**Métodos de detección de malware oculto**

Existen varias maneras de poder detectar cuándo un software malicioso se ha introducido en un sistema o equipo determinado. A continuación mencionamos las más importantes:

**Detección basada en firmas**

Es uno de los mecanismos más usados debido a su sencillez, velocidad y destacable precisión a la hora de identificar malware específico.

Este modo de detección compara la “firma” de los archivos a analizar con una base de datos y si hay coincidencia, el archivo es identificado como malware. Hay 3 tipos de firmas: archivos completos, archivos parciales, cadena de firmas.

1. **Archivos completos**

Se computa un “fingerprint” del archivo completo usando un hash criptográfico por ejemplo, SHA-1 y compararlos con los hashes de las firmas almacenados en una base de datos, debido a las características de estos hashes, el mínimo cambio en el archivo de entrada cambiaría por completo el “fingerprint” resultante lo cual impedirá compararlo exitosamente con la base de datos ya que es prácticamente imposible que 2 archivos distintos generen el mismo fingerprint. Por lo tanto los malwares pueden ser detectados de manera rápida y eficaz.

1. **Archivos parciales**

A diferencia del caso anterior, el “fingerprint” computará sólo de ciertas secciones del archivo. En los archivos de malware hay secciones que no tienen tanta relevancia y pueden ser modificadas fácilmente por lo que el análisis se centra en las secciones críticas del funcionamiento del malware y así ser capaces de detectar variantes de malware de manera eficaz.

1. **Cadena de firmas**

Siguiendo el mecanismo de las anteriores firmas se analizarán las posibles cadenas de caracteres y se compararán con la base de datos, este método no es muy eficaz con malware oculto ya que no suelen interactuar con el usuario, pero con malware como el “Stone virus” dado que contiene una cadena diciendo “Your PC is now stoned” un archivo con esa cadena puede ser identificado como posible malware, se tendrán más características en cuenta para determinar qué es malware.

**Ventajas y Desventajas**

La detección basada en firmas es una técnica muy fácil de implementar y gracias a la utilización de hashes criptográficos es muy eficaz a la hora de encontrar amenazas conocidas por lo que su tasa de falsos positivos es bajísima.

Las desventajas más significativas de la detección basado en firmas son causadas debido a la base de datos. La base de datos juega un papel crucial en el sistema, por lo que tiene que estar siendo actualizada continuamente para que se puede detectar nuevos malware generados lo cual lo convierte en un sistema vulnerable a zero-day malware. Otra gran desventaja es que cuanto más grande sea la base de datos, más carga habrá sobre la CPU del sistema por lo que es difícil encontrar un equilibrio entre rapidez del sistema y eficacia de la base de datos. Esta técnica de detección tampoco es capaz de combatir malware que muta su código en cada infección como polymorphic y metamorphic malware.

**Detección basada en comportamiento**

La detección basada en comportamiento evita el uso de las firmas criptográficas. Se analiza un archivo para determinar si es un posible malware centrándonos mucho más en qué hace el archivo al ejecutarlo en vez del contenido del mismo. Este tipo de mecanismo permite detectar malware que muta ya que siempre usarán los recursos y servicios del sistema de manera similar. Un detector basado en comportamiento está dividido en tres bloques:

El primer bloque recoge información dinámica/estática del ejecutable, el segundo bloque convierte la información obtenida en representaciones intermedias la cual el tercer bloque utilizará para determinar si el archivo analizado es considerado como malware.

**Ventajas y desventajas**

La principal ventaja de este método es su capacidad de detectar el malware que el método por firmas no es capaz como malware desconocido y variaciones de polymorphic malware. Las desventajas más significativas son la tasa alta de falsos positivos y la cantidad de tiempo durante el escaneo de archivos.

**Detección basado en análisis Heurístico**

Detección de malware utilizando usando análisis heurísticos fue creado para superar las desventajas encontradas en la detección por firmas o comportamiento. Los métodos usados para ello son el procesamiento de datos y las técnicas de aprendizaje automáticas para aprender el comportamiento de un ejecutable. Las características de este tipo de detección se pueden dividir generalmente en cuatrobloques.

1-     Llamada a sistema API

Casi todos los programas usan llamadas API para enviar sus solicitudes al sistema operativo. Las secuencias de llamadas API se comportan como fragmentos de código malware. Hofmeyr fue de los primeros que consideró estas secuencias como características de malware[3] por lo cual se desarrolló una detección basada en llamadas a sistema. La distancia de haming que es el número mínimo de sustituciones que se tiene que cambiar en una cadena para que sea idéntica a otra, fue utilizado para comparar las secuencias de llamadas API y poder clasificarlo como malware.

2-     Código Operacional

Código operacional es la subdivisión de lenguaje de instrucciones de máquina que identifica la operación a realizar. Un programa está definido por una serie de instrucciones de ensamblador. Bilar mostró la habilidad de utilizar código operacional como características en la detección de malware, analizó y demostró la alta fiabilidad para detectar amenazas de un ejecutable [4]. Hay varios tipos de detección de malware basados en código operacional por ejemplo un enfoque centrado en detectar variantes de malware ofuscado. Normalmente el procedimiento es descomponer el ejecutable en instrucciones de ensamblador y calcular la relevancia de estas instrucciones basándose en la técnica de frecuencia-peso de término se realiza un vector que representa las características del ejecutable para poder ser estudiado como posible amenaza. También es posible detectar malware desconocido  y metamorphic basado en gráficas de medición de similitud para poder llevar a cabo esta técnica de detección basada en código operacional se extrae código operacional de archivos benignos y malwares y se cuenta el número de código operacional

3-     Diagrama de flujos

       Los diagramas de flujo son representaciones  gráficas que describe el funcionamiento de un algoritmo o proceso. Está compuesto por nodos que representan una instrucción del programa y de líneas de flujo que muestra el flujo de control entre instrucciones. Se puede detectar malware a través del uso del diagramas de flujos[3] mediante un conjunto de operaciones de normalización realizadas después de “desmontar” el ejecutable para reducir el efecto de tecnicas de mutacion y poder desvelar conexiones de flujo entre código benigno y malware se crea un diagrama de flujo el cual se comparará con diagramas de flujos de malware normalizado.

 Zhao [4] propuso un método de detección de malware utilizando diagramas de flujo. Primero obtiene el diagrama siguiendo los pasos previo, a continuación utiliza las características obtenidas como información de los nodos, líneas de flujos y sub-flujos para utilizar un algoritmo de procesamiento de datos como  [5][6][7].

Los diagramas de flujo pueden ser incluso utilizados como firmas de detección. Como se ha mencionado los diagramas de flujo está compuesto por nodos y líneas de flujo y como sabemos cada ensamblador consiste en cuatro tipo de instrucciones: saltos condicionales, saltos incondicionales, llamadas a función y retorno de llamadas. Bosfante [8] abstrajo cualquier secuencia de instrucciones seguidas en un nuevo nodo llamado inst y al final del programa se creó otro nodos denominado end. Utilizando estos seis nodos se construye un diagrama de flujo del ejecutable. Para reducir el tamaño del diagrama de flujo los nodos de salto incondicional y los nodos inst son eliminados y se enlazan los nodos que estuvieran unidos por los eliminados. El resultado se utilizara como una firma.

4-     Caracteristica h­­­ibrida

Cada bloque previamente estudiado tiene sus propias ventajas y desventajas a la hora de detectar malware por ejemplo con el uso de diagramas de flujo se puede detectar fácilmente malware simple pero para detectar malware polimórfico o metamórfico no es suficiente es por eso que a la hora de crear un sistema de detección se usa una combinación de los bloques expuestos para obtener un mejor resultado. Eskandari[9] creó un sistema el cual utilizaba diagramas de flujo y llamadas a Sistema API.C

**Prevención de malware oculto**

Existen muchos y muy efectivos métodos de detección de malware que nos ayudan a permanecer seguros y la mayoría son capaces de detectar el malware antes de que este cause problema al sistema atacado pero como todos sabemos en mejor prevenir que curar por lo que poniendo en práctica las siguientes técnicas de prevención de malware creamos otra capa de seguridad.

**Proteger vulnerabilidades**

Hoy en día uno de las herramientas más usados a la hora de entregar malware son los “exploit kit” y la manera más eficaz para protegerse contra ellos es teniendo las aplicaciones y el sistem operativo actualizado. Actualizaciones de buscadores, plugins y sistemas operativos son a menudo lanzadas para parchear vulnerabilidades de seguridad descubiertas.Otra técnica bastante efectiva es habilitar “click-to-play” plugins ya que usando anuncios maliciosos  exploit kits pueden llegar a tu sistema ya que sin esta opción habilitada ni siquiera necesitas hacer clic en un anuncio. Por último evitar la utilización de sistemas operativos que han dejado de tener mantenimiento periódico como por ejemplo  windows 7 que este año dio por finalizado su mantenimiento periódico.

**Cuidado con la ingeniería social**

La estafa utilizando la ingeniería social es otro metodo de infeccion muy utilizado en la actualidad. Ya se un correo electrónico,campana social sospechosa hasta posible llamadas del soporte técnico de tu sistema operativo.Estando alerta de las siguientes tácticas de ingeniería social se puede prevenir ser engañado.

1- Leer emails meticulosamente. La suplantación de identidad a través de un correo electrónico es exitosa sólo cuando los lectores no prestan atención o no saben en qué fijarse. Comprobando que la dirección de correo esté verificada, los enlaces de redireccionamiento llevan a páginas oficiales, no solicitan información privada por medios no seguros etc ayudan a verificar si el correo electrónico es o no legítimo.

2-Instalar programas falsos. Ignorando los pop-ups que ofrezcan ayuda contra malware mediante una aplicación no oficial ya que una empresas reales de seguridad nunca se promociona mediante pop-ups y menos alertando de que han detectado que estas infectado cuando ni siquiera tienes su aplicación instalada.

**Navegación Segura**

“Siguiendo consejos básicos y manteniendo buenos hábitos a la hora de navegar evitaremos el 95% de los ataques apuntando” , Adam Kujaya. Algunos de los consejos y hábitos son:

1. Usar contraseñas seguras o administradores de contraseñas
2. Asegurarse de tener acceso a internet a través de redes seguras
3. Cerrar sesion de páginas cuando termines de utilizarlas**.**

**SISTEMAS DE DETECCIÓN DE INTRUSIONES**

**Sistema de Detección de Intrusiones (IDS)**

Un IDS es un hardware o software que automatiza la detección monitorizando sistemas y redes de computadores en busca de evidencias de violación de una determinada política de seguridad y genera informes interpretables por bases de datos. En general, un IDS consta de

1. Un registro de eventos que es su fuente de información.

Los datos de las fuentes de información se pueden obtener de fuentes externas, asociadas a la red o internas de los sistemas individuales. Cada fuente puede detectar ciertos eventos específicos.

2.Un motor de análisis para encontrar signos de intrusiones.

Se encarga de organizar y caracterizar los datos de actividad del sistema o red en busca de información que sea de interés. Los modos más comunes son 2: Detección de un mal uso (misuse detection) y detección de anomalías (anomaly detection).

3. Un componente de respuesta.

Las respuestas a intrusiones se dividen en 2 tipos: activas y pasivas. Los sistemas comerciales actuales contemplan una variedad de respuestas activas, pasivas y combinaciones de ambas.

* Respuestas activas.

Tras producirse un evento, realizan una acción preconfigurada. Estas acciones pueden suponer un problema ya que son respuestas automatizadas. Entre ellas están::

*Recopila información adicional*: incrementa el nivel de sensibilidad de las fuentes y redirige el ataque a máquinas señuelo.

*Trata de detener el ataque:* corta la conexión o niega el acceso por IP. Reconfigura mecanismos de filtrado de paquetes y deshabilita interfaces de red.

*Toma acciones contra el atacante:* contacta con el administrador de la red atacante o contraataca.

* Respuestas pasivas.

Se limitan a registrar el evento sucedido. Entre ellas se encuentran:

*Alarmas y notificaciones:* Pop-ups, notificaciones remotas a teléfonos móviles o busca y notificación vía e-mail desaconsejada.

*SNMP traps:* algunos IDS generan alarmas y las envían a centrales de gestión de red. Permiten adaptar toda la infraestructura de red ante un ataque y libera al equipo afectado por el ataque de la carga de procesar las acciones de respuesta al mismo.

Además de la diferenciación de los IDS por las características de sus componentes, se pueden dividir según su arquitectura en dos tipos: integrados en el objeto de la detección y separado de él.

Se pueden monitorizar varios elementos: *hosts* con logs a nivel del Sistema Operativo, *redes* mediante la captura de paquetes y *aplicaciones* a partir de sus logs entre otros.

También se pueden seguir diferentes estrategias de monitorización. Dependiendo del tipo de control puede ser centralizado o distribuido. Se puede detectar anomalías y la mala utilización, a intervalos de bloques o en tiempo real. Debemos de ajustar estos detalles adecuadamente para cumplir nuestros objetivos.

Un punto a tener en cuenta cuando se utiliza un IDS es su tolerancia a fallos e invisibilidad.. Un atacante no debería de poder identificar al IDS, puesto que lo atacarían directamente. Por lo que las respuestas no deben de alertarlos. El uso de criptografía y túneles cifrados hará más difícil que un posible atacante detecte las alertas generadas por el IDS.

**Host-based IDS (HIDS)**

Este tipo de IDS obtiene la información de los mismos dispositivos. Los logs del sistema son más fáciles de interpretar, pero están menos detallados.

Los registros de auditorías proporcionan información procedente del *Kernel* del sistema operativo. En general, esa información es más detallada, más fiable y más difícil de alterar.

Los monitores de objetivos monitorizan el estado de determinados objetos del sistema. Por ejemplo, los verificadores de integridad.

Ventajas: permiten determinar qué procesos y usuarios están involucrados en un determinado ataque (y distinguir entre procesos y usuarios), observar el desenlace de diversos ataques (escalada de privilegios, buffer Overflows), detectar anomalías en la ejecución de procesos (troyanos y otro software malicioso, intentos de modificación del Kernel).

Inconvenientes: tienen poca escalabilidad ya que solo puede haber un IDS por host. Estos IDS pueden ser también objetivo de ataque y pueden ser deshabilitados en ataques DoS. No son adecuados cuando se produce sniffing de red o para ataques de sondeo de redes enteras. Además, al generar gran cantidad de información, afectan al rendimiento del host.

**Application-based IDS**

Este tipo de IDS trabaja a nivel de aplicación y han de emplearse junto con un HIDS o NIDS para proteger apropiadamente.

Ventajas: detectan ataques específicos a una aplicación y el análisis es más exhaustivo.

Inconvenientes: los logs de las aplicaciones están menos protegidos. Trabajan a nivel de abstracción de usuario y no son capaces de detectar troyanos

**Network-based IDS (NIDS)**

Capturan y analizan los paquetes de la red a la que están conectados. Consiste en un conjunto de hosts (sensores), tarjetas en modo promiscuo en segmentos de red y una máquina en el perímetro de la red.

Ventajas: escalabilidad (podemos controlar una red de gran tamaño), poco impacto en el funcionamiento de la red y seguridad (máquinas dedicadas e invisibles a los atacantes).

Inconvenientes: eficiencia (es difícil analizar grandes cantidades de tráfico), dificultades en redes conmutadas, no pueden analizar información cifrada a nivel de red, no detectan el resultado del ataque y pueden tener problemas con la fragmentación.

**HIDS vs NIDS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Function** | **HIDS** | **NIDS** | **Comments** |
| **Protección en LAN** | **\*\*\*\*** | **\*\*\*\*** | **Los 2 sistemas protegen en LAN** |
| **Proteccion fuera de LAN** | **\*\*\*\*** | **-** | **Solo HIDS protege fuera de LAN** |
| **Ease of Implementation** | **\*\*\*\*** | **\*\*\*\*** | **Both NIDS and HIDS are equal form a central control perspective** |
| **Ancho de banda en LAN** | **0** | **2** | **NIDS usa ancho de banda LAN.** |
| **Gastos de red** | **1** | **2** | **NIDS tiene el doble de requisitos de ancho de banda de cualquier LAN** |
| **Ancho de banda(internet)** | **\*\*** | **\*\*** | **Los dos IDS lo necesitan** |
| **Frecuencia de actualización para clientes** | **\*\*\*\*** | **-** | **HIDS actualiza todos los clientes mediante un archivo de patrón centra** |
| **Compatibilidad multiplataforma** | **\*\*** | **\*\*\*\*** | **NIDS más adaptables** |
| **Escaneo de registros de maquinas locales** | **\*\*\*\*** | **-** | **Solo HIDS puede hacer estos escaneos** |
| **REchazo de paquetes** | **-** | **\*\*\*\*** | **Solo NIDS tiene esta funcionalidad** |
| **Disable risk factor** | **\*** | **\*\*\*\*** | **NIDS failure rate is much higher than HIDS failure rate.  NIDS has one point of failure.** |

**Bibliografia**

**Signatures vs behaving**

[**https://www.cyberdefensemagazine.com/advanced-malware-detection/**](https://www.cyberdefensemagazine.com/advanced-malware-detection/)

**Heuristics pasivos/activos**

[**https://www.welivesecurity.com/la-es/2010/02/16/tipos-heuristica/**](https://www.welivesecurity.com/la-es/2010/02/16/tipos-heuristica/)

[**https://support.eset.com/en/kb127-what-are-heuristics**](https://support.eset.com/en/kb127-what-are-heuristics)

[**https://www.welivesecurity.com/la-es/2013/03/18/heuristica-antivirus-deteccion-proactiva-amenazas/**](https://www.welivesecurity.com/la-es/2013/03/18/heuristica-antivirus-deteccion-proactiva-amenazas/)

**Deteccion en general**

**https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?referer=https://scholar.google.com/&httpsredir=1&article=1329&context=etd\_projects**

**Deteccion por comportamiento**

**1-G. Jacob, H. Debar, and E. Filiol, “Behavioral detection of malware: from a survey towards an established taxonomy,” Journal in Computer Virology, pp. 251–266, 2008  (COMPONENTES DE DETECION POR COMPORTAMIENTO)**

**[2]D. Bilar, “OpCodes as predictor for malware,” International Journal of Electronic Security and Digital Forensics, vol. 1, no. 2, p. 156, 2007.**

**[3] S. Hofmeyr, S. Forrest, and A. Somayaji, “Intrusion detection using sequences of system calls.” Journal of Computer Security, pp. 151–180, 1998.**

**[4] D. Bruschi, L. Martignoni and M. Monga “Detecting self-mutating malware using control-flow graph matching,” In: Büschkes, R. and Laskov, P. (eds) Detection of Intrusions and Malware & Vulnerability Assessment, volume 4064 of LNCS, pp 129–143. Springer, Berlin. 2006.**

**[5]T. M. Mitchell, “Machine learning and data mining,” Commun. ACM, vol. 42, no. 11, 1999.**

**[6] L. Breiman. “Bagging Predictors.” Machine Learning, 24(2):123–140, 1996.**

**[7] L. Breiman. “Random Forests.” Machine Learning, 45(1):5–32, 2001**

**[8]M. Eskandari and S. Hashemi “Metamorphic malware detection using control flow graph mining”. International Journal of Compute Science Network Security,pp 1-6 ,2011**

**Prevencion de malware**

[**https://blog.malwarebytes.com/101/2016/08/10-easy-ways-to-prevent-malware-infection/**](https://blog.malwarebytes.com/101/2016/08/10-easy-ways-to-prevent-malware-infection/)

**“following some basic tips and maintaining good habits while online, you will evade infection from over 95 percent of the attacks targeting you,” says Adam Kujawa, Head of Intelligence for Malwarebytes.**

**Tabla HIDS vs NIDS**

[**http://techgenix.com/Hids\_vs\_Nids\_Part1/**](http://techgenix.com/Hids_vs_Nids_Part1/)

**Información sobre IDS y tipos:**

[**http://www.adminso.es/index.php/4.2.5.\_IDS\_basados\_en\_host\_(HIDS)**](http://www.adminso.es/index.php/4.2.5._IDS_basados_en_host_(HIDS))

[**https://www.csirtcv.gva.es/es/paginas/sistemas-detectores-de-intrusi%C3%B3n-nids-y-hids.html**](https://www.csirtcv.gva.es/es/paginas/sistemas-detectores-de-intrusi%C3%B3n-nids-y-hids.html)