# Sistemas de Detección de Intrusos

## Snort y sus amigos

Javi Moreno

vierito5@gmail.com

http://vierito.es/wordpress

Seguridad y Redes

29 de julio de 2009 - Campus Party Valencia

# Índice

# Contents

# Objetivos

- Familiarizarnos con los Sistemas de Detección de Intrusos
- Conocer los distintos tipos de herramientas que existen
- Conocer qué aplicación debemos usar en cada caso
- Sacarles partido ;) Que no sean más una carga en lugar de una ayuda

# Terminología ¿'NIDS, HIDS, ...?

## IDS - Instrusion Detection System

Es un software diseñado para **detectar** intentos no deseados de acceder o manipular sistemas informáticos

## IPS - Instrusion Prevention System

Es un software diseñado para **detectar y actuar contra** intentos no deseados de acceder o manipular sistemas informáticos

## NIDS - Network Intrusion Detection System

Intenta detectar ataques o actividad no deseada mediante la monitorización y análisis del **tráfico de red**

## HIDS - Host Intrusion Detection System

Monitoriza y analiza la actividad **dentro del ámbito de una máquina** en lugar de en sus interfaces externos

# Terminología ¿'NIDS, HIDS, ...?

## IDS - Instrusion Detection System

Es un software diseñado para **detectar** intentos no deseados de acceder o manipular sistemas informáticos

## IPS - Instrusion Prevention System

Es un software diseñado para **detectar y actuar contra** intentos no deseados de acceder o manipular sistemas informáticos

## NIDS - Network Intrusion Detection System

Intenta detectar ataques o actividad no deseada mediante la monitorización y análisis del **tráfico de red**

## HIDS - Host Intrusion Detection System

Monitoriza y analiza la actividad **dentro del ámbito de una máquina** en lugar de en sus interfaces externos

# Según su comportamiento

## Pasivos

- Detecta posibles brechas de seguridad
- Genera un log
- Alerta
- Destinado a analizar mucho tráfico

## Activos/Reactivos

Los llamados **IPS**:

- Detecta posibles brechas de seguridad
- Genera un log
- Actua (bloqueo, +info, . . . )
- Alerta
- Destinado a analizar menos tráfico que un IDS

Muchas herramientas cuentan con ambos modos.

# Según su comportamiento

## Pasivos

- Detecta posibles brechas de seguridad
- Genera un log
- Alerta
- Destinado a analizar mucho tráfico

## Activos/Reactivos

Los llamados **IPS**:

- Detecta posibles brechas de seguridad
- Genera un log
- Actua (bloqueo, +info, . . . )
- Alerta
- Destinado a analizar menos tráfico que un IDS

Muchas herramientas cuentan con ambos modos.

# Según su comportamiento

## Pasivos

- Detecta posibles brechas de seguridad
- Genera un log
- Alerta
- Destinado a analizar mucho tráfico

## Activos/Reactivos

Los llamados **IPS**:

- Detecta posibles brechas de seguridad
- Genera un log
- Actua (bloqueo, +info, . . . )
- Alerta
- Destinado a analizar menos tráfico que un IDS

Muchas herramientas cuentan con ambos modos.

# Unos cuantos ejemplos

- Tripwire $\Longrightarrow$ HIDS
- AIDE $\Longrightarrow$ HIDS
- SNORT $\Longrightarrow$ NIDS/NIPS
- BRO $\Longrightarrow$ NIDS

# Según su área de actuación

## De red

- Contenido del tráfico
- Cantidad del tráfico
- Uso de puertos
- Escaneos de hosts

## De host

- Integridad de binarios
- Llamadas al sistemas
- Exploits de Buffer Overflow, Format String
- Escaneos de puertos del host

# Según su área de actuación

## De red

- Contenido del tráfico
- Cantidad del tráfico
- Uso de puertos
- Escaneos de hosts

## De host

- Integridad de binarios
- Llamadas al sistemas
- Exploits de Buffer Overflow, Format String
- Escaneos de puertos del host

# Posicionamiento de un IDS/IPS de red

En principio...

## IDS

- Offline
- Recibe copia del tráfico
- Generará un informe
- Generalmente monitorizan una subred/VLAN
- Método: TAPs, Port Mirroring,...

## IPS

- Inline
- El tráfico los atraviesa
- Discriminará el tráfico
- Puede bloquearlo
- Generalmente entre subredes

- En realidad la detección ataques es muy similar
- Se basan en reconocer firmas de ataques conocidos.

# Posicionamiento de un IDS/IPS de red

En principio...

## IDS

- Offline
- Recibe copia del tráfico
- Generará un informe
- Generalmente monitorizan una subred/VLAN
- Método: TAPs, Port Mirroring,...

## IPS

- Inline
- El tráfico los atraviesa
- Discriminará el tráfico
- Puede bloquearlo
- Generalmente entre subredes

- En realidad la detección ataques es muy similar
- Se basan en reconocer firmas de ataques conocidos.

# Posicionamiento de un IDS/IPS de red

En principio...

## IDS

- Offline
- Recibe copia del tráfico
- Generará un informe
- Generalmente monitorizan una subred/VLAN
- Método: TAPs, Port Mirroring,...

## IPS

- Inline
- El tráfico los atraviesa
- Discriminará el tráfico
- Puede bloquearlo
- Generalmente entre subredes

- En realidad la detección ataques es muy similar
- Se basan en reconocer firmas de ataques conocidos.

# Network TAP

¿Qué es?

- Es un dispositivo *hardware* que proporciona acceso al flujo de datos a través de la red
- Disponible para todo tipo de tecnologías
- Nos quedamos con una copia de los paquetes
- Transparente para los dos puntos A y B, entre los cuales 'pinchamos'

Ejemplos clarificadores:

- Vampire tap! (10BASE5)
- *empalmar* cables en un hub
- pinchar un teléfono (no exactamente...)

# Network TAP

¿Qué es?

- Es un dispositivo *hardware* que proporciona acceso al flujo de datos a través de la red
- Disponible para todo tipo de tecnologías
- Nos quedamos con una copia de los paquetes
- Transparente para los dos puntos A y B, entre los cuales 'pinchamos'

Ejemplos clarificadores:

- Vampire tap! (10BASE5)
- *empalmar* cables en un hub
- pinchar un teléfono (no exactamente...)

# Network TAP

¿Qué es?

- Es un dispositivo *hardware* que proporciona acceso al flujo de datos a través de la red
- Disponible para todo tipo de tecnologías
- Nos quedamos con una copia de los paquetes
- Transparente para los dos puntos A y B, entre los cuales 'pinchamos'

Ejemplos clarificadores:

- Vampire tap! (10BASE5)
- *empalmar* cables en un hub
- pinchar un teléfono (no exactamente...)

# Network TAP

Inconvenientes:

- hardware adicional $\Longrightarrow$ más caro que otras tecnologías
- red muy grande $\Longrightarrow$ muchos dispositivos
- dificultad con algunos canales full-duplex
- *(para un malo maloso)* colocarlo puede suponer un pequeño corte en la red

# Port Mirroring

¿Qué es?

- Dentro de un switch enviamos una copia los paquetes que pasan
- De un puerto o una VLAN al sistema monitorizador que está en otro puerto
- Llamado SPAN por Cisco

Esta sería la diferencia más clara entre un hub y un switch.

- En un hub el paquete que entra se retransmite a todos los puertos
- En un switch usaremos una tabla de MACs y enviaremos al puerto destino
- Esa capacidad de direccionamiento de un switch es configurable ;)

# Port Mirroring

¿Qué es?

- Dentro de un switch enviamos una copia los paquetes que pasan
- De un puerto o una VLAN al sistema monitorizador que está en otro puerto
- Llamado SPAN por Cisco

Esta sería la diferencia más clara entre un hub y un switch.

- En un hub el paquete que entra se retransmite a todos los puertos
- En un switch usaremos una tabla de MACs y enviaremos al puerto destino
- Esa capacidad de direccionamiento de un switch es configurable ;)

# Sistemas Distribuidos - DIDS

Se usan cuando hay demasiado ancho de banda a analizar por el IDS/IPS

- Sondas $\Longrightarrow$ sistemas detectores, recogen la información
- Sistema Central $\Longrightarrow$ reciben la información de las las sondas y la interpreta globalmente

Existen sistemas distribuidos donde ya se interpreta la información en las sondas y sólo se envían a la central información más susceptible, aprovechando recursos. *Ej. Videovigilancia (Siemens)*
Sistemas adaptables $\Longrightarrow$ inteligencia artificial, aprender firmas

# Sistemas Distribuidos - DIDS

- Todos los elementos del sistema han de *entenderse*. Diferentes estándares, tener que morir a un fabricante, ...
- Gran cantidad de información. Recibimos de muchas sondas.
- Inteligencia en las sondas y en la central. Correlación y constrastación de la información.
- Detección de ataques fragmentados

# Sistemas Distribuidos - DIDS

- Necesidad de una estandarización de alertas
- Integrar sensores de diferentes fabricantes
- CIDF (Common Intrusión Detection Framework)
- IDEF (Intrusión Detection Exchange Format)

# Filosofía de Uso

- Ha de convertirse en una herramienta **útil**
- **No (sólo) engrosar la lista de tareas** de un admin

- Una sola herramienta no es una solución buena
- Necesitamos un conjunto de herramientas que nos proporcionen una **perspectiva global**

- Cuanto más esfuerzo se invierta mejor será, siempre *hay algo más que poner*
- Generalmente cuanto más gráficos sean los resultados mejor se interpretan

# Filosofía de Uso

- Ha de convertirse en una herramienta **útil**
- **No (sólo) engrosar la lista de tareas** de un admin

- Una sola herramienta no es una solución buena
- Necesitamos un conjunto de herramientas que nos proporcionen una **perspectiva global**

- Cuanto más esfuerzo se invierta mejor será, siempre *hay algo más que poner*
- Generalmente cuanto más gráficos sean los resultados mejor se interpretan

# Filosofía de Uso

- Ha de convertirse en una herramienta **útil**
- **No (sólo) engrosar la lista de tareas** de un admin

- Una sola herramienta no es una solución buena
- Necesitamos un conjunto de herramientas que nos proporcionen una **perspectiva global**

- Cuanto más esfuerzo se invierta mejor será, siempre *hay algo más que poner*
- Generalmente cuanto más gráficos sean los resultados mejor se interpretan

# Filosofía de Uso

- Antes de un despliegue debemos plantearnos las necesidades, meditarlo bien.
- ¿**dónde** ponemos cada sonda IDS/IPS?
- ¿qué **anchos de banda** necesitamos cubrir?
- Reglas $\Longrightarrow$ **eficiencia**
- ¿qué **parametros extra** necesitamos controlar?
- el dinero manda $\Longrightarrow$ centrarnos en los **servicios expuestos a internet** o los que manejen **información más confidencial**

Exige una meditación y planificación previa importante. Hay que transmitir esa importancia a quien paga que para se implante el sistema.

# Filosofía de Uso

- Antes de un despliegue debemos plantearnos las necesidades, meditarlo bien.
- ¿**dónde** ponemos cada sonda IDS/IPS?
- ¿qué **anchos de banda** necesitamos cubrir?
- Reglas $\Longrightarrow$ **eficiencia**
- ¿qué **parametros extra** necesitamos controlar?
- el dinero manda $\Longrightarrow$ centrarnos en los **servicios expuestos a internet** o los que manejen **información más confidencial**

Exige una meditación y planificación previa importante. Hay que transmitir esa importancia a quien paga que para se implante el sistema.

# Filosofía de Uso

- Antes de un despliegue debemos plantearnos las necesidades, meditarlo bien.
- ¿**dónde** ponemos cada sonda IDS/IPS?
- ¿qué **anchos de banda** necesitamos cubrir?
- Reglas $\Longrightarrow$ **eficiencia**
- ¿qué **parametros extra** necesitamos controlar?
- el dinero manda $\Longrightarrow$ centrarnos en los **servicios expuestos a internet** o los que manejen **información más confidencial**

Exige una meditación y planificación previa importante. Hay que transmitir esa importancia a quien paga que para se implante el sistema.

# Filosofía de Uso

- Antes de un despliegue debemos plantearnos las necesidades, meditarlo bien.
- ¿**dónde** ponemos cada sonda IDS/IPS?
- ¿qué **anchos de banda** necesitamos cubrir?
- Reglas $\Longrightarrow$ **eficiencia**
- ¿qué **parametros extra** necesitamos controlar?
- el dinero manda $\Longrightarrow$ centrarnos en los **servicios expuestos a internet** o los que manejen **información más confidencial**

Exige una meditación y planificación previa importante. Hay que transmitir esa importancia a quien paga que para se implante el sistema.

# Filosofía de Uso

- Antes de un despliegue debemos plantearnos las necesidades, meditarlo bien.
- ¿**dónde** ponemos cada sonda IDS/IPS?
- ¿qué **anchos de banda** necesitamos cubrir?
- Reglas $\Longrightarrow$ **eficiencia**
- ¿qué **parametros extra** necesitamos controlar?
- el dinero manda $\Longrightarrow$ centrarnos en los **servicios expuestos a internet** o los que manejen **información más confidencial**

Exige una meditación y planificación previa importante. Hay que transmitir esa importancia a quien paga que para se implante el sistema.

# Filosofía de Uso

- Antes de un despliegue debemos plantearnos las necesidades, meditarlo bien.
- ¿**dónde** ponemos cada sonda IDS/IPS?
- ¿qué **anchos de banda** necesitamos cubrir?
- Reglas $\Longrightarrow$ **eficiencia**
- ¿qué **parametros extra** necesitamos controlar?
- el dinero manda $\Longrightarrow$ centrarnos en los **servicios expuestos a internet** o los que manejen **información más confidencial**

Exige una meditación y planificación previa importante. Hay que transmitir esa importancia a quien paga que para se implante el sistema.

# Filosofía de Uso

- Antes de un despliegue debemos plantearnos las necesidades, meditarlo bien.
- ¿**dónde** ponemos cada sonda IDS/IPS?
- ¿qué **anchos de banda** necesitamos cubrir?
- Reglas $\Longrightarrow$ **eficiencia**
- ¿qué **parametros extra** necesitamos controlar?
- el dinero manda $\Longrightarrow$ centrarnos en los **servicios expuestos a internet** o los que manejen **información más confidencial**

Exige una meditación y planificación previa importante. Hay que transmitir esa importancia a quien paga que para se implante el sistema.

# Filosofía de Uso

- Antes de un despliegue debemos plantearnos las necesidades, meditarlo bien.
- ¿**dónde** ponemos cada sonda IDS/IPS?
- ¿qué **anchos de banda** necesitamos cubrir?
- Reglas $\Longrightarrow$ **eficiencia**
- ¿qué **parametros extra** necesitamos controlar?
- el dinero manda $\Longrightarrow$ centrarnos en los **servicios expuestos a internet** o los que manejen **información más confidencial**

Exige una meditación y planificación previa importante. Hay que transmitir esa importancia a quien paga que para se implante el sistema.

# Filosofía

- Existen soluciones todo en uno: *firewall*, *VPN*, *IDS/IPS*, *antivirus*, *proxy*, *gestor de contenidos*
- Suelen ser comerciales
- Los dispositivos vienen con unas reglas predefinidas (firmas)
- Es necesario monitorizar durante un periodo de pruebas
- A partir de ahí afinamos la configuración.
- Tenemos varias formas de plantearlo...

# Filosofía

- Existen soluciones todo en uno: *firewall*, *VPN*, *IDS/IPS*, *antivirus*, *proxy*, *gestor de contenidos*
- Suelen ser comerciales
- Los dispositivos vienen con unas reglas predefinidas (firmas)
- Es necesario monitorizar durante un periodo de pruebas
- A partir de ahí afinamos la configuración.
- Tenemos varias formas de plantearlo...

# Filosofía

## Modo 1

- eliminar (reducir) alertas de falsos positivos, eventos de severidad baja
- alerta de ataques con un impacto fuerte
- anomaly detection
- Si es un ataque importante los administradores ya podrán en marcha su plan de respuesta

## Modo 2

- afinar los falsos positivos
- dejar todo tipo de eventos activos
- en caso de intrusión $\Longrightarrow$ mucha más información para un posterior análisis

# Filosofía

## Modo 1

- eliminar (reducir) alertas de falsos positivos, eventos de severidad baja
- alerta de ataques con un impacto fuerte
- anomaly detection
- Si es un ataque importante los administradores ya podrán en marcha su plan de respuesta

## Modo 2

- afinar los falsos positivos
- dejar todo tipo de eventos activos
- en caso de intrusión $\Longrightarrow$ mucha más información para un posterior análisis

# Filosofía

Son planteamientos opuestos El segundo está limitado por:

- mantenimiento
- almacenamiento
- rotados de logs

Pero es la única útil en caso de necesitar un análisis forense.

**La forma de juntarlo sería usar:**

- IPS: Anomaly detection
- IDS: logging pseudo-completo

# Filosofía

Son planteamientos opuestos El segundo está limitado por:

- mantenimiento
- almacenamiento
- rotados de logs

Pero es la única útil en caso de necesitar un análisis forense.

**La forma de juntarlo sería usar:**

- IPS: Anomaly detection
- IDS: logging pseudo-completo

# Filosofía

## Si nos basamos en reglas:

- base de datos de ataques y firmas **conocidos**
- el tráfico se analiza y se etiqueta directamente
- muy importantes las actualizaciones
- un ataque similar pero no idéntico pasaría

## Si nos basamos en detección de anomalías:

- se construye un perfil en función a un supuesto funcionamiento correcto
- permitirá detectar ataques no conocidos
- a partir de estos sistemas podemos generar firmas de ataques
- generalmente produce muchos falsos positivos

# Filosofía

## Si nos basamos en reglas:

- base de datos de ataques y firmas **conocidos**
- el tráfico se analiza y se etiqueta directamente
- muy importantes las actualizaciones
- un ataque similar pero no idéntico pasaría

## Si nos basamos en detección de anomalías:

- se construye un perfil en función a un supuesto funcionamiento correcto
- permitirá detectar ataques no conocidos
- a partir de estos sistemas podemos generar firmas de ataques
- generalmente produce muchos falsos positivos

# Contents

# SNORT

Es un NIDS Open Source Fue creado en principio como un simple sniffer Posteriormente se le añadieron módulos que permiten el procesado de los paquetes capturados y muchas cosas más.
Consta básicamente de cuatro módulos:

1. Sniffer
2. Preprocesador
3. Sistema detector
4. Sistema de salida

# Arquitectura de Snort

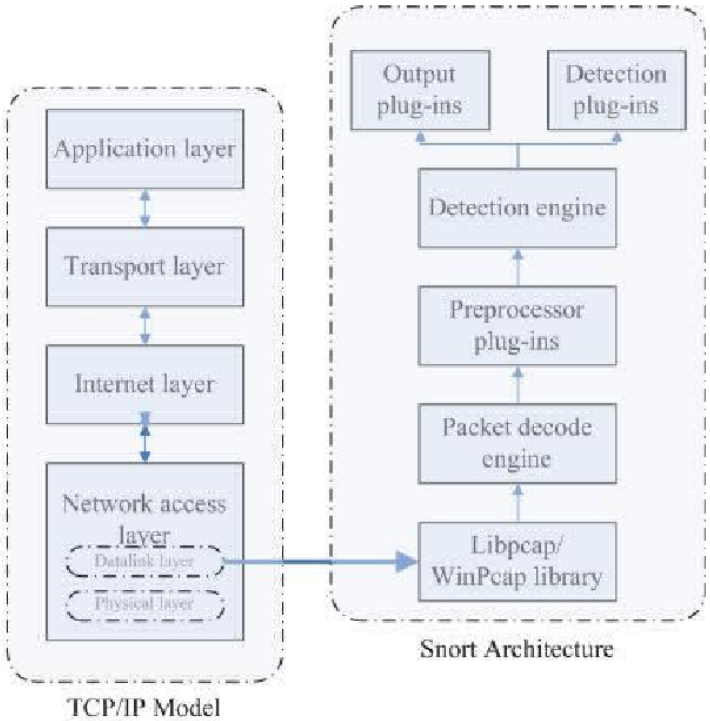


Figura: Arquitectura de Snort.

# Recursos necesarios

Cuanto más mejor, de todo, y punto
CPU, RAM, I/O, AODS

## Sobre recursos y anchos de banda

experiencias en la sala?

# El Sniffer

- Todo tipo de tráfico: IP, TCP, UDP, ICMP, protocolos de enrutamiento como RIP u OSPF, ...
- Puede generar logs en ficheros tipo tcpdump que luego podemos analizar en Wireshark o el propio tcpdump

# El Preprocesador

- Coge los paquetes y los analiza y/o modifica antes de enviarlos al sistema de detección
- Es necesario porque si no tenemos:
  - paquetes independientes
  - nada de conexiones TCP por ej.
  - paquetes fragmentados

El preprocesador tiene todo tipo de plugins

# Plugins del Preprocesador

- frag3: para manejar paquetes fragmentados. Sustituye al viejo frag2 con bastantes mejoras
- stream4: para mantener información del estado de las conexiones TCP
- sfPortscans: permite detectar distintos tipos de escaneos de puertos
- HTTPInspect: sesiones web
- SSH: detecta algunos exploits contra este servicio
- ...

# El Sistema de Detección

- Es donde se aplican una serie de reglas
- La cantidad de reglas van a influir seriamente en el rendimiento
- En base a ellas se etiquetan los paquetes

Y en aquí es donde está el meollo de la cuestión de los IDS/IPS

# El Sistema de Salida

- Encargado de almacenar los resultados
- Multitud de opciones distintas: syslog, dB mysql, ...

# Configuración en /etc/snort/snort.conf

- HOME_NET $\Longrightarrow$ La red local donde se instala Snort. Por ej, 192.168.1.024
- EXTERNAL_NET $\Longrightarrow$ La red externa, Internet por ej. Se puede usar any
- _SERVERS $\Longrightarrow$ Sirve para especificar a Snort dónde se encuentran ciertos servidores
- _PORTS $\Longrightarrow$ Como _SERVERS pero con los puertos de determinados servicios.
- RULE_PATH $\Longrightarrow$ Especifica el archivo de reglas que usará snort

Existen otras variables y podremos crear las nuestras con la sintaxis var VARIABLE y acceder mediante $VARIABLE

# Configuración en /etc/snort/snort.conf

- HOME_NET $\Longrightarrow$ La red local donde se instala Snort. Por ej, 192.168.1.024
- EXTERNAL_NET $\Longrightarrow$ La red externa, Internet por ej. Se puede usar any
- _SERVERS $\Longrightarrow$ Sirve para especificar a Snort dónde se encuentran ciertos servidores
- _PORTS $\Longrightarrow$ Como _SERVERS pero con los puertos de determinados servicios.
- RULE_PATH $\Longrightarrow$ Especifica el archivo de reglas que usará snort

Existen otras variables y podremos crear las nuestras con la sintaxis var VARIABLE y acceder mediante $VARIABLE

# Las reglas

Además de lo anterior deberemos especificarle los paths a las reglas, tantos como queramos

- include <path>

# Del Preprocesador

**frag3**

- existen diferentes pilas TCP/IP
- 2 tipo de config
- global $\Longrightarrow$ común
  - max_frags: paquetes fragmentados simultáneos
  - memcap: max mem en bytes
  - prealloc_frags: reservar mem para fragmentos

Ej. preprocessor frag3_global: prealloc_nodes 8192

# Del Preprocesador

- *engine* $\Longrightarrow$ específica de cada pila
  - timeout <segs>
  - min_ttl <valor>: no analizar paquetes que no llegan al destino
  - ttl_limit <saltos>: max dif entre fragmentos
  - detect_anomalies: detectar malformados
  - bind_to <lista IPs>: a qué IPs se aplica
  - policy <type>: tipo de máquina, *first, last, bsd, linux, ...*

  Ej. preprocessor frag3_global preprocessor frag3_engine: policy linux, detect_anomalies, bind_to [192.168.1.1,192.168.1.35]

# Del Preprocesador

- listas entre corchetes
- valores separados por comas
- podemos hacer uso de variables previamente definidas Ej. preprocessor frag3_engine: policy first, detect_anomalies, bind_to $Win_PDC

# Del Preprocesador

**stream4**

- detect_scans: detecta ciertos intentos
- disable_evasion_alerts
- detect_state_problems: como números de secuencia incorrectos
- max_sessions <num>: núm max de sesiones a analizar
- no_inspect: desactiva la inspección
- state_protection: protección DoS para el propio Snort

Existen muchas opciones y además otro plugin llamado *stream4_reassemble* con otras tantas opciones

# Del Preprocesador

**stream4**

- detect_scans: detecta ciertos intentos
- disable_evasion_alerts
- detect_state_problems: como números de secuencia incorrectos
- max_sessions <num>: núm max de sesiones a analizar
- no_inspect: desactiva la inspección
- state_protection: protección DoS para el propio Snort

Existen muchas opciones y además otro plugin llamado *stream4_reassemble* con otras tantas opciones

# Del Preprocesador

## sfPortscan

- Para detectar escaneos: TCP, UDP, IP Portscan, Decoy, distribuidos, barridos.
- Se recomienda desactivar las alertas de evasión
- Es **necesario** activar preprocessor flow Ej. preprocessor stream4: disable_evasion_alerts preprocessor flow: stats_interval 0 hash 2
- proto <protocolo>: TCP, UDP, IGMP, ip_proto, all, ...
- scan_type <level>: portscan, portsweep, decoy_portscan, all, ...
- watch_ip < ip1—ip2/cidr[:[port—port2-port3]]>
- ignore_scanners <lista IPs>, ignore_scanned <lista IPs>
- logfile <fichero>

# Del preprocesador

sfPortscan espera los parámetros entre llaves

Ej. preprocessor sfportscan: proto { all }, memcap { 10000000 }, sense_level { low }, ignore_scanners { $ADMIN_IP }

# Del Preprocesador

## HTTPInspect

- es de nivel de aplicación, transacciones http
- configuración global y específica para cada servidor
- la global aparece 1 sola vez en snort.conf
  Ej1. preprocessor http_inspect: global iis_unicode_map <map_filename> codemap <integer> [detect_anomalous_servers] [proxy_alert]
  Ej2. preprocessor http_inspect: global iis_unicode_map unicode.map 1252 proxy_alert
- la configuración del server a su vez es de 2 tipos: individual o por defecto
- con la opción *profile* podemos especificar que server web es
  Ej. preprocessor http_inspect_server: server default profile all ports { 80 }

# Del Preprocesador

## HTTPInspect

- es de nivel de aplicación, transacciones http
- configuración global y específica para cada servidor
- la global aparece 1 sola vez en snort.conf
  Ej1. preprocessor http_inspect: global iis_unicode_map <map_filename> codemap <integer> [detect_anomalous_servers] [proxy_alert]
  Ej2. preprocessor http_inspect: global iis_unicode_map unicode.map 1252 proxy_alert
- la configuración del server a su vez es de 2 tipos: individual o por defecto
- con la opción *profile* podemos especificar que server web es
  Ej. preprocessor http_inspect_server: server default profile all ports { 80 }

# Del Preprocesador

## SSH

- detectar algunos exploits: Gobbles, CRC32, Secure CRT y Protocol Mismatch
- server_ports { port1 port2 port3}
- max_encrypted_packets: alertar de Gobbles y CRC32
- max_client_bytes: alertar de Gobbles y CRC32
- autodetect: el protocolo
- disable_gobbles, disable_srvoverflow, disable_proto_mismatch, ...
- disable_paysize
- disable_recognition: detectar tráfico no SSH en puertos SSH

# Sistema de Detección - Las Reglas

- Por defecto, snort incluye un conjunto de reglas
- Conviene tenerlas actualizadas
- Hay conjuntos de reglas de pago, suscripción en `www.snort.org`
- Reglas de la comunidad
- `http://www.emergingthreats.net/`
- Nuestras propias reglas
- Existe un lenguaje para ello

# Sistema de Detección - Las Reglas

- Ej1. alert tcp $HOME_NET any -> any 6667:6669 ( msg: “IRC! IRC!”; )
- Ej2. alert tcp $HOME_NET any -> any 6667:6669 ( msg: “IRC! IRC!”;sid: 1000001;) alert tcp $HOME_NET any -> any 1024: ( msg: “Acceso a puertos elevados!”;sid: 1000002;)
- Veamos ahora la forma de los logs:
  - 02/18-14:01:35.899436 [**] [1:1000002:0] Acceso a puertos elevados! [**] [Priority: 0] {TCP} 192.168.1.3:41804 -> 207.46.27.34:1863

# Sistema de Detección - Las Reglas

- Ej1. alert tcp $HOME_NET any -> any 6667:6669 ( msg: “IRC! IRC!”; )
- Ej2. alert tcp $HOME_NET any -> any 6667:6669 ( msg: “IRC! IRC!”;sid: 1000001;) alert tcp $HOME_NET any -> any 1024: ( msg: “Acceso a puertos elevados!”;sid: 1000002;)
- Veamos ahora la forma de los logs:
  - 02/18-14:01:35.899436 [**] [1:1000002:0] Acceso a puertos elevados! [**] [Priority: 0] {TCP} 192.168.1.3:41804 -> 207.46.27.34:1863

# Sistema de Detección - Las Reglas

- Ej1. alert tcp $HOME_NET any -> any 6667:6669 ( msg: “IRC! IRC!”; )
- Ej2. alert tcp $HOME_NET any -> any 6667:6669 ( msg: “IRC! IRC!”;sid: 1000001;) alert tcp $HOME_NET any -> any 1024: ( msg: “Acceso a puertos elevados!”;sid: 1000002;)
- Veamos ahora la forma de los logs:
  - 02/18-14:01:35.899436 [**] [1:1000002:0] Acceso a puertos elevados! [**] [Priority: 0] {TCP} 192.168.1.3:41804 -> 207.46.27.34:1863

# Sistema de Detección - Las Reglas

Podremos definir qué hacer con las coincidencias:

- accion: alert, pass, log, activate, dynamic
- acción: activate y dynamic permiten activar reglas en función de otras
- protocolos, direcciones, operador de direcciones (sentido)
- metadatos: mensajes, identificador, prioridad
- contenido: análisis del contenido de los paquetes (opcional *nocase*) Ej. alert tcp $HOME_NET any -> any 80 (content: "porn"; msg:"Guarrete a la vista!";sid: 1000001;) alert tcp $EXT_NET any -> $HOME_NET any ( content: "|909090|";msg:"NOPs en el tráfico!";sid: 1000002;)
- detección: TTL, campo ID, ...
- post-detección: registrar aparte, poner tags, reaccionar

# Sistema de Salida

- Recoger alertas y almacenarlas
- De nuevo, plugins
- Para una base de datos
  - Ej. output database: <log | alert>, <database type>, <parameter list>
  - lista de parámetros: dbname, user, password, host, port, detail (fast, full), enconding (hex, ascii, base64), sensor_name
  - output database: alert, mysql, user=snort password=password dbname=snort host=localhost

# Snort in-line

- Hasta ahora hemos visto opciones para trabajar como IDS
- Tendríamos una serie de reglas extras para funcionar como IPS
- funcionamiento conjunto con iptables, drop, reject, sdrop

# Oinkmaster

Oinkmaster

- Rule Management
- Facilita la actualización de reglas desde `www.snort.org`
- También con las community, otras third-party y propias

# BASE

**BASE - Basic Analysis and Security Engine**

- Web-based front-end para Snort
- `http://base.secureideas.net/screens.php`

# Contents

# Time Machine

## Time Machine

- Es como un Wayback Machine de lo ocurrido en la red
- `http://www.net.t-labs.tu-berlin.de/research/tm/`
- Target: analizar tráfico ya ocurrido de un enlace a 10, 20Gbps
- MUCHÍSIMA INFORMACIÓN, no se puede guardar todo
- Sólo vamos a almacenar la información que consideremos significante
- Información indexada
- Basado en BRO IDS, tcpdump

# OSSIM

**Open Source Security Information Management**

- Pretende hacer una compilación de herramientas
- Complementar un sistema IDS/IPS
- Facilitar la administración
- Aprovechar mejor las features de los sistemas
- Ya hemos dicho que una sola herramienta no es la solución

# OSSIM

- Herramientas:
  - Arpwatch: anomalías y cambios en MACs
  - P0f: detección pasiva de OS y cambios de OS
  - Pads: anomalías en servicios
  - Snort, el IDS ;)
  - Nessus: identificación de vulnerabilidad, correlación cruzada (juntar el IDS con un escaneador de vulns)
  - Space: the statistical packet anomaly detection engine. Nos podría proporcionar algo de heurística
  - Tcptrack: datos de sesiones, correlación
  - Ntop: información sobre el tráfico, gráficas, estadísticas.
  - Nagios: información de servicios y hosts
  - Osiris: HIDS
  - OSSEC: integridad, rootkits, etc.
  - OCS-NG: inventario

# Portsentry

**Portsentry**

- Detecta escaneos de puertos
- Responde con bloqueos
- Puede servirnos como honeypot sencillo y eficaz
- Listas de puertos TCP, UDP

# Hping

**Hping**

- Analiza y desensambla paquetes TCP/IP
- En realidad es más útil para el otro lado $\Longrightarrow$ análisis de la configuración de firewall y cortafuegos

# Tripwire - AIDE

## Tripwire - AIDE

- Tripwire y AIDE (su sucesor) son HIDS
- chequeo de integridad
- chequeo de atributos

# Y así una gran lista que complementan . . .

- mod_sec
- phpids
- green-sql
- apache-scalp
- . . .

# Contents

# Conclusiones

## Conclusiones

- Disponemos de todo tipo de herramientas
- **En conjunto tienen un valor añadido**
- Es necesario una **planificación previa** de las necesidades
- Dedicarle recursos humanos y económicos para crear un sistema completo y eficaz
- Plantear un **sistema acorde con los recursos disponibles**
- Hay mucho software libre disponible de calidad
- **Paciencia**, hay mucho que conocer y configurar

# :wq!

## ¿preguntas?

vierito5@gmail.com

http://vierito.es/wordpress

Seguridad y Redes