones impropias. La integridad es la pérdida si se realizan cambios sin autorización a los datos o al sistema TI debido a actos intencionales o accidentales. Si la pérdida de la integridad del sistema o los datos no es corregida, el uso continuo del sistema contaminado o corrupto podría resultar en imprecisiones, fraude o decisiones erróneas. Además, la violación de integridad puede ser la primera etapa de un ataque exitoso contra la disponibilidad o confidencialidad del sistema. Es por ello que la

pérdida de integridad reduce la seguridad de un sistema.

Pérdida de disponibilidad. Si un sistema no está disponible a sus usuarios finales, la actividad de la organización se puede ver afectada. La pérdida de funcionalidad de un sistema y la efectividad operacional.

Pérdida de confidencialidad. La confidencialidad de un sistema o datos se refiere a la protección de la información de acceso no autorizado. El acceso no autorizado, no anticipado o no intencional puede resultar en la pérdida de confianza, vergüenza o acción legal contra la organización.
Para calcular la determinación del riesgo es posible utilizar una matriz de evaluación del riesgo [Stoneburneri et al., 2002], ver tabla 7. Aunque su uso es subjetivo es una herramienta que

Probabilidad Amenaza	Impacto Bajo (10)	Impacto Medio (50)	Impacto Alto (100)
Alta (1.0)	Baja 10x1.0=10	Media 50x1.0=50	Alto 100x1.0=100
Media (1.0)	Baja 10x0.5=5	Media 50x0.5=25	Media 100x0.5=50
Baja (0.1)	Baja 10x0.1=1	Baja 50x0.1=5	Baja 100x0.1=10

Tabla 7: Tabla de determinación del riesgo.

permite obtener los niveles de riesgo totales.

- La probabilidad asignada para cada nivel de amenaza es 1.0 para alto, 0.5 para medio y 0.1 para bajo.
- El valor asignado para cada nivel de impacto es 100 para alto, 50 para medio y 10 para bajo.

De acuerdo con esto es posible obtener el nivel de riesgo, considerando un nivel alto (>50 a 100), nivel medio (>10 a 50) y nivel bajo (1 a 10).

- **Recomendaciones de control.**
  En esta fase se indican aquellos factores que deberían ser considerados por la organización para minimizar o eliminar los riesgos identificados.

  - Efectividad de opciones recomendadas, por ejemplo ¿Se cuenta con los recursos de cómputo necesarios?.
  - Legislación y regulación.
  - Políticas de la organización.

- Impacto operacional.
- Seguridad y confiabilidad.

No debe olvidarse que no todos los posibles controles recomendados pueden ser implementados para reducir las pérdidas. Para determinar cuáles controles son requeridos y apropiados para cierta organización se debe de contar con un análisis de costo-beneficio.

- **Documentación de resultados.**
  Una vez finalizado el proceso es necesario tenerlo documentado en un reporte oficial o un resumen. Esto servirá para futuras referencias y como pruebas testimoniales del proceso y las acciones realizadas. La documentación de resultados describe las amenazas y vulnerabilidades, mide el riesgo y provee recomendaciones para la implementación de controles.

La mitigación del riesgo se compone de priorizar, evaluar e implementar los controles recomendados de reducción del riesgo apropiados. Existen diferentes opciones para mitigar el riesgo entre ellas están:

- Asumir el riesgo: se acepta el riesgo potencial y se continúa con las operaciones de la organización o se implementan controles para reducir el riesgo a un nivel aceptable.
- Evitar el riesgo: se elimina la cause del riesgo y/o las consecuencias del riesgo.
- Limitar el riesgo: se implementan controles que minimicen el impacto adverso de un ataque.
- Planear el riesgo: se cuentan con un plan que prioriza, implementa y mantiene los controles.
- Investigación y aceptación: se conoce la vulnerabilidad y se investigan los controles para corregir dicha vulnerabilidad.
- Transferir el riesgo: usar otras opciones para compensar el riesgo, por ejemplo comprar un seguro.

Anexo 2. Auditoría informática

Existen dos clases de auditorías, la interna y la externa. Una auditoría consiste de la evaluación de los procesos del sistema y controles de una organización y es elaborado siguiendo uno o varios estándares o procesos documentados [Wright, 2007]. Las auditorías se diseñan para proveer una valoración independiente a través de la prueba y evaluación de una serie de representaciones de un sistema o proceso. Una auditoría puede además proveer un análisis de la efectividad operativa de los controles internos. Una auditoría debe seguir un riguroso programa.

Una auditoría difiere de una inspección en que una auditoría hace representaciones de resultados y/o desempeños pasados. Una inspección evalúa resultados actuales. Para que una auditoría sea válida, debe ser conducida de acuerdo con los principios aceptados [Wright, 2007]. Una parte que componen a la auditoría informática es la auditoría de seguridad, en la cual se valoran aspectos relacionados con los sistemas operativos, software comercial, de comunicaciones, de bases de datos, de proceso, de aplicaciones e indudablemente las instalaciones y aspectos físicos. Algunos puntos esenciales en la auditoría de seguridad informática son de acuerdo con [Herrera, 2008]:

- **Organizacional**

  Determinar si las responsabilidades relativas a la seguridad física y lógica se encuentran debidamente asignadas.

  Revisar las normas, políticas y procedimientos dictados para garantizar la seguridad de los activos de la organización (equipo, software e información) y determinar si éstos definen claramente las responsabilidades de los usuarios, administradores y personal en general.

  Determinar si las tareas asignadas garantizan la seguridad de los activos y si son difundidas y entendidas en forma correcta. Verificar asimismo, que éstas sean consistentes con las políticas de seguridad comúnmente aceptadas.

  Determinar el grado de conocimiento general de los usuarios sobre la importancia de la seguridad física y lógica de los activos para su adecuada salvaguarda.

  Revisar el organigrama para determinar si existe una área responsable de la administración de la seguridad e información.

- **Seguridad física**

Ubicación física de las instalaciones.

Determinar si la ubicación del centro de cómputo es adecuada para proteger la integridad de los activos de la organización.

- **Acceso a las Instalaciones**

  Determinar si las medidas de seguridad implementadas para regular el acceso a las instalaciones son adecuadas.

  Verificar que existan procedimientos para regular el acceso a las instalaciones.

  Verificar que el acceso al equipo del centro de cómputo sólo lo realice el personal autorizado.

- **Administración de contraseñas**

  Garantizar que el acceso lógico al equipo esté restringido por procedimientos y políticas de acceso utilizando contraseñas.

  Verificar que existan procedimientos autorizados para la asignación y actualización de las claves de acceso a los equipos.

  Solicitar al administrador de los equipos las listas, tablas o matrices de las claves de acceso y verificar si se incluyen las claves de acceso a dispositivos periféricos.

  Revisar los procedimientos para la administración de contraseñas.

  Corroborar que el personal (operativo y usuario) está consciente de los riesgos y problemas derivados de divulgar su clave de acceso.

  Verificar la aplicación de las sanciones especificadas en los procedimientos para la divulgación y mal uso de las contraseñas.

  Determinar si los usuarios son restringidos a terminales específicas o a días y horarios específicos.

  Verificar que el acceso al equipo de monitoreo es controlado.

  Determinar si el departamento del usuario valida periódicamente los permisos, alcance y estratificación de las claves de acceso asignadas a cada uno de sus miembros.

  Comprobar que durante el proceso de información crítica se efectúan comprobaciones periódicas que permitan certificar la identidad y permanencia del usuario, mediante información que sólo él conozca.

- **Acceso a la Información**

  Asegurar que el acceso a la información está restringido con la adecuada estratificación de niveles de acceso.

  Determinar si los procedimientos prevén que las claves asignadas a los usuarios consideren el nivel de acceso para:

Equipos

Archivos

Programas de las aplicaciones

Comandos del sistema operativo, etcétera.

Verificar la existencia de la documentación mediante la cual se justificaron las asignaciones de claves de acceso a los equipos e información.

Verificar que existan procedimientos para la asignación de claves de acceso temporal o de emergencia.

Determinar si es necesario obtener una autorización especial para este tipo de accesos y si este tipo de autorizaciones se limitan a un periodo dado y si se informa de ello a la administración. Verificar que estos accesos temporales se concedan con poca frecuencia.

Implementar autorizaciones a diferentes niveles con la finalidad de proteger adecuadamente la integridad y confidencialidad de datos y programas.

Verificar que existan procedimientos para la asignación de perfiles de acceso a los sistemas.

Verificar que exista una bitácora automatizada, en la que se registren:

Todos los intentos de acceso al sistema, válidos e inválidos.

Todos los requerimientos hechos al sistema para respaldar programas, datos, o transacciones.

Todas las modificaciones de datos críticos o programas.

Verificar que existan procedimientos para detectar las posibles violaciones.

Verificar que la seguridad de la bitácora está protegida.

Verificar que periódicamente se revise la bitácora por el supervisor indicado.

El propósito de una auditoría de seguridad informática es el prevenir mediante la verificación de los controles implementados por la organización que se cumplan las metas de la seguridad informática y con ello se protejan los activos de la organización. El no realizar al menos auditorías internas puede ocasionar que la organización sea más propensa a ataques por códigos maliciosos, basta recordar que algunos gusanos logran éxito en su propagación debido a contraseñas débiles (que se pudieron evitar con la aplicación de una auditoría de seguridad informática) en los recursos compartidos en equipos Windows.

Anexo 3. Manejo y respuesta a incidentes

Existe en [Mell et al., 2005] una guía muy completa de la integración del manejo y respuesta a incidentes relacionado con los códigos maliciosos. En general el proceso del manejo y respuesta de incidentes de códigos maliciosos es el siguiente:

- **Preparación**: cualquier organización debe tener medidas que aseguren que pueden atender efectivamente a incidentes ocasionados por códigos maliciosos. Algunas recomendaciones incluyen:

  Desarrollar directivas y procedimientos para el manejo de incidentes de códigos maliciosos.

  Realizar de manera regular entrenamiento y ejercicios orientados a códigos maliciosos.

  Diseñar un grupo de trabajo que sea responsable de coordinar las respuestas a incidentes de códigos maliciosos.

  Establecer varios mecanismos de comunicación para que la coordinación entre los manejadores de incidentes, staff técnico, administración y usuarios pueda sostenerse durante eventos adversos.

- **Detección y análisis**: la clave es la capacidad de detectar y validar los incidentes de códigos maliciosos rápidamente debido a que las infecciones pueden propagarse en cuestión de minutos, basta recordar SQL Slammer. Algunas acciones recomendadas son:

  Monitorear las alertas de los códigos maliciosos para identificar y evitar incidentes de software malicioso.

  Revisar datos de incidentes de códigos maliciosos, por ejemplo reportes de usuarios, del equipo de TI y controles técnicos para identificar actividad maliciosa.

  Tener a la mano un conjunto de herramientas para el análisis y erradicación de códigos maliciosos actualizadas.

  Establecer un conjunto de criterios de prioridades que identifiquen el nivel apropiado de respuesta a varios incidentes relacionados con los códigos maliciosos.

- **Contención**: esta fase está compuesta de: detener la propagación del malware y prevenir mayores daños a los sistemas. El éxito de la contención de los códigos maliciosos depende de la variedad de estrategias (métodos) para una situación particular. Algunas recomendaciones de contención incluyen:

  Puede ser útil proveer a los usuarios con instrucciones en cómo identificar infecciones y qué medidas tomar si un sistema es infectado, pero no siempre se debe de confiar totalmente en los usuarios para contener incidentes de códigos maliciosos.

  Si el código malicioso no puede ser identificado y detenido mediante suites de software de seguridad, las organizaciones deberían estar preparados para usar otras herramientas de seguridad para contenerlo. Las organizaciones deberían estar preparadas para enviar copias del o los códigos maliciosos desconocidos a las compañías de software de seguridad para su análisis, organizaciones de respuesta a incidentes y vendedores antivirus; esto con el propósito de tener una guía para manejar nuevas amenazas.

  Las organizaciones deberían estar preparadas para apagar o bloquear servicios usados por el código malicioso para contener el incidente y tomar conciencia de las consecuencias de hacerlo. Las organizaciones deberían estar preparadas para responder a problemas causados por otras organizaciones deshabilitando sus propios servicios en respuesta al incidente del código malicioso.

  Las organizaciones deberían estar preparadas para colocar restricciones adicionales temporales en la conectividad de red para contener el incidente del código malicioso (por ejemplo los bots y los gusanos); reconocer el impacto que dichas restricciones pueden tener en las funciones de la organización.

  Un aspecto fundamental es identificar los equipos infectados por los códigos maliciosos como un medio de contención. Aunque identificar los equipos infectados es en ocasiones complicado debido a la naturaleza dinámica del cómputo.

- **Erradicación**: cualquier organización responderá afirmativamente a la erradicación del código malicioso de los sistemas infectados. Las organizaciones deberían estar preparadas para usar varias combinaciones de técnicas de erradicación simultáneamente para distintas situaciones.

- **Recuperación**: los dos aspectos principales de recuperación de incidentes de códigos maliciosos son restaurar la funcionalidad y los datos de los sistemas infectados y el levantamiento temporal de las medidas de contención. Las organizaciones deberían considerar cuidadosamente las posibles peores situaciones y determinar la forma como debe realizarse la recuperación.

- **Posterior al incidente**: un incidente de código malicioso puede ser extremadamente caro. Lo fundamental es aprender de experiencias propias y/o conocidas en un incidente de software

malicioso, entre ellas están mejorar las capacidades de manejo y defensas contra los códigos maliciosos, identificar los cambios necesarios a las políticas de seguridad, configuraciones de software, detección y prevención de software malicioso.

Las organizaciones deberían establecer capacidades de prevención y manejo de incidentes de códigos maliciosos que les permitan mitigar y eliminar las amenazas presentes y futuras.

Glosario

API Interfaz de programación de aplicaciones. Conjunto de funciones y procedimientos que permite la comunicación entre componentes de software.
Atacante persona que trata de introducirse a un sistema sin autorización y con la intención de realizar algún tipo de daño u obtener un beneficio.
Bot surge de la palabra robot debido a la similitud que tienen con dichos dispositivos. Es un programa que contiene instrucciones para actuar en forma independiente, pudiendo realizar una diversidad de comandos o acciones en forma automática o hasta controlada en forma remota.
Confidencialidad se refiere a que la información solo puede ser conocida por individuos autorizados.
Disponibilidad se refiere a la seguridad que la información pueda ser recuperada en el momento que se necesite, esto es, evitar su pérdida o bloqueo, bien sea por ataque doloso, mala operación accidental o situaciones fortuitas o de fuerza mayor.
DLL Biblioteca de enlace dinámico que son utilizadas a petición del programa por parte del sistema operativo Windows.
DNS Domain Name System abreviatura para Sistema de nombres de dominio, un sistema para asignar nombres a equipos y servicios de red que se organiza en una jerarquía de dominios.
DoS negación de servicio, ataque hacia un equipo con el objetivo de deshabilitar los servicios ofrecidos por la víctima.
DoS negación de servicio, ataque hacia un equipo con el objetivo de deshabilitar los servicios ofrecidos por la víctima.
DDoS negación de servicio distribuido, ataque realizado desde la red mediante el cual un conjunto de equipos envían un ataque dirigido al mismo tiempo a un equipo con el fin de bloquearlo y así negar el o los servicio(s) ofrecidos.
GUI Interfaz gráfica de usuario.
Hash es una función o método para generar claves o llaves que representen de manera casi unívoca a un archivo.
HIDS acrónimo en inglés de sistema de detección de intrusos en un *host*. Busca detectar anomalías que indican un riesgo potencial, revisando las actividades en el equipo. Puede tomar medidas protectoras.
Honeypot es una trampa para detectar, desviar o contrarrestar de alguna manera, los intentos de

uso no autorizado de los sistemas de información.
Host dispositivo de la red que ofrece servicios a otras computadoras conectadas a dicha red.
HTTP HyperText Transfer Protocol es el protocolo de transferencia de hipertexto usado en cada transacción de la Web.
Ingeniería social es la manipulación de un usuario legítimo con el objetivo de obtener información confidencial y/o realizar alguna acción indebida.
Inodo es un conjunto de datos en lo que se refiere a ciertas características de un archivo, tales como el conjunto de permisos y el dueño del archivo.
Integridad se refiere a la seguridad de que una información no ha sido alterada, borrada, reordenada, copiada, etcétera, bien durante el proceso de transmisión o en su propio equipo de origen.
IRC Internet Relay Chat es un protocolo de comunicación en tiempo real basado en texto, que permite debates en grupo o entre dos personas.
Malware es la contracción de malicious software en inglés.
man-in-the-middle interceptar la comunicación entre dos nodos, escuchando el medio de transmisión.
md5 es un algoritmo de reducción criptográfico de 128 bits.
NIDS acrónimo en inglés de sistema de detección de intrusos en una red. Busca detectar anomalías que inicien un riesgo potencial, tales como ataques de denegación de servicio, escaneadores de puertos o intentos de entrar en un equipo, analizando el tráfico en la red en tiempo real.
Payload son el o los efectos nocivos e incluso irreparables, que ocasionan los códigos maliciosos.
PDA acrónimo en inglés de asistente digital personal.
PE formato ejecutable portátil es un formato de archivo para ejecutables, código objeto y DLLs usado en sistemas operativos Windows.
Pharming redirigir la petición de un sitio web hacia otro sitio web, normalmente malicioso.
Phishing scam actividad maliciosa que intenta obtener información sensible de la víctima haciéndose pasar como una entidad confiable a través de un medio electrónico.
PID número que se asocia con un proceso para identificarlo en el sistema operativo.
Pop-up ventanas emergentes que son mostradas generalmente sin que el usuario lo solicite.
Rootkit es un programa (o combinación de varios programas) diseñados para tomar el control de un sistema de equipo de cómputo, sin autorización de el usuario y administrador del sistema.
Screenshot es una imagen tomada por la computadora para grabar los objetos visibles en el monitor o en otro dispositivo de salida visual.
sha acrónimo en inglés de Algoritmo de Hash Seguro, existen distintas versiones como SHA-1, SHA-224, SHA-256, SHA-384, y SHA-512.
SMB Server Message Block es un protocolo de red que permite compartir archivos e impresoras (entre otras cosas) entre nodos de una red.
Spyware contracción del inglés spy + software. También conocido como software espía.
Spam envío de correo masivo a distintas cuentas de correo electrónico, normalmente su contenido

es de índole comercial.

Spammer individuo que envía spam.

TCP/IP conjunto de reglas que permite a las computadoras comunicarse en una red. Define protocolos para diferentes tipos de comunicaciones entre computadoras.

TI acrónimo para Tecnologías de la Información, que agrupa un conjunto de sistemas de cómputo para administrar y procesar la información.

UID acrónimo en inglés de identificación del usuario, es un número entero que asocia un nombre de usuario con dicho número; el uid es único.

Zombie denominación que recibe el equipo o la red bajo control de un atacante.

Bibliografía

[Arnold et al., 2005] Arnold, B., Chess, D., Morar, J., Segal, A., and Swimmer, M. (2005). An environment for controlled worm replication and analysis. *IBM SciPapers*.

[Ashburn et al., 2004] Ashburn, M. W., Sulcoski, M., and Lach, J. (2004). Netss: A networked environment for testing suspicious software. *Proceedings of the 2004 Systems and Information Engineering Design Symposium*.

[Bhattarakosol and Suttichaya, 2007] Bhattarakosol, P. and Suttichaya, V. (2007). Multiple equivalent scale scan: An enhancing technique for malware detection. *Second International COnference on Systems and Networks Communications*.

[Borghello, 2006] Borghello, C. (27 de octubre de 2006). La historia del malware. *Eset*.

[CAIDA, 2008] CAIDA (2008). The spread of the sapphire/slammer worm. http://www.caida.org/publications/papers/2003/sapphire/sapphire.html.

[Carpenter et al., 2007] Carpenter, M., Liston, T., and Skoudis, E. (2007). Hiding virtualization from attackers and malware. *IEEE Security & Privacy*.

[Carvey, 2005] Carvey, H. (2005). Malware analysis for windows administrators. *Digital Investigation*.

[Chen and Ji, 2005] Chen, Z. and Ji, C. (2005). Spatial-temporal modeling of malware propagation in networks. *IEEE Transactions on neural networks*.

[Commission, 2005] Commission, F. T. (Julio 2005). Ftc consumer alert. http://www.ftc.gov/bcp/conline/pubs/alerts/spywarealrt.shtm.

[Dai and Kuo, 2007] Dai, S.-Y. and Kuo, S.-Y. (2007). Mapmon: A host-based malware detection tool. *13th IEEE International Symposium on Pacific Rim Dependable Computing*.

[Distler, 2007] Distler, D. (14 de diciembre de 2007). Malware analysis: An introduction. *SANS*.

[F-secure, 2007a] F-secure (4 de diciembre de 2007a). F-secure reports amount of malware grew by 100% during 2007. http://www.f-secure.com/f-secure/pressroom/news/fsnews_20080331_1_eng.html.

[F-secure, 2007b] F-secure (4 de diciembre de 2007b). F-secure reports amount of malware grew by 100 % during 2007. http://www.f-secure.com/f-secure/pressroom/news/fs_news_20071204_1_eng.html.

[Farmer and Venema, 2005] Farmer, D. and Venema, W. (2005). *Forensic Discovery*. Addison-Wesley.

[Fuentes, 2008] Fuentes, L. F. (10 de abril de 2008). Malware una amenaza de internet. *Revista Digital Universitaria*.

[Garetto et al., 2003] Garetto, M., Gong, W., and Towsley, D. (2003). Modeling malware spreading dynamics. *IEEE*.

[Garuba et al., 2008] Garuba, M., Liu, C., and Washington, N. (2008). A comparative analysis of anti-malware software, patch management, and host-based firewalls in preventing malware infections on client computers. *Fifth International Conference on Information Technology: New Generations*.

[Harley and Lee, 2007] Harley, D. and Lee, A. (22 de marzo de 2007). Análisis heurístico: detectando malware desconocido. *Eset*.

[Hernández, 1999] Hernández, A. H. (1999). *Virus Informático*. DGSCA, UNAM.

[Herrera, 2008] Herrera, Z. Z. (1 de agosto de 2008). Auditoría de seguridad. http://www.isaca.org.mx/CGI-BIN/isaca/mambo451/index.php?option=content\&task=view\&id=50\&Itemid=2.

[Hsu et al., 2006] Hsu, F., Chen, H., Rsitenpart, T., Li, J., and Su, Z. (2006). Back to the future: A framework for automatic malware removal and system repair. In *Proceedings of the 22nd Annual Computer Security Applications Conference*.

[Hypponen, 2008] Hypponen, M. (2008). Crime online and online crime. *F-Secure*.

[Jensen, 2008] Jensen, J. (2008). A novel testbed for detection of malicious software functionality. *The Third International Conference on Availability, Reliability and Security*.

[Kaspersky Labs., 2008] Kaspersky Labs. (2 de junio de 2008). Virus encyclopedia. http://www.viruslist.com/en/viruses/encyclopedia.

[Kent et al., 2006] Kent, K., Chevalier, S., Grance, T., and Dang, H. (2006). *Guide to Integrating Forensic Techniques into Incident Response*. National Institute of Standards and Technology.

[King et al., 2006] King, S. T., Chen, P. M., Wang, Y.-M., Verbowski, C., Wang, H. J., and Lorch, J. R. (2006). Subvirt: Implementing malware with virtual machines. *IEEE Symposium on Security and Privacy*.

[Koch, 2007] Koch, C. (2 de junio de 2007). A brief history of malware and cybercrime. www.cio.com.

[Llewellyn-Jones et al., 2005] Llewellyn-Jones, D., Merabti, M., Shi, Q., and Askwith, B. (2005). Short paper: Harnessing emergent ubiquitous computing properties to prevent malicious code propagation. In *Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*.

[Martignoni et al., 2007] Martignoni, L., Christodorescu, M., and Jha, S. (2007). Omniunpack: Fast, generic, and safe unpacking of malware. *23rd Annual Computer Security Applications Conference*.

[Mary Landesman, 2006] Mary Landesman (10 de enero de 2006). Top ten malware events 2005. http://antivirus.about.com/od/virusdescriptions/a/2005virus.htm.

[Masood, 2004] Masood, S. G. (20 de mayo de 2004). Malware analysis for administrators. http://www.securityfocus.com/infocus/1780.

[McAfee, 2005] McAfee (2005). A brief history of malware an educational note for service providers. *McAfee System Protection Solutions*.

[McGraw and Morriset, 2000] McGraw, G. and Morriset, G. (Septiembre Octubre 2000). Attacking malicious code: A report to the infosec research council. *IEEE Software*.

[McRee, 2007] McRee, R. (2007). Malcode analysis techniques for incident handlers. *Secureworld Expo 2007*.

[Mell et al., 2005] Mell, P., Kent, K., and Nusbaum, J. (2005). *Guide to Malware Incident Prevention and Handling*. National Institute of Standards and Technology.

[Microsoft, 2008] Microsoft (2008). Microsoft security intelligence report home and small business edition. *Microsoft Corporation*.

[Microsoft, 2006] Microsoft (23 de octubre de 2006). What is spyware? http://www.microsoft.com/protect/computer/basics/spyware.mspx.

[Moser et al., 2007] Moser, A., Kruegel, C., and Kirda, E. (2007). Exploring multiple execution paths for malware analysis. *IEEE Symposium on Security and Privacy*.

[Moskovitch et al., 2007] Moskovitch, R., Nissim, N., and Elovici, Y. (2007). Malicious code detection and acquisition using active learning. *IEEE*.

[mwcollect, 2008] mwcollect (2008). Malware collection made easy. http://www.mwcollect.org/.

[National Bureau of Standards, 1997] National Bureau of Standards (1997). Data encryption standard. *FIPS Publication*.

[Nicol, 2005] Nicol, D. M. (2005). Modeling and simulation in security evaluation. *IEEE Security & Privacy*.

[Norman, 2008] Norman (6 de julio de 2008). Norman sandbox analyzer. http://www.norman.com/microsites/malwareanalyzer/Products/analyzer.

[of Homeland Security et al., 2006] of Homeland Security, U. D., Council, S. I. I. T. T., and the Anti-Phishing Working Group. (Octubre 2006). The Crimeware Landscape: Malware, Phishing, Identity Theft and Beyond. *Anti-Phishing Working Group*.

[Perriot, 2002] Perriot, F. (Diciembre de 2002). Virus bulletin. http://www.virusbtn.com/resources/viruses/indepth/opaserv.xml.

[Pinilla, 2006] Pinilla, V. D. (2006). Apuntes de temas selectos de filosofía y de la ciencia y la tecnología. *Facultad de Ingeniería, UNAM*.

[Richardson, 2008] Richardson, R. (2008). Csi computer crime and security survey. *CSI/FBI*.

[Serrano and Luna, 2006] Serrano, L. F. F. and Luna, R. A. (2006). Combatiendo códigos maliciosos. In *DSC*.

[Skoudis, 2004] Skoudis, E. (Mayo 2004). The evolution of malware. *Intelguardians*.

[Skoudis and Zeltser, 2003] Skoudis, E. and Zeltser, L. (2003). *Malware: Fighting Malicious Code*. Prentice Hall.

[Smith, 2008] Smith, B. (Febrero de 2008). A storm (worm) is brewing. *IEEE Computer Society*.

[Stoneburneri et al., 2002] Stoneburneri, G., Goguen, A., and Feringa, A. (2002). *Risk Management Guide for Information Technology Systems*. National Institute of Standards and Technology.

[Symantec, 2008] Symantec (2008). Symantec internet security threat report trends for july–december 07. *Symantec*.

[Szor, 2005] Szor, P. (2005). *Art of Computer Virus Research and Defense, The*. Addison Wesley Professional.

[US-CERT, 2006] US-CERT (2006). Defending cell phones and pdas against attack. http://www.us-cert.gov/cas/tips/ST06-007.html.

[Vixie and Dragon, 2008] Vixie, P. and Dragon, D. (2008). Malware repository requirements. *Defcon 14*.

[Walker, 2005] Walker, A. (2005). *Absolute Beginner's Guide To: Security, Spam, Spyware & Viruses*. Que.

[Wen et al., 2008] Wen, Y., Zhao, J., and Wang, H. (2008). Implicit detection of hidden processes with a local-booted virtual machine. *International Conference on Information Security and Assurance*.

[Willems et al., 2007] Willems, C., Holz, T., and Freiling, F. (2007). Toward automated dynamic malware analysis using cwsandbox. *IEEE Security & Privacy*.

[Wright, 2007] Wright, C. S. (2007). A taxonomy of information systems audits, assessments and reviews. *SANS Institute*.

[Wu and Shi, 2006] Wu, Y. and Shi, W. (2006). Portal monitoring based anti-malware framework: Design and implementation. *IEEE*.

[XiTi Monitor, 2008] XiTi Monitor (2 de junio de 2008). Operating systems survey 2008. http://www.xitimonitor.com/en-us/internet-users-equipment/operating-systems-april-2008/index-1-2-7-129.html.

[Yin et al., 2008] Yin, H., Song, D., Egele, M., Kruegel, C., and Kirda, E. (2008). Panorama: Capturing system-wide information flow for malware detection and analysis. *ACM*.