SISTEMA DE DETECCIÓN DE INTRUSOS

SWAP

AUTORES:CARLOS ARIZA GARCIA, FERNANDO TALAVERA MENDOZA, LEANDRO VEGA PALMA

INTRODUCCIÓN

Cuando se desea montar un servidor, este se convierte en un objetivo de ataques malintencionados. Por tanto existen numerosas medidas de seguridad para proteger los recursos informáticos de una empresa. En consecuencia, conseguir que un sistema sea invulnerable es sumamente costoso.

Hoy en día, dentro de las soluciones tecnológicas que se encuentran disponibles para reforzar la seguridad de una red encontramos los firewall. Un firewall es sistema encargado del cumplimiento de las políticas de control de acceso a la red, lo cual se hace a través de reglas (ufw, iptables). Dicho de otra forma, el firewall actúa de muro perimetral para proteger una red de ataques del exterior a nuestro sistema, pero posee una serie de desventajas como:

- El atacante puede pasar el firewall, controlando la red.
- El firewall protege de los accesos no autorizados hacia la red interna pero no protege las propias máquinas. Supongamos que el balanceador tiene establecido el firewall mientras que las máquinas internas se limitan a realizar su propio trabajo (Web, correo, etc).
- El firewall no protege de ataques internos.

Como podemos observar, el firewall posee una serie de desventajas que propia acarrear graves consecuencias y causar daños dentro del sistema informático de la empresa, por tanto debemos de optar por otra alternativa. En nuestro caso, vamos a implementar un sistema de detección de intrusiones, que en simples términos se podría comparar como una alarma antirrobo.

UN POCO DE HISTORIA

El primer trabajo que se realizó sobre los IDSs fue en 1980, realizado por James P. Anderson, año en el que empezaron a incrementarse los problemas de seguridad de los computadores, donde estudió cómo mejorar la seguridad y vigilancia en los clientes. La idea original detrás de los IDSs automáticos está acreditada a James por su papel en “How to use

accounting audit files to detect unauthorized access". Este estudio allanó el camino de la detección en los mainframes.

En la década de los 80 se diseñó el primer sistema IDS, que funcionaba en tiempo real, desarrollado por Dorothy Denning y Peter Neumann. Se crea un auge desde 1995, cuando se produce la crisis de los firewalls. Actualmente, es uno de los campos con más investigación y avances.

INTRUSIÓN

Antes de continuar vamos definir qué se entiende por intrusión. Una intrusión es cualquier conjunto de acciones que puede comprometer la integridad, confidencialidad o disponibilidad de una información o un recurso informático. Los intrusos pueden utilizar debilidades y brechas en la arquitectura de los sistemas y el conocimiento interno del sistema operativo para superar el proceso normal de autenticación.

La detección de intrusos se puede detectar a partir de la caracterización anómala del comportamiento y del uso que hacen de los recursos del sistema. Este tipo de detección pretende cuantificar el comportamiento normal de un usuario. Para una correcta distinción hay que tener en cuenta las tres distintas posibilidades que existen en un ataque, atendiendo a quién es el que lo lleva a cabo:

- Penetración externa. Que se define como la intrusión que se lleva a cabo a partir un usuario o un sistema de computadores no autorizado desde otra red.

- Penetraciones internas. Son aquellas que llevan a cabo por usuarios internos que no están autorizados al acceso.

- Abuso de recursos. Se define como el abuso que un usuario lleva a cabo sobre unos datos o recursos de un sistema al que está autorizado su acceso.

La idea central de este tipo de detección es el hecho de que la actividad intrusiva es un subconjunto de las actividades anómalas. Esto puede parecer razonable por el hecho de que si alguien consigue entrar de forma ilegal en el sistema, no actuará como un usuario normal. Sin embargo en la mayoría de las ocasiones una actividad intrusiva resulta del agregado de otras actividades individuales no que por sí solas no constituyen un comportamiento intrusivo de ningún tipo, como por ejemplo, el acceso múltiple de varias cuentas de forma

no consentida. Idealmente el conjunto de actividades anómalas es el mismo del conjunto de actividades intrusivas, de todas formas esto no siempre es así:

- Intrusivas pero no anómalas: denominados Falsos Negativos (el sistema no indica intrusión). En este caso la actividad es intrusiva pero como no es anómala no es detectada.

- No intrusivas pero anómalas: denominados Falsos Positivos (el sistema erróneamente indica la existencia de intrusión). En este caso la actividad es no intrusiva, pero como es anómala el sistema "decide" que es intrusiva. Deben intentar minimizarse, ya que en caso contrario se ignorarán los avisos del sistema, incluso cuando sean acertados.

- No intrusiva ni anómala: son Negativos Verdaderos, la actividad es no intrusiva y se indica como tal.

- Intrusiva y anómala: se denominan Positivos Verdaderos, la actividad es intrusiva y es detectada.

Los primeros no son deseables, porque dan una falsa sensación de seguridad del sistema y el intruso en este caso puede operar libremente en el sistema. Los falsos positivos se deben de minimizar, en caso contrario lo que puede pasar es que se ignoren los avisos del sistema de seguridad, incluso cuando sean acertados. Los detectores de intrusiones anómalas requieren mucho gasto computacional, porque se siguen normalmente varias métricas para determinar cuánto se aleja el usuario de lo que se considera comportamiento normal.

Leer más:[REFINFO9].

DEFINICIÓN

Volviendo a lo anteriormente mencionado, un IDS (Intrusion Detection System) es un mecanismo que monitoriza el tráfico en la red con el fin de detectar actividades sospechosas o accesos de usuarios no autorizados a un computador o a una red. Es un componente más en la seguridad de una empresa y que normalmente se integra con un firewall, ya que se combina la inteligencia del IDS y el poder de bloqueo del este.

Mientras que esta sería su principal funcionalidad, otros IDSs son capaces de tomar decisiones, (los llamados reactivos) ante estos accesos no autorizados o tráfico malicioso, pudiendo bloquearlos. Para su correcto funcionamiento, una vez instalado es necesario configurarlo adecuadamente atendiendo a las necesidades de la empresa, ya que puede provocar falsas alarmas.

CARACTERÍSTICAS

Todo IDS, independientemente del mecanismo en el que esté basado, debe cumplir estos puntos para ser considerado un IDS fiable y de calidad:

1. Independencia: debe funcionar continuamente sin supervisión humana. El sistema debe ser lo suficientemente fiable para poder ser ejecutado en background dentro del equipo que está siendo observado.

2. Tolerancia a fallos: debe ser tolerante a fallos, es decir mantenerse ante caídas del sistema

3. Resistencia a perturbaciones: en relación con el punto anterior, debe ser resistente a perturbaciones. El sistema puede monitorizarse a sí mismo para asegurarse de que no ha sido perturbado.

4. Sobrecarga mínima: debe imponer mínima sobrecarga sobre el sistema.

5. Detección de intrusos: debe observar desviaciones sobre el comportamiento estándar.

6. Adaptabilidad: debe ser fácilmente adaptable al sistema ya instalado. El mecanismo de defensa debe adaptarse de manera sencilla a esos patrones. También debe hacer frente a los cambios de comportamiento del sistema según se añaden nuevas aplicaciones al mismo.

TIPOS DE IDSS

1. Según su naturaleza

   - Pasivos: sólo detectan la intrusión, no toman ninguna medida

   - Reactivos: detectan las intrusiones y responden ante ellas tomando las medidas posibles y adecuadas, cómo bloquear direcciones IP o cortar el acceso a recursos restringidos.

2. Según su software

NIDS (NETWORK-BASED IDS):

La mayor parte de los sistemas de detección de intrusos son de este tipo. Estos detectan ataques a todo el segmento de la red en el que se encuentren, o en un backbone de la red. Escuchando en dichas localizaciones, un NIDS puede monitorizar el tráfico que afecta a múltiples hosts que están conectados a ese segmento de red, protegiendo así a estos hosts. Su interfaz debe funcionar en modo promiscuo capturando así todo el tráfico de la red. Este es una especie de modo "invisible" en el que no tienen dirección IP para que un atacante no pueda determinar su presencia y localización.

Los NIDS no sólo vigilan el tráfico entrante, sino también el saliente o el tráfico local, ya que algunos ataques podrían ser iniciados desde el propio sistema protegido. A pesar de la vigilancia, su influencia en el tráfico es casi nula. A menudo están formados por un conjunto de sensores desplegados en varios puntos estratégicos en la red, donde pueden monitorizar todo el tráfico de esta.

Ventajas

- Un IDS bien localizado puede monitorizar una red grande, siempre y cuando tenga la capacidad suficiente para analizar todo el tráfico.

- Los NIDSs tienen un impacto pequeño en la red, siendo normalmente dispositivos pasivos que no interfieren en las operaciones habituales de ésta.

- Se pueden configurar para que sean muy seguros ante ataques haciéndolos invisibles al resto de la red.

Desventajas

- Pueden tener dificultades procesando todos los paquetes en una red grande o con mucho tráfico y pueden fallar en reconocer ataques lanzados durante periodos de tráfico alto. Algunos vendedores están intentando resolver este problema implementando IDSs completamente en hardware, lo cual los hace mucho más rápidos.

- Los IDSs basados en red no analizan la información cifrada. Este problema se incrementa cuando la organización utiliza cifrado en el propio nivel de red (IPSec) entre hosts, pero se puede resolver con una política de seguridad más relajada (por ejemplo, IPSec en modo túnel).

- Los NIDSs no saben si el ataque tuvo o no éxito, lo único que pueden saber es que el ataque fue lanzado. Esto significa que después de que un NIDS detecte un ataque, los administradores deben manualmente investigar cada host atacado para determinar si el intento de penetración tuvo éxito o no.

- Algunos NIDS tienen problemas al tratar con ataques basados en red que viajan en paquetes fragmentados. Estos paquetes hacen que el IDS no detecte dicho ataque o que sea inestable e incluso pueda llegar a caer.

HIDS (HOST-BASED IDS):

Los HIDSs se ejecutan en todos los ordenadores o dispositivos en la red con acceso directo tanto a internet cómo a la red interna de la empresa. La principal ventaja de los HIDS frente a los NIDS es que pueden detectar anomalías en los paquetes de la red que se originan desde dentro de la organización o tráfico malicioso que los NIDSs no son capaz de detectar.

Los HIDSs también detectan tráfico malicioso generado desde el mismo host, en el caso en el que se infecte el host con algún malware e intente propagarse a los demás dispositivos. Esto permite que el IDS analice las actividades que se producen con una gran precisión, determinando exactamente qué procesos y usuarios están involucrados en un ataque particular dentro del sistema operativo.

Los HIDSs fueron el primer tipo de IDSs desarrollados e implementados. A diferencia de los NIDSs, los HIDSs pueden ver el resultado de un intento de ataque, al igual que pueden acceder directamente y monitorizar los ficheros de datos y procesos del sistema atacado.

Ventajas

- Los IDSs basados en host, al tener la capacidad de monitorizar eventos locales a un host, pueden detectar ataques que no pueden ser vistos por un IDS basado en red.

- Pueden a menudo operar en un entorno en el cual el tráfico de red viaja cifrado, ya que la fuente de información es analizada antes de que los datos sean cifrados en el host origen y/o después de que los datos sea descifrados en el host destino.

Desventajas

- Los IDSs basados en hosts son más costosos de administrar, ya que deben ser gestionados y configurados en cada host monitorizado. Mientras que con los NIDS teníamos un IDS por múltiples sistemas monitorizados, con los HIDS tenemos un IDS por sistema monitorizado.

- Si la estación de análisis se encuentra dentro del host monitorizado, el IDS puede ser deshabilitado si un ataque logra tener éxito sobre la máquina.

- No son adecuados para detectar ataques a toda una red (por ejemplo, escaneos de puertos) puesto que el IDS solo ve aquellos paquetes de red enviados a él.

- Pueden ser deshabilitados por ciertos ataques de DoS.

- Usan recursos del host que están monitorizando, influyendo en el rendimiento del sistema monitorizado.

TIPOS DE ANÁLISIS

Hay dos acercamientos al análisis de eventos para la detección de ataques: detección de abusos o firmas y detección de anomalías. La detección de abusos es la técnica usada por la mayoría de sistemas comerciales. La detección de anomalías, en la que el análisis busca patrones anormales de actividad, ha sido y continúa siendo objeto de investigación. La detección de anomalías es usada de forma limitada por un pequeño número de IDSs.

En el IDS basado en firmas, existen patrones o reglas que contemplan tráfico malicioso conocido y que serán cotejadas por la herramienta. Cuando se encuentre una coincidencia con un patrón, recibiremos una alerta. Alertas de este tipo nos pueden avisar de problemas como malware, actividades de escaneo de red o ataques contra servidores, entre otras cosas.

Si pasamos al modelo basado en anomalías, el payload o “carga” del tráfico es bastante menos relevante respecto a la actividad que genera este. Un IDS basado en análisis de anomalías se basa en una “línea base” de funcionamiento, a partir de la cual se buscará actividad inusual que se desvíe de los promedios , así como actividad no contemplada anteriormente.

¿DÓNDE COLOCAR UN IDS?

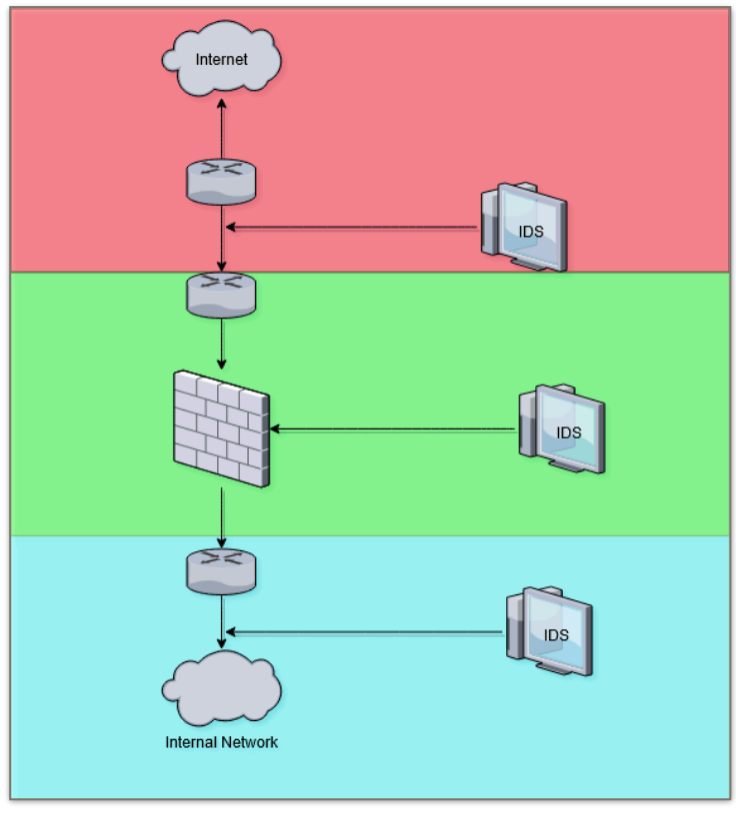
EN UNA ORGANIZACIÓN

Podríamos distinguir tres zonas diferenciadas en las que podemos instalar un IDS. En cada zona varía la gravedad de las alertas y la sensibilidad del sistema. Estas tres zonas son:

Zona roja: Localizada en el extremo exterior de nuestra red. Analiza todos los datos que entran y salen de ella, por lo que debe ser configurado para ser poco sensible y habrá más posibilidades de que surjan falsas alarmas.

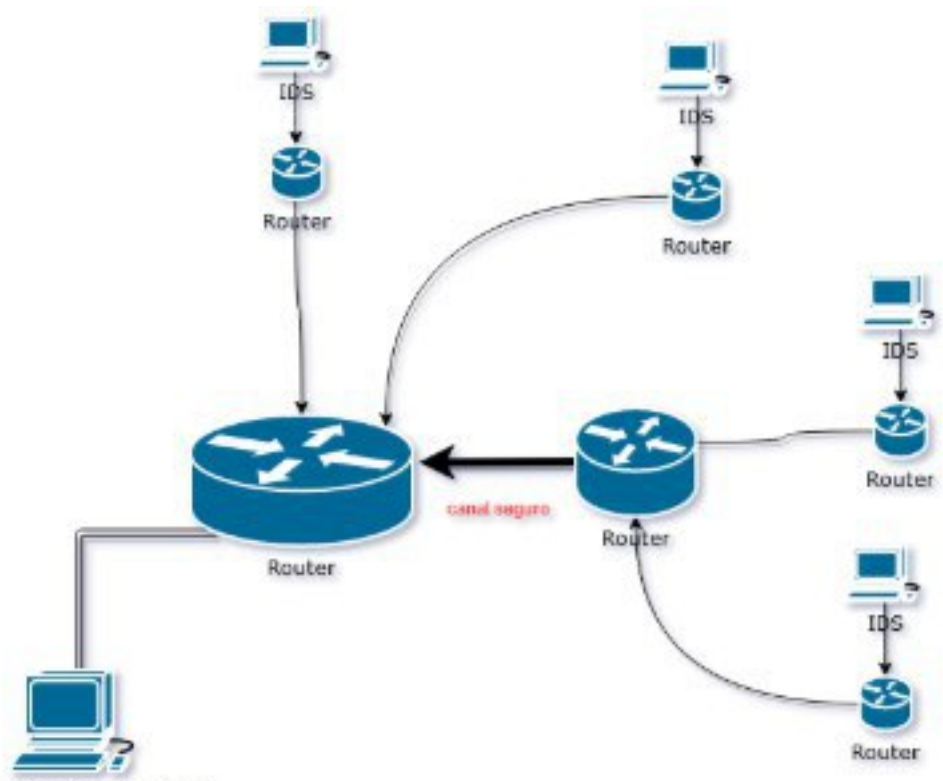
Zona verde: Se situaría en la DMZ si la hubiera o detrás del firewall. En este punto, el firewall ya habrá bloqueado una parte importante de las amenazas, por lo que habría que configurar el IDS para que fuera un poco más sensible que otro situado en la zona roja. En esta zona debería producirse un número menor de falsas alarmas, puesto que las acciones que se pueden hacer en esta zona son más limitadas.

Zona azul: Estaría situada directamente en nuestra red interna. Esta es una zona de confianza, por lo que la sensibilidad del IDS deberá ser máxima. La cantidad de alertas en esta zona debería ser mínima, y cada aviso debe ser estudiado con total rapidez.



EN UN ISP

El tráfico que gestiona un ISP es demasiado grande como para poder ser gestionado por sólo un IDS. Por lo tanto, es necesario tener los sensores y el centro de procesamiento de las alertas separados y conectados por un canal seguro cifrado. De esta forma, se implantarían sensores en los nodos a los que dé servicio el ISP y estos, de forma periódica, enviarán las alertas a la estación de análisis. Que se haga periódicamente es importante para evitar que algún atacante pudiera reconocer si sus intrusiones han sido detectadas o no.



IDS FRENTE A IPS

Un sistema de prevención de intrusos (o por sus siglas en inglés IPS) es un software que ejerce el control de acceso en una red informática para proteger a los sistemas computacionales de ataques y abusos. La tecnología de prevención de intrusos es considerada por algunos como una extensión de los sistemas de detección de intrusos (IDS), pero en realidad es otro tipo de control de acceso, más cercano a las tecnologías cortafuegos.

Un IPS es un dispositivo de seguridad de red que monitoriza el tráfico de red y/o las actividades de un sistema, en busca de actividad maliciosa. Entre sus principales funciones, se encuentran no sólo la de identificar la actividad maliciosa, sino la de intentar detener esta

actividad. Siendo esta última una característica que distingue a este tipo de dispositivos de los IDS.

Los IPS se clasifican en cuatro diferentes tipos:

1. Basados en Red Lan (NIPS): monitorizan la red lan en busca de tráfico de red sospechoso al analizar la actividad por protocolo de comunicación lan.

2. Basados en Red Wireless (WIPS): monitorean la red inalámbrica en busca de tráfico sospechoso al analizar la actividad por protocolo de comunicación inalámbrico.

3. Análisis de comportamiento de red (NBA): Examina el tráfico de red para identificar amenazas que generan tráfico inusual, como ataques de denegación de servicio ciertas formas de malware y violaciones a políticas de red.

4. Basados en Host (HIPS): Se efectúa mediante la instalación de paquetes de software que monitoriza un host único en busca de actividad sospechosa.

IDSS OPEN SOURCE

Una vez mostrado lo que es un IDS, sus características, tipos y demás, vamos a tratar sobre los distintos IDSs, más importante y los más utilizados, que existen de Open Source.

Trataremos sobre sus principales características, cuando se crearon, sus ventajas e inconvenientes, etc.

SNORT

Snort es una herramienta gratuita de prevención de intrusos (IPS) de red abierta. Fue creado por Martin Roesch en 1998. La principal ventaja de usar Snort es su capacidad para realizar análisis de tráfico en tiempo real y registro de paquetes en redes, así como capacidades completas de prevención de intrusiones. Snort es una herramienta para detectar gusanos variados, exploits, escaneo de puertos y otras amenazas maliciosas. Se puede

configurar en tres modos principales: sniffer, registrador de paquetes y detección de intrusión de red.

- En modo sniffer, el programa solo leerá paquetes y mostrará la información de la consola.
- En modo de registrados, los paquetes se registrarán en el disco.
- En modo de detección de intrusos, el programa controlará el tráfico en tiempo real y lo comparará con las reglas definidas por el usuario.

Snort es la herramienta gratuita mas descargada (4 millones de veces) y es probablemente el IPS más utilizado en todo el mundo. Es compatible con varias plataformas de hardware y sistemas operativos como Linux, OpenBSD, Solaris, HP-UX, MacOS, Windows, etc.

Ventajas

- Gratis para descargar y es Open Source.
- Fácil para escribir reglas para la prevención de intrusos.
- Altamente flexible y dinámico en términos de implementaciones en vivo.
- Buen soporte de la comunidad, para resolución de problemas.

Inconvenientes

- No tiene interfaz de usuario para la manipulación de la reglas.
- Lento en el procesamiento de paquetes de red.
- No se puede detectar una división de forma en varios paquetes TCP.

SURICATA

Suricata es un monitoreo de seguridad IPDS y red de alto rendimiento. De código abierto, rápido y altamente robusto desarrollado por Open Information Security Foundation (OISF). El motor es capaz de detectar intrusiones en tiempo real, prevenir intrusiones en línea y monitorear la seguridad de la red. Suricata consiste en algunos módulos como Capturar, Recolectar, Decodificar, Detectar y Producir. Captura el tráfico que pasa en un flujo antes de la decodificación, que es altamente óptimo. Pero a diferencia de Snort, configura

flujos separados después de la captura y especifica cómo se separará el flujo entre los procesadores.

Ventajas

- El procesamiento del tráfico de red en la séptima capa del modelo OSI que, a su vez, mejora su capacidad para detectar actividades de malware.
- Detecta y analiza automáticamente protocolos como IP, TCP, UDP, ICMP, HTTP, TLS, FTP, SMB y FTP para que las reglas se apliquen a todos los protocolos.
- Las características avanzadas consisten en multi-threading y aceleración GPU.

Inconvenientes

- Menos soporte en comparación con otros IDS como Snort.
- Complicado en la operación y requiere más recursos del sistema para un funcionamiento completo.

SECURITY ONION

Es una distribución de Linux que incluye herramientas de seguridad open source para la detección de intrusos, monitoreo de seguridad en la red y gestión de registros. La distro esta basada en Ubuntu y las herramientas que trae son: Kibana, Snort, Suricata, Bro, ELSA y muchas más.

Security Onion proporciona alta visibilidad y contexto al tráfico de la red, alertas y actividades sospechosas. Pero requiere una gestión adecuada por parte del administrador del sistema para revisar alertas, monitorear la actividad de la red y actualizar periódicamente las reglas de detección basadas en IDS. Security Onion tiene tres funciones principales:

- Captura de paquete completo.
- Sistemas de detección de intrusiones basados en host y basados en red.
- Potentes herramientas de análisis.

Ventajas

- Proporciona un entorno altamente flexible para que los usuarios ajusten la seguridad de la red según los requisitos.
- Consiste en herramientas de administración de sensores preinstaladas, analizadores de tráfico y rastreadores de paquetes, y puede operarse sin ningún software IDS / IPS adicional.
- Tiene actualizaciones periódicas para mejorar los niveles de seguridad.

Inconvenientes

- No es compatible con Wi-Fi para administrar la red.
- Requiere una labor de administración por parte del administrador para aprender varias herramientas para hacer un uso eficiente de la distribución.
- No hay copias de seguridad automáticas de los archivos de configuración, excepto las reglas; por lo tanto, se requiere el uso de software de terceros para esta actividad.

BRO NETWORK SECURITY MONITOR

Bro es una plataforma de seguridad open source desarrollado por Vern Paxson, y es utilizado para recopilar mediciones de red, realizar investigaciones forenses, establecer líneas de base de tráfico y mucho más.

BroIDS comprende un conjunto de archivos de registro de toda la actividad en la red como sesiones HTTP con URIs, key headers, MiME types, respuestas de servidor, peticiones de DNS, certificados SSL, sesiones SMTP, etc. Además, provee de funcionalidades sofisticadas para detección de malware, vulnerabilidades del software, ataques de fuerza bruta SSH y validación de cadenas de certificados SSL. BroIDS se divide en dos capas:

- Bro Event Engine: esto hace la tarea de analizar paquetes de tráfico de red en vivo o grabados utilizando C ++ para generar eventos cuando sucede algo inusual en la red.
- Bro Policy Scripts: estos analizan eventos para crear políticas de acción, y los eventos se manejan mediante scripts de políticas, como enviar correos electrónicos, generar alertas, ejecutar comandos del sistema e incluso llamar a números de emergencia.

Ventajas

- Muy flexible ya que BroIDS usa un lenguaje de scripting para permitir a los usuarios establecer reglas de monitoreo para cada objeto protegido.
- Trabaja de forma eficiente con redes de gran volumen de tráfico y maneja grandes proyectos de red.
- Capaz de analizar en profundidad el tráfico y admite analizadores para múltiples protocolos.

Inconvenientes

- Es difícil para manejar debido a su arquitectura compleja.
- Experiencia en programación para un manejo competente del sistema BroIDS.

OSSEC

OSSEC es un host basado en IDS, que realiza tareas variadas como el análisis de registros, la verificación de integridad, la supervisión del registro de Windows, la detección de rootkits, las alertas basadas en el tiempo y la respuesta activa. El sistema está equipado con una arquitectura centralizada y multiplataforma permitiendo multiples sistemas para ser monitorizado por un administrador.

El sistema OSSEC comprende por los siguientes componentes:

- Aplicación principal: Ossec es compatibles con Linux, Windows, Solaris y Mac.
- Windows agent: cuando OSSEC es instalado en ordenadores y servidores Windows.
- Interfaz web: GUI web para definir reglas y monitorear la red.

Ventajas

- Sistema multiplataforma que provee alertas en tiempo real y configurables.
- Administración centralizada, con y sin agentes.
- Se puede usar tanto para agentes de servidor como sin agentes de servidor.

Inconvenientes

- El proceso de actualización sobreescribe las reglas existentes con reglas listas para usar.
- Las claves de precompartimiento pueden ser problemáticas.
- El sistema operativo Windows solo es compatible con el modo servidor-agente.

Leer más: [REFDEMO1][RERDEMO2][REFDEMO3].

DEMOSTRACIÓN

Para la demostración hemos creado una máquina virtual con Ubuntu Server y vamos a instalarlo con Snort. Previamente, le hemos asignado un adaptador puente para poder realizar ataques sobre esa máquina y poder comprobar cómo funciona el IDS/IPS.

Tras la instalación del Ubuntu Server, procedemos a la instalación de todas las dependencias para la instalación correcta de Snort:

```
sudo apt install -y gcc libpcre3-dev zlib1g-dev libpcap-dev openssl
libssl-dev libnghttp2-dev libdumbnet-dev bison flex libdnet
```

Después de esto vamos a instalar el Snort.

INSTALACIÓN DESDE LA FUENTE

La instalación consiste en diversos pasos: descargar el código, configurarlo, compilarlo, instalarlo en el directorio apropiado, por último configurar las reglas de detección.

Comenzamos, creando un directorio de descarga en el directorio de Home y cambiarlo en él con esté comando:

```
mkdir ~/snort_src && cd ~/snort_src
```

Snort utiliza lo que es conocido como Data Acquisition Library (DAQ) para hacer llamadas abstractas a bibliotecas de captura de paquetes. Descargamos el último DAQ desde la página oficial de Snort con wget:

```
wget https://www.snort.org/downloads/snort/daq-2.0.6.tar.gz
```

Después de la descarga extraemos el código y saltamos al nuevo directorio:

```
tar -xvzf daq-2.0.6.tar.gz
cd daq-2.0.6
```

Arrancamos el script por los valores por defecto, y luego compilamos el programa con make y instalamos:

```
./configure && make && sudo make install
```

Con DAQ instalado, ahora podemos empezar con Snort, cambiando de directorio:

```
cd ~/snort_src
```

Después, descargamos la última versión del código de Snort con wget:

```
wget https://www.snort.org/downloads/snort/snort-2.9.11.1.tar.gz
```

Una vez descargado, lo extraemos y instalamos con `sourcefire` enabled, arrancamos make y instalamos:

```
tar -xvzf snort-2.9.11.1.tar.gz
cd snort-2.9.11.1
./configure --enable-sourcefire && make && sudo make install sudo
```

Con todo esto hecho, procedemos a la configuración.

CONFIGURANDO SNORT PARA ARRANCAR EN MODO NIDS

En la configuración de Snort, debemos de editar algunos archivos de configuración, descargar las reglas, y poner Snort de testeo.

Empezamos, actualizando las bibliotecas compartidas:

```
sudo ldconfig
```

Snort se instala en Ubuntu en el directorio /usr/local/bin/snort, es buena práctica, crear un enlace simbólico al /usr/sbin/snort.

```
sudo ln –s /usr/local/bin/snort /usr/sbin/snort
```

CONFIGURANDO LA ESTRUCTURA DE DIRECTORIOS

Creamos la estructura de directorios para guardar la configuración de Snort:

```
sudo mkdir –p /etc/snort/rules
sudo mkdir /var/log/snort
sudo mkdir /usr/local/lib/snort_dynamicrules
```

Establecemos los permisos de los nuevos directorios:

```
sudo chmod –R 5775 /etc/snort
sudo chmod –R 5775 /var/log/snort
sudo chmod –R 5775 /usr/local/lib/snort_dynamicrules
sudo chown –R snort:snort /etc/snort
sudo chown –R snort:snort /var/log/snort
sudo chown –R snort:snort /usr/local/lib/snort_dynamicrules
```

Creamos nuevos archivos para lista blanca, negra y las reglas locales:

```
sudo touch /etc/snort/rules/white_list.rules
sudo touch /etc/snort/rules/black_list.rules
sudo touch /etc/snort/rules/local.rules
```

Luego copiamos los archivo de configuración desde la carpeta de descarga:

```
sudo cp ~/snort_src/snort-2.9.11.1/etc/*.conf* /etc/snort
sudo cp ~/snort_src/snort-2.9.11.1/etc/*.map /etc/snort
```

A continuación, deberá descargar las reglas de detección que Snort seguirá para identificar posibles amenazas. Snort proporciona tres niveles de conjuntos de reglas, comunidad, registro y reglas de suscriptor.

- Las reglas comunitarias son de libre acceso, aunque un poco limitadas.
- Al registrarse gratis en su sitio web, obtiene acceso a su código Oink, que le permite descargar los conjuntos de reglas de usuarios registrados.
- Por último, las reglas de suscriptor son solo eso, disponible para los usuarios con una suscripción activa a los servicios de Snort.

Debajo, puede encontrar instrucciones para descargar las reglas de la comunidad o los conjuntos de reglas del usuario registrado.

OPCIÓN 1. USANDO REGLAS DE LA COMUNIDAD

Podemos descargar la reglas de la comunidad usando wget:

```
wget https://www.snort.org/rules/community -O ~/community.tar.gz
```

Extraemos las reglas y las copiamos a la carpeta de configuración:

```
sudo tar -xvf ~/community.tar.gz -C ~/
sudo cp ~/community-rules/* /etc/snort/rules
```

OPCIÓN 2. USANDO REGLAS DE LOS USUARIOS REGISTRADOS

En este caso debemos de registrarnos en la propia página de Snort y una vez hecho eso, nos proporcionará un código conocido como oinkcode. Podemos descargar las reglas sustituyendo <oinkcode> por nuestro código:

```
wget https://www.snort.org/rules/snortrules-snapshot-29111.tar.gz?
oinkcode=<oinkcode> -O ~/registered.tar.gz
```

Extraemos:

```
sudo tar -xvf ~/registered.tar.gz -C /etc/snort
```

Los conjuntos de reglas para los usuarios registrados incluyen una gran cantidad de reglas de detección preconfiguradas útiles. Si primero probó Snort con las reglas de la comunidad, puede habilitar reglas adicionales al quitar el comentario de sus inclusiones al final del archivo snort.conf.

CONFIGURANDO LA RED Y ESTABLECER LAS REGLAS

Con todo puesto en marcha, editamos snort.conf para modificar algunos parámetros:

```
sudo nano /etc/snort/snort.conf
```

Y editamos estos campos:

```
# Setup the network addresses you are protecting
ipvar HOME_NET <server public IP>/32

# Set up the external network addresses. Leave as "any" in most
situations
ipvar EXTERNAL_NET !$HOME_NET

# Path to your rules files (this can be a relative path)
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

# Set the absolute path appropriately
var WHITE_LIST_PATH /etc/snort/rules
var BLACK_LIST_PATH /etc/snort/rules

# unified2
# Recommended for most installs
output unified2: filename snort.log, limit 128
```

Por último, añadimos nuestras reglas y las de la comunidad:

```
include $RULE_PATH/local.rules
include $RULE_PATH/community.rules
```

Una vez hecho esto, guardamos el archivo de configuración.

VALIDANDO LA CONFIGURACIÓN

Podemos comprobar si nuestra configuración es correcta con el siguiente comando:

```
sudo snort -T -c /etc/snort/snort.conf
```

Y aparecerá el siguiente mensaje:

```
        --== Initialization Complete ==-

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.11.1 GRE (Build 268)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/
contact#team
           Copyright (C) 2014-2017 Cisco and/or its affiliates. All
rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.7.4
           Using PCRE version: 8.38 2015-11-23
           Using ZLIB version: 1.2.8

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.0
           Preprocessor Object: SF_DCERPC2  Version 1.0
           Preprocessor Object: SF_SSH  Version 1.1
           Preprocessor Object: SF_FTPTELNET  Version 1.2
           Preprocessor Object: SF_SDF  Version 1.1
           Preprocessor Object: SF_DNP3  Version 1.1
           Preprocessor Object: SF_REPUTATION  Version 1.1
           Preprocessor Object: SF_IMAP  Version 1.0
           Preprocessor Object: SF_SMTP  Version 1.1
           Preprocessor Object: SF_GTP  Version 1.1
           Preprocessor Object: SF_MODBUS  Version 1.1
           Preprocessor Object: SF_POP  Version 1.0
           Preprocessor Object: SF_DNS  Version 1.1
           Preprocessor Object: SF_SSLPP  Version 1.1
           Preprocessor Object: SF_SIP  Version 1.1

Snort successfully validated the configuration!
Snort exiting
```

PROBANDO LA CONFIGURACIÓN

Vamos a probar la configuración aplicando una regla sencilla, modificamos el siguiente archivo:

```
sudo nano /etc/snort/rules/local.rules
```

Y escribimos lo siguiente:

```
alert icmp any any -> $HOME_NET any (msg:"ICMP test"; sid:10000001;
rev:001;)
```

La regla consiste en las siguientes partes:

- Acción para el tráfico que coincida con la regla, alerta en este caso
- Protocolo de tráfico como TCP, UDP o ICMP como aquí
- La dirección de origen y el puerto, simplemente marcados como cualquiera para incluir todas las direcciones y puertos
- La dirección de destino y el puerto, $ HOME_NET como declarado en la configuración y cualquiera para el puerto
- Algunos bits adicionales
  - Mensaje de registro
  - Identificador de regla único (sid) que para las reglas locales debe ser 1000001 o superior
  - Número de versión de la regla.

Guarde las reglas locales y salga del editor.

Arrancamos Snort con opción -A para que puede imprimir las alertas por stdout. Para ello debemos escoger la interfaz de red correcta:

```
sudo snort -A console -i <interfaz> -u snort -g snort -c /etc/snort/
snort.conf
```

Ahora vamos hacer ping a la máquina y vemos los mensajes que muestran:

```
07/12-11:20:33.501624  [**] [1:10000001:1] ICMP test [**] [Priority:
0] {ICMP} 83.136.252.119 -> 80.69.173.202
```

Snort graba las alertar en un log /var/log/snort/snort.lo.<timestamp>, donde el timestamp es el instante cuando Snort fue arrancado en tiempo de Unix. Puedes leer los log con:

```
snort -r /var/log/snort/snort.log.<timestamp>
```

ARRANCAR SNORT EN SEGUNDO PLANO

Para arrancar Snort en Ubuntu como un servicio, necesitamos añadir el script de arranque:

```
sudo nano /lib/systemd/system/snort.service
```

Y añadimos lo siguiente:

```
[Unit]
Description=Snort NIDS Daemon
After=syslog.target network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/snort -q -u snort -g snort -c /etc/snort/
snort.conf -i eth0

[Install]
WantedBy=multi-user.target
```

Con el servicio definido, cargamos el demonio systemctl:

```
sudo systemctl daemon-reload
```

Arrancamos Snort con nuestra configuración con esto:

```
sudo systemctl start snort
```

Y comprobamos el estado del servicio de Snort:

```
sudo systemctl status snort
```

En definitiva, ya tenemos totalmente configurado nuestro servicio Snort y solamente bastaría con empezar a añadir reglas para detectar ataques y las distintas configuraciones para poder integrar está máquina con nuestro servidor, consiguiendo mayor protección.

ANÁLISIS (CONCLUSIÓN)

Un IDS es una herramienta más en la seguridad de una red. Su importancia reside en la necesidad de mantener la información que transmitimos segura y sin modificaciones ajenas a nuestra voluntad. Por lo tanto, es necesario implementar un sistema de detección de intrusos en cualquier empresa que trabaje con datos en la red y que tenga información que pueda ser objetivo de ataques.

Aunque cada día surgen nuevas formas de intentar entrar en redes ajenas y , por lo tanto, un IDS nunca podrá ser fiable al cien por ciento, este sistema es esencial en el conjunto de elementos de seguridad de cualquier red medianamente importante, ya que al complementarlo con otros sistemas de seguridad hacemos muy difícil para el atacante poder penetrar en nuestra red sin que lo sepamos.

BIOGRAFIA

- [REFINFO1]Sistema de detección de intrusos. En wikipedia

https://es.wikipedia.org/wiki/Sistema_de_detecci%C3%B3n_de_intrusos

- [REFINFO2]Sistema de detección de intrusiones (IDS). En CMM

http://es.ccm.net/contents/162-sistema-de-deteccion-de-intrusiones-ids

- [REFINFO3]Herramientas Opensource: IDS (Sistema de Detección de Intrusos). (22 febrero, 2017)

https://noticias.cec.es/index.php/2017/02/22/herramientas-opensource-ids-sistema-de-deteccion-de-intrusos/

- [REFINFO4]Proyecto Final de Carrera Implantación de un Sistema de Detección de Intrusos en la Universidad de Valencia. Emilio José Mira Alfaro

https://www.rediris.es/cert/doc/pdf/ids-uv.pdf

- [REFINFO5]Detección de Intrusos en Tiempo Real

https://www.segu-info.com.ar/proteccion/deteccion.htm

- [REFINFO6]Intrusion detection system (IDS). En searchsecuriy

http://searchsecurity.techtarget.com/definition/intrusion-detection-system

- [REFINFO7]Intrusion Detection Systems: Definition, Need and Challenges. SANS Institute InfoSec Reading Room

https://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-systems-definition-challenges-343

- [REFINFO8]Tipos de sistema de deteccion de intrusos. En Prezi (Terry Landeta, 6 de Diciembre de 2013)

https://prezi.com/x2nei7eawr4p/tipos-de-sistema-de-deteccion-de-intrusos/

- [REFINFO9]SISTEMA DE DETECCIÓN DE INTRUSOS. En monografias

http://www.monografias.com/trabajos11/intru/intru.shtml

- [REFINFO10]Intrusion Prevention System | Intrusion Detection System - [IPS IDS] Considerations for Info Sec. Youtube

https://www.youtube.com/watch?v=EJ7inytlS7M

- [REFINFO11]EVOLUCIÓN DE LOS SISTEMAS DE DETECCIÓN, PREVENCIÓN Y ANÁLISIS DE INCIDENTES. En revista.seguridad

https://revista.seguridad.unam.mx/numero-10/evoluci%C3%B3n-de-los-sistemas-de-detecci%C3%B3n-prevenci%C3%B3n-y-an%C3%A1lisis-de-incidentes

- [REFINFO12]Sistemas de detección de intrusos: un enfoque práctico (Antonio Villalón Huerta)

http://www.shutdown.es/uji.pdf

- [REFINFO13]The History and Evolution of Intrusion Detection. SANS Institute InfoSec Reading Room

https://www.sans.org/reading-room/whitepapers/detection/history-evolution-intrusion-detection-344

HERRAMIENTAS OPEN SOURCE Y ELABORACIÓN DEL DESARROLLO

- [REFDEMO1]https://opensourceforu.com/2017/04/best-open-source-network-intrusion-detection-tools/

- [REFDEMO2]https://www.esecurityplanet.com/network-security/10-open-source-security-breach-prevention-and-detection-tools.html

- [REFDEMO3]https://www.alienvault.com/blogs/security-essentials/open-source-intrusion-detection-tools-a-quick-overview

- [REFDEMO4]https://tools.kali.org/information-gathering/fragrouter

- [REFDEMO5]https://protegermipc.net/2017/02/22/mejores-ids-opensource-deteccion-de-intrusiones/

- [REFDEMO6]http://blog.thinkst.com/p/canarytokensorg-quick-free-detection.html

- [REFDEMO7]https://www.welivesecurity.com/la-es/2014/01/13/primeros-pasos-implementacion-ids-snort/