Sistemas de detección de intrusos

Antonio Villalón Huerta

toni@aiind.upv.es

Mayo, 2005

Índice de materias

Introducción

- Conceptos básicos
- Historia
- Clasificación
- Arquitectura

Introducción: Conceptos básicos

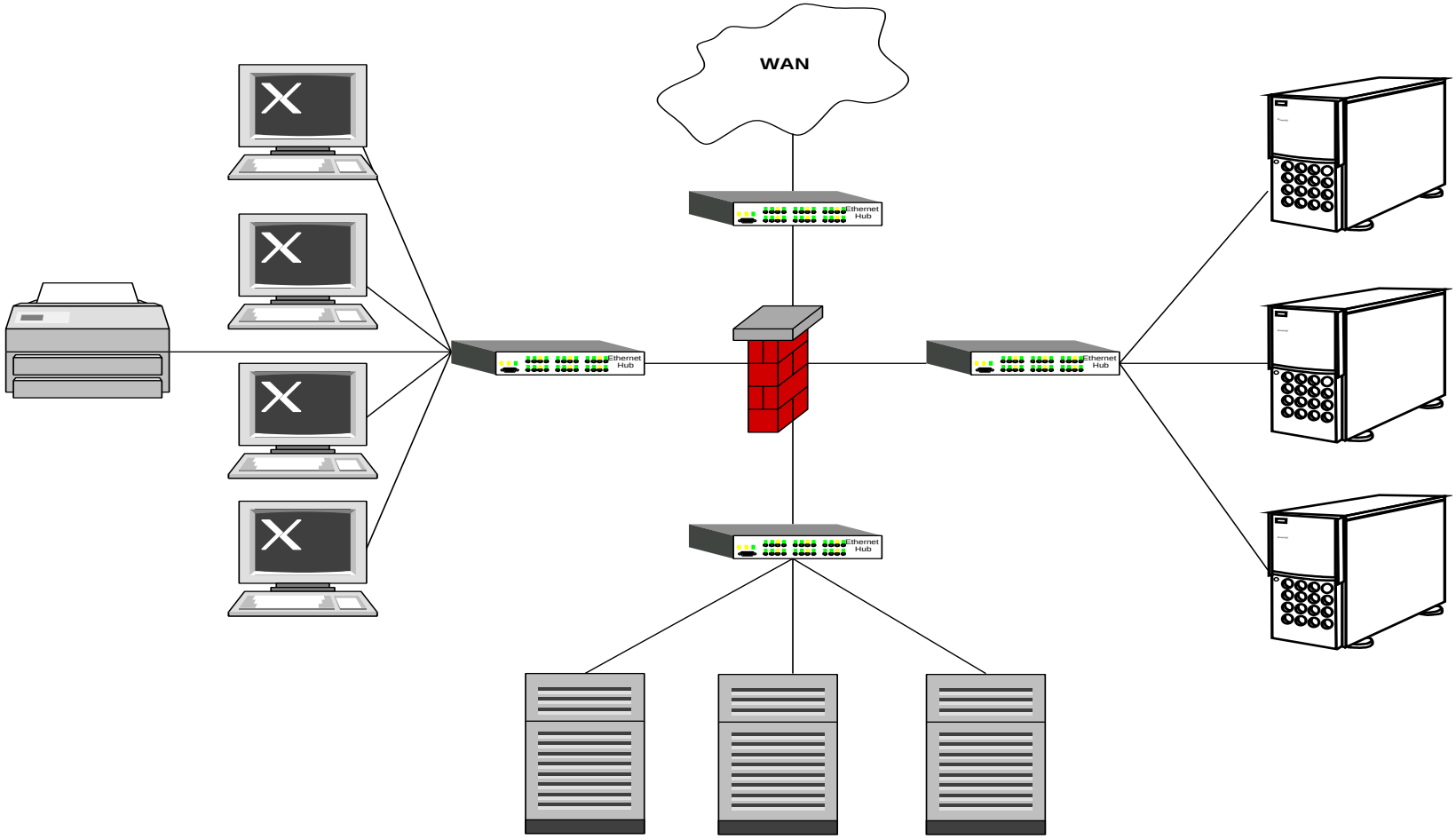
La **seguridad** de la información se define como la preservación de...

- ...su **confidencialidad**: sólo quienes estén autorizados pueden acceder a la información.
- ...su **integridad**: la información y sus métodos de proceso son exactos y completos.
- ...su **disponibilidad**: los usuarios autorizados tienen acceso a la información y los sistemas que la tratan siempre que lo requieran

Introducción: Conceptos básicos

- Debemos considerar la información y los sistemas que la tratan como **activos** de una organización: elementos que dan valor a la misma, y que por tanto hay que salvaguardar.
- Los activos presentan **vulnerabilidades** (debilidades de cualquier tipo que comprometen su seguridad).
- Por definición existen **amenazas** (escenarios en los que un evento o acción, deliberada o no, compromete la seguridad de un activo).
- Si un sistema presenta una vulnerabilidad y existe una amenaza, aparece un **riesgo** asociado (probabilidad de que el evento se materialice).
- Cuando el riesgo se materializa, la medida del daño causado se denomina **impacto**.

¿Qué queremos proteger?



Introducción: Conceptos básicos

¿Cómo protegerlo?

A las medidas que eliminen la vulnerabilidad o la amenaza, o disminuyan el riesgo o impacto asociados, se les denomina **defensas**, **salvaguardas**, **controles**...

- Políticas, normativas, procedimientos...
- Formación.
- ...
- Controles técnicos:
  - Cortafuegos.
  - Antivirus.
  - Analizadores de vulnerabilidades.
  - ...
  - **Sistemas de detección de intrusos.**

Introducción: Conceptos básicos

Algunas definiciones

- **Intrusión:** Conjunto de acciones que intentan comprometer la integridad, confidencialidad o disponibilidad de un recurso.
  - No sólo penetraciones contra un sistema.
- **Detección de intrusos:** Análisis automático de parámetros que modelan la actividad de un entorno con el propósito de detectar e identificar intrusiones.
- **Sistema de detección de intrusos (IDS):** Mecanismo de seguridad que lleva a cabo la detección de intrusos.
  - No tiene por qué ser un programa o producto concreto.

Introducción: Conceptos básicos

¿Por qué un IDS?

- La prevención no siempre es suficiente.
  - Los sistemas cortafuegos no son mágicos.
- Proporcionan conocimiento del entorno.
- Detección rápida de actividades sospechosas...
- ...¡y respuesta ante ellas!
- Registro adicional de incidentes (¿Pruebas judiciales?).
- ...

NOTA: Ningún mecanismo de seguridad *sustituye a*, sino que trabaja *junto a*.

Introducción: Historia

Década de los ochenta

- Primer trabajo sobre IDSes: 1980 (James P. Anderson).
- 1984–1986: diseño del primer sistema (IDES, *Intrusion Detection Expert System*), que funcionaba en tiempo real (Dorothy Denning, Peter Neumann).
- Detección orientada al *host*.

  $\Rightarrow$ Excepción: NSM, *Network System Monitor*, Universidad de California.
- Fuerte implicación militar en todos los proyectos.

Introducción: Historia

Década de los noventa

- Crecimiento exponencial de las redes de ordenadores: se orienta el trabajo hacia la detección en red.
- Aparecen los primeros productos comerciales (1990).
- Auge desde 1995 (crisis de los *firewalls*).

Introducción: Historia

Actualmente...

- Uno de los campos con más investigación y avances:
  - Correlación de eventos.
  - Aprendizaje automático.
  - Minería de datos.
  - ...
- Multitud de productos comerciales.
- Entornos complejos:
  - Orientación hacia sistemas distribuidos.
  - Intentos por facilitar la gestión.
  - ...

Introducción: Clasificación

Clasificación de los IDSes

- En función del origen de los datos a analizar:
  - IDSes basados en máquina (HIDS, *Host–based IDS*).
  - IDSes basados en red (NIDS, *Network–based IDS*).
- En función de la técnica de análisis utilizada:
  - Detección de anomalías (*Anomaly detection*).
  - Detección de usos indebidos (*Misuse detection*).
- Otras clasificaciones:
  - Tiempo real vs. periódicos.
  - Activos vs. pasivos.
  - Centralizados vs. distribuidos.
  - ...

Introducción: Clasificación

Clasificación → Origen de datos → HIDS

- Sólo procesa datos asociados a un recurso.
- Tipos:
  - Verificadores de integridad del sistema (SIV, *System Integrity Verifiers*).
  - Monitores de registros (LFM, *Log File Monitors*).
  - Sistemas de decepción (*Deception Systems*).

Introducción: Clasificación

Clasificación → Origen de datos → NIDS

- Procesa datos asociados a varios recursos.
- No tienen por qué estar ubicados en **toda** la red (de hecho casi ningún NIDS lo está).
- En la actualidad, los más utilizados.

Introducción: Clasificación

Clasificación → Técnica de análisis → *Anomaly Detection*

- IDEA: Una intrusión es una anomalía. Conozco lo que es 'normal' en un sistema, para poder detectar lo que no lo es (anomalías).
- PROBLEMA: La modelización del comportamiento es **muy compleja**.
- Sistemas expertos o aprendizaje automático (redes neuronales, reconocimiento geométrico...).
- Poco utilizados en sistemas reales.

Introducción: Clasificación

Clasificación → Técnica de análisis → *Misuse Detection*

- IDEA: No conozco lo que es 'normal' en un sistema, sino que directamente conozco lo anormal y lo puedo detectar.
- PROBLEMA: Sólo detecto los ataques que conozco.
- Diferentes aproximaciones. La más habitual: *pattern matching* (cada intrusión tiene un patrón asociado).
- En la actualidad, los más utilizados en sistemas reales.

Introducción: Clasificación

Clasificación → Otros

- Tiempo real vs. Periódicos:

  ⇒ El IDS trabaja permanentemente o a intervalos.

- Activos vs. Pasivos:

  ⇒ El IDS es capaz de iniciar una respuesta automática o simplemente informa.

- Centralizados vs. Distribuidos:

  ⇒ El IDS y la lógica asociada al mismo se implementan en un único sistema o se distribuyen en una red.

Introducción: Arquitectura

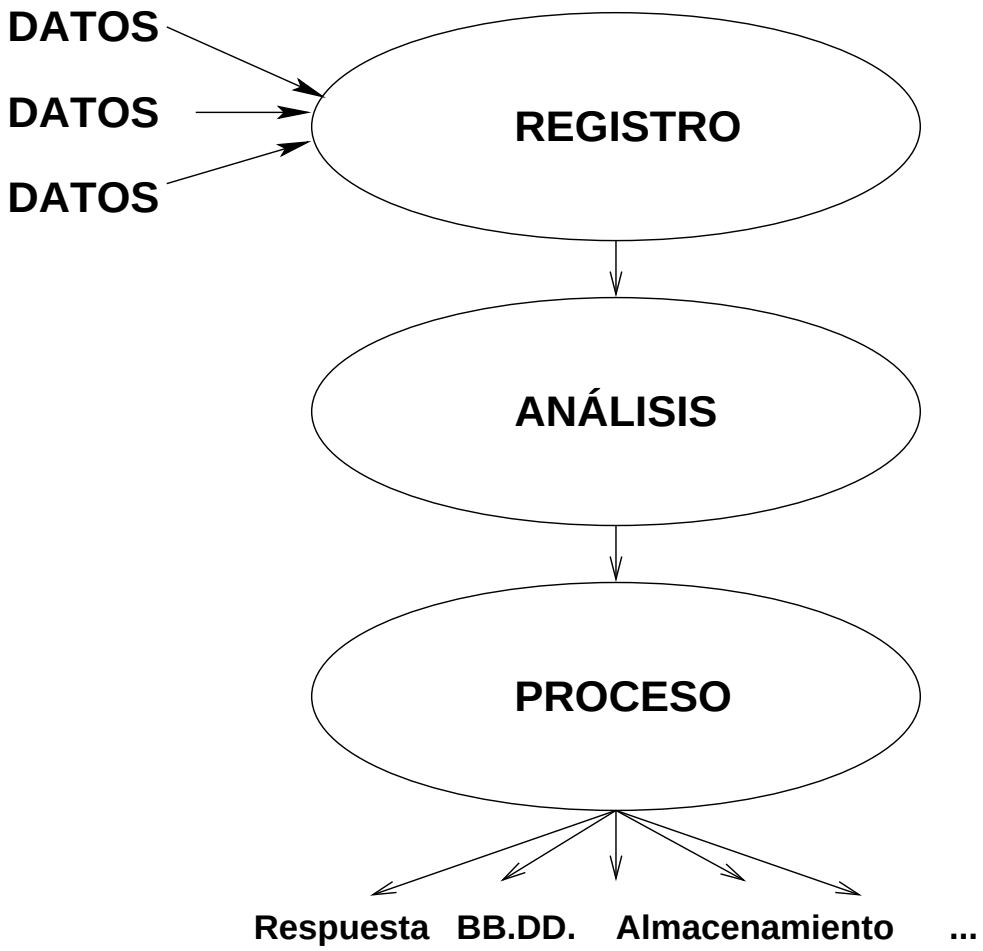
Premisa: Las actividades de un entorno informático pueden ser modelizadas y monitorizadas.

Tres elementos clave en un IDS:

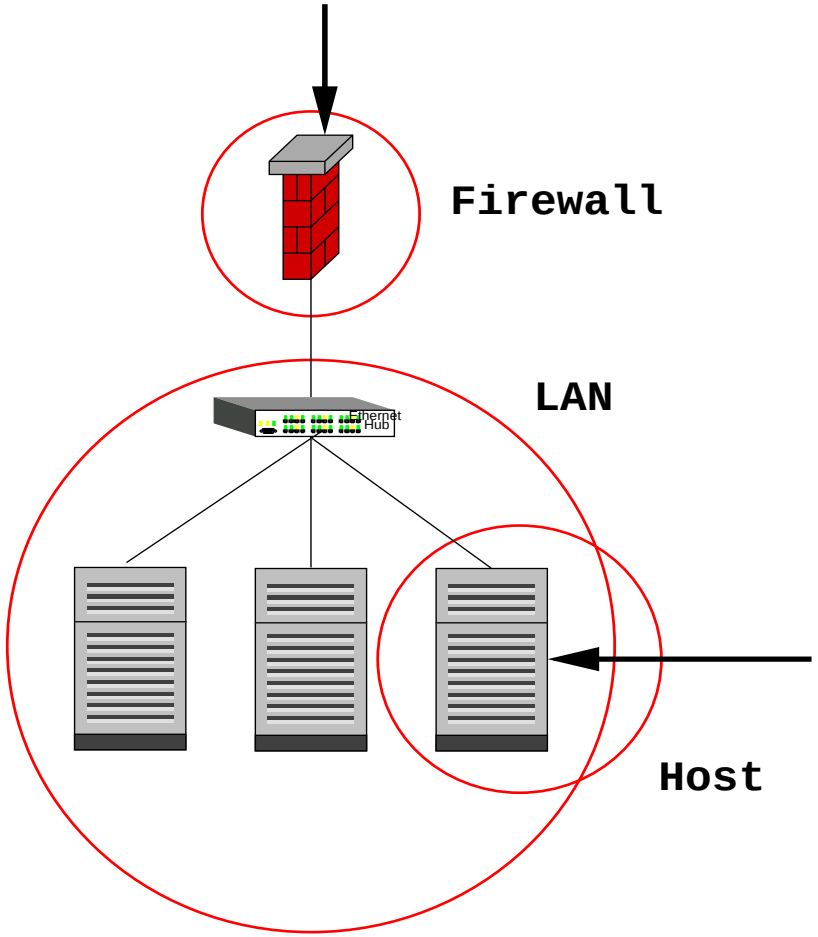
- Fuente de datos que genere eventos significativos del entorno (sensor).
- Motor de análisis de los datos registrados (árbitro).
- Procesador de los resultados del análisis (actor).

Introducción: Arquitectura

Gráficamente...



¿Dónde detectar?



Introducción: Arquitectura

Diseño: propiedades cuantitativas

- Tasa de falso positivo (FPR, *False Positive Rate*).
  $\Rightarrow$ Frecuencia con la que el IDS informa de actividades maliciosas que en realidad no lo son.
- Tasa de falso negativo (FNR, *False Negative Rate*).
  $\Rightarrow$ Frecuencia con la que el IDS no informa de actividades maliciosas reales.
- Tasa de error cruzado (CER, *Crossover Error Rate*).
  $\Rightarrow$ Frecuencia en la que FPR=FNR.
  $\Rightarrow$ Medida común para evaluar IDSes.

Introducción: Arquitectura

Diseño: propiedades cualitativas

- Mecanismos **automatizados**: no requieren supervisión en su funcionamiento habitual.
- **Aceptabilidad** por parte de usuarios y administradores.
- **Adaptabilidad** ante nuevas situaciones.
  $\Rightarrow$ ¡Escalabilidad!
- **Tolerancia** a fallos.
- ...

Técnicas de análisis

- Introducción
- Detección de anomalías
- Detección de usos indebidos

Análisis: Introducción

Introducción

- El primer paso para detectar una intrusión es el **análisis** de ciertos datos... ¿Cuáles?
  - Patrones de tráfico sospechosos.
  - Actividades no habituales de un usuario.
  - Consumo excesivo de ancho de banda.
  - Degradación en el rendimiento de un sistema.
  - ...
- Dos grandes técnicas de análisis:
  - Detección de anomalías (*Anomaly Detection*).
  - Detección de usos indebidos (*Misuse Detection*).

Análisis: Detección de anomalías

- Toda intrusión se puede considerar una **anomalía** (uso o comportamiento anormal de un entorno).

  $\Rightarrow$ Si puedo detectar anomalías, detecto también las intrusiones.

- Necesito conocer el comportamiento 'normal' de un entorno para detectar desviaciones con respecto al mismo.
- Línea de investigación muy atractiva.
- Entornos 'reales': problemática.

Análisis: Detección de anomalías

¿Cómo conocer la normalidad?

- Métodos estadísticos.
- Especificaciones de normalidad.

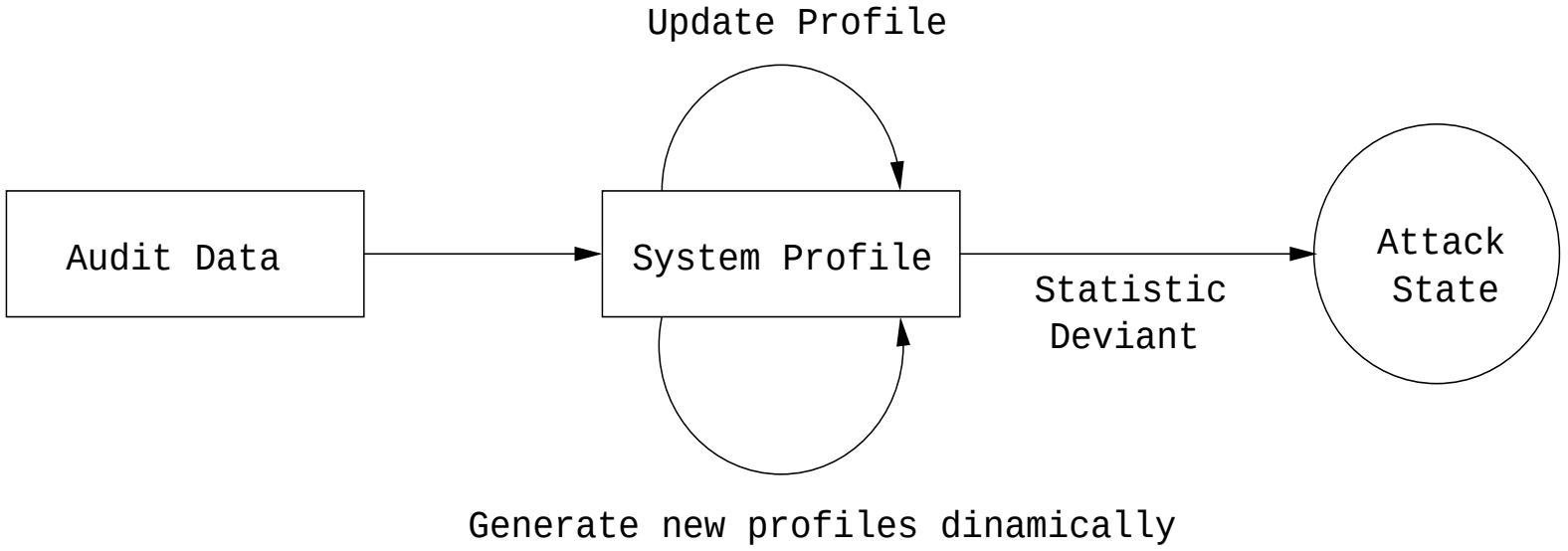
Análisis: Detección de anomalías

Métodos estadísticos

- Observamos y medimos la actividad de los actores sobre los objetos de un entorno.
- Generamos un 'perfil de normalidad', perfil dinámico que se irá actualizando con nueva información.
- Las actividades en un momento dado se almacenan en otro perfil, **perfil actual** (*current profile*).
- Comparamos ambos en busca de desviaciones.

Análisis: Detección de anomalías

Gráficamente...



[*An introduction to intrusion detection.* Aurobindo Sundaram, ACM Crossroads Student Magazine, 1996]

Análisis: Detección de anomalías

¿Qué medimos para generar perfiles?

- Ratios de progreso de ciertas actividades en el entorno, para determinar su **intensidad**.
- Medidas de la actividad representadas en forma **numérica**.
- Medidas de la **distribución** de una actividad con respecto a otras ('categorías').
- Distribución de **registros** de auditoría (*logs*).

Análisis: Detección de anomalías

Especificaciones de normalidad:

- Indicamos mediante un conjunto de reglas el comportamiento normal del entorno.
- **Premisa:** Es posible describir el comportamiento normal mediante gramáticas.
- Consideramos anomalías las desviaciones con respecto a la gramática.
- Nombre propio: *Specification–based anomaly detection.*

Análisis: Detección de anomalías

Problemas:

- Es complejo especificar lo 'normal', y mucho más aprenderlo.
- Velocidad del aprendizaje: rapidez vs. lentitud.

Ventajas:

- Detección de nuevos ataques (*novel attacks*).

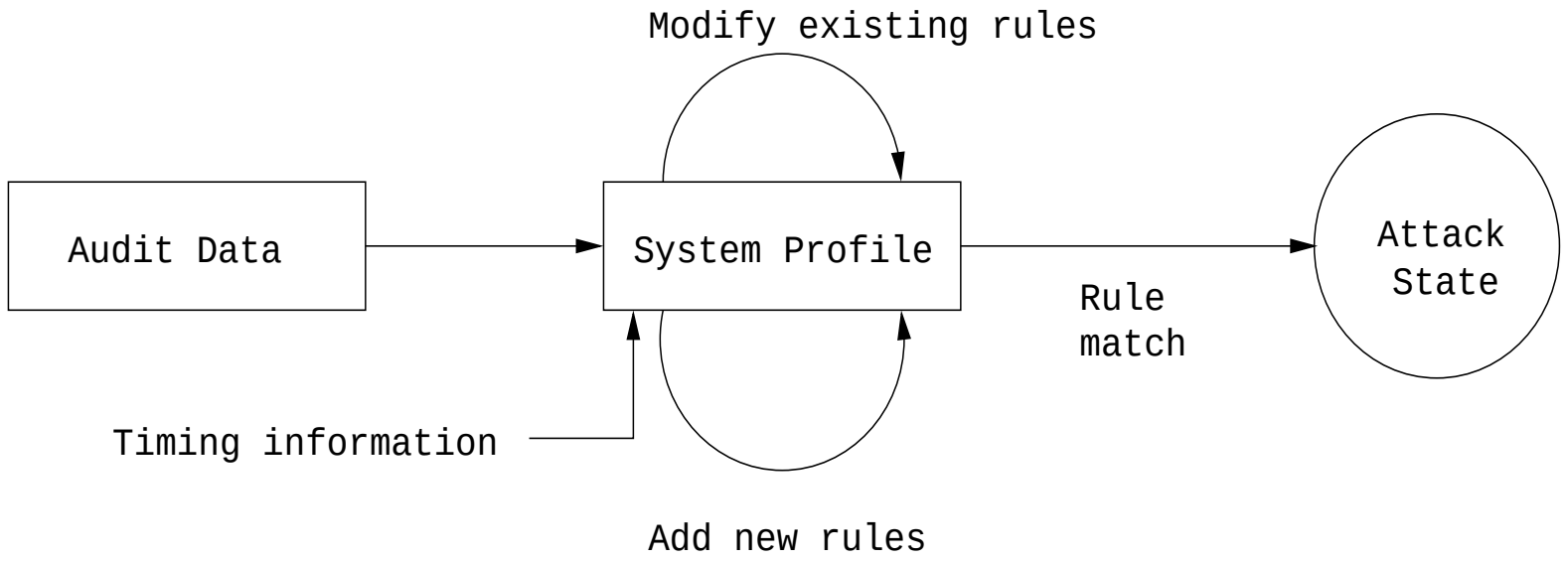
Análisis: Detección de usos indebidos

Introducción

- Conozco las posibles intrusiones contra un entorno (y soy capaz de describirlas).
- ¿Cómo las puedo describir?
  - Sistemas expertos.
  - Transición entre estados.
  - Comparación y emparejamiento de patrones *(Pattern Matching)*.
  - Modelos.
- Una vez he descrito las intrusiones, simplemente monitorizo y espero a que sucedan.

Análisis: Detección de usos indebidos

Graáficamente...



[*An introduction to intrusion detection.* Aurobindo Sundaram, ACM Crossroads Student Magazine, 1996]

Análisis: Detección de usos indebidos

Sistemas expertos

- Intrusiones codificadas como reglas *if–then–else*.
- Cada regla detecta eventos únicos o secuencias de eventos.
- Generalmente se basan en el análisis de registros de auditoría.
- Si se cumple la condición especificada, se ejecuta una acción concreta.

Análisis: Detección de usos indebidos

Análisis de transición entre estados

- Estado: imagen de parámetros del sistema en un momento determinado.
- Una intrusión se considera una secuencia de eventos que conducen de un estado inicial a un estado de intrusión consumada.
- Estado inicial inmediatamente posterior al inicio de la intrusión.
- Estado final: intrusión completada en su totalidad.
- Si identifico los estados intermedios, puedo detener la intrusión antes de que sea consumada.

Análisis: Detección de usos indebidos

Pattern Matching

- **Premisa:** El sistema está comprometido cuando recibe el patrón de una intrusión (no importa el estado inicial).
- Especifico los patrones de intrusión y espero a que sucedan.
- Cuando detecto uno de estos patrones, tomo una decisión.
- Aproximación más usada en los sistemas 'reales'.

Análisis: Detección de usos indebidos

Detección basada en modelos

- Conceptualmente similar a la basada en transición entre estados, combina la detección de usos indebidos con un razonamiento para concluir la existencia o no de una intrusión.
- Se definen **escenarios** de ataque: secuencias de comportamiento que forman el ataque.
- En un momento dado, un subconjunto de escenarios (**activos**) representa los ataques que se pueden estar produciendo.
- Un **anticipador** analiza registros y genera la lista de eventos a comprobar para determinar si una intrusión es real.
- Aproximación excesivamente teórica y poco usada en la práctica.

Análisis: Detección de usos indebidos

Problemas:

- La detección de usos indebidos sólo detecta ataques conocidos (y que puedo especificar).
- Es posible camuflar ataques o modificarlos ligeramente.

Ventajas:

- Robustez del esquema: *a priori*, garantiza menos falsos positivos que la detección de anomalías.

IDSes basados en máquina

IDSes basados en máquina

- Introducción
- Verificadores de integridad
- Monitores de registros
- Sistemas de decepción

HIDS: Introducción

Introducción

- Entornos de detección que procesan datos asociados a **un único** sistema.
- Suelen estar ubicados en la propia máquina, aunque no tiene por qué ser así.
- Habitualmente, gestión costosa:
  - Elevado número de sistemas en la organización.
  - Entornos hetereogéneos.
  - Responsabilidad difusa.
  - ...
- Se suelen reservar para máquinas especialmente críticas.

HIDS: SIV

Verificadores de integridad

- IDEA: Queremos ser capaces de detectar cualquier modificación de ciertos ficheros del sistema.
- ¿Cómo? Generalmente mediante funciones resumen (p.e. MD5).
- Problemas:
  - Resumen inicial. ¿En quién confio?
  - Nivel de privilegio.
  - ...
- Ejemplos: Solaris ASET, Tripwire, md5sum...

Funciones resumen

- *Hash Functions.*
- Proyecciones de un conjunto (generalmente de muchos o infinitos elementos) sobre otro de tamaño fijo y más pequeño que el anterior:

  $$H : A \rightarrow B$$

- Ejemplo:

$$H(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases}$$

- No todas las funciones resumen son interesantes en SIV: han de cumplir ciertas propiedades.

HIDS: SIV

Propiedades de las funciones resumen en SIV

- Entrada puede ser de tamaño indeterminado.
- Salida de tamaño fijo, varios órdenes de magnitud más pequeño que la entrada.
- Dado x, calcular $H(x)$ es computacionalmente barato.
- $H(x)$ es de un solo sentido (*One–Way Hash Function*).
  $\nexists\, H^{-1}(x)$ o su cálculo es computacionalmente difícil.
- $H(x)$ no presenta colisiones.
  $x \neq y \rightarrow H(x) \neq H(y)$

Ejemplo: MD5 (*Message Digest 5*).

Verificación de integridad: ejemplos

```
# find /sbin -type f -exec md5sum {} \; >/etc/firmas.orig
# find /sbin -type f -exec md5sum {} \; >/etc/firmas.now
# diff /etc/firmas.orig /etc/firmas.now \
  2>/dev/null >/dev/null \
  || echo "Cambios en /sbin" | mailx -s "PROBLEMAS" root
#
```

```
# find /dev -type f -perm -4000 && \
  echo "Setuidados en /dev" | mailx -s "PROBLEMAS" root
#
```

HIDS: LFM

Monitores de registros

- IDEA: Queremos monitorizar los *logs* para detectar patrones que puedan denotar una intrusión.
- ¿Cómo? Programas específicos: desde productos comerciales hasta simples *shellscripts*.
- Problemas:
  - Conocimiento del entorno.
  - Nivel de privilegio.
  - ...
- Ejemplos: Swatch, LogSentry...

Monitores de registros: ejemplos

```
# tail -100 /var/log/messages | grep -w su
Apr  9 18:03:55 bruja su[2389]: - pts/2 toni-root
Apr  9 18:04:02 bruja su[2390]: + pts/2 toni-root
Apr  9 18:04:50 bruja su[2408]: + pts/4 toni-root
#
```

```
# grep -w phf /var/log/apache/access_log
192.168.0.1 - - [11/Dec/2002:02:50:09 +0100] \
   "GET /cgi-bin/phf" 404 -
192.168.0.3 - - [12/Dec/2002:15:24:38 +0100] \
   "GET /cgi-bin/phf" 404 -
#
```

HIDS: Honeypots

Sistemas de decepción

- *Honeypot*: Recurso de un sistema informático cuyo valor reside en su uso ilegítimo o no autorizado.
- Simulación de un entorno real con el fin de atraer la atención de atacantes y recabar información sobre los mismos.
- IDEA: Queremos proporcionar **conocimiento negativo** a un atacante para poder estudiar sus intenciones, sus métodos, para ganar tiempo y pruebas tras una denuncia...
- ¿Cómo? Simulando vulnerabilidades.
- Ejemplos: PortSentry, FakeBO...

HIDS: Honeypots

¿Por qué decepcionar?

- Conocimiento de la plataforma tecnológica real y del riesgo al que se expone.
- Estudio de medidas defensivas a aplicar en el entorno real.
- Mejora de la capacidad de reacción: más tiempo para responder y mayor conocimiento del ataque.
- Monitorización del atacante en un entorno muy controlado.
- Reducción (¿ausencia?) de falsos positivos.
- Detección de ataques automatizados (p.e. *spammers*, gusanos...).

HIDS: Honeypots

Los *honeypots* se clasifican en función de su nivel de interacción con el atacante:

- Baja interacción: el sistema **emula** vulnerabilidades.
  - Facilidad de implantación.
  - Riesgo mínimo.
  - Interacción limitada: captura de poca información.
  - Ejemplo: *honeytokens*.
- Alta interacción: no existe emulación, se implantan sistemas y servicios reales.
  - Captura de grandes cantidades de información.
  - Complejidad de implantación y explotación.
  - Riesgo considerable: ofrecemos a los atacantes sistemas completos con los que interactuar.
  - Ejemplo: *honeynets*.

HIDS: Honeypots

HoneyTokens

- Caso particular de *honeypot*: recurso con el que un atacante interactua (al que accede de alguna forma).
- Idea original: *The Cuckoo´s Egg* (Cliff Stoll).
- Implementación sencilla y eficacia en la detección.
- No tiene porqué ser un servicio o un sistema: es válido cualquier tipo de 'entidad' digital.
- Ejemplos:
  - Nombre de usuario (*login*) falso.
  - Número de tarjeta de crédito.
  - Registro de base de datos (JFK).
  - Fichero o directorio.
  - ...

Sistemas de decepción: ejemplos

```
# grep uucp /etc/services
uucp    540/tcp    uucpd    # uucp daemon
# echo "Sonrie a la camara oculta" | nc -l -p 540 &
[3] 2597
#
```

```
# cat /tmp/get_root
#!/bin/sh
(whoami && date ) >>/tmp/.registro
while :; do
    /bin/echo -n "# "
    read orden && echo $orden >>/tmp/.registro
done
# touch /tmp/.registro && chmod 222 /tmp/.registro
# chmod +x /tmp/get_root
#
```

Problemas y peligros de la decepción

- Vulnerabilidades asociadas al sistema de decepción.
- Aspectos legales:
  - ¿Pruebas judiciales válidas?
  - ¿Uso ilícito de sistemas por parte de la propia organización?
  - ...
- Rendimiento vs. coste.
- Posible pérdida de imagen de la organización.
- ...

IDSes basados en red

IDSes basados en red

- Introducción
- IDS en el cortafuegos
- IDS en la red
- *Honeynets*

Introducción

- Entornos de detección que procesan datos asociados a diferentes sistemas de una red.
- Generalmente, se ejecutan en uno solo de los *hosts* de la red que monitorizan.
- Facilidad de detección en una red:
  - Mucho *software* gratuito.
  - Pocos recursos *hardware* (y baratos).
  - Escasas modificaciones de la arquitectura.
  - ...

NIDS: Introducción

Las tramas que circulan por una red pueden ser portadoras de un tráfico malicioso en cada uno de sus campos:

- Fragmentación (DF, MF).
- Dirección origen y destino.
- Puerto origen y destino.
- *Flags* TCP.
- Campo de datos.
- ...

Debemos **analizar** las tramas en busca de patrones que puedan denotar una intrusión (¿y **detenerlas**?).

NIDS: Introducción

¿Dónde analizar estas tramas?

- En el cortafuegos corporativo.
  $\Rightarrow$ O en los cortafuegos.
- En la propia red.

En cada caso existen pros y contras. Vamos a verlos...

NIDS: Cortafuegos

¿Por qué detectar en nuestro *firewall*?

- Elemento crítico en cualquier plataforma.
- Nexo de unión con el mundo exterior.
- Punto por el que pasa todo el tráfico (el 'bueno' y el 'malo').
- Mecanismos simples (no implican modificar la arquitectura, ni probablemente incorporar nuevo *software*)...
- ...y efectivos (baja probabilidad de falsos positivos).

NIDS: Cortafuegos

¿Qué información maneja mejor un cortafuegos?

La incluida en las diferentes cabeceras de una trama:

- Direcciones origen y destino.
- Puertos origen y destino (servicios).
- Protocolos.
- *Flags*.
- ...

NIDS: Cortafuegos

¿Qué ataques detecto mejor en un cortafuegos?

Los que 'vienen' en dichas cabeceras:

- Todo tipo de barridos de puertos.
- *IP–Spoofing*.
- Ciertos troyanos.
- Tráfico anómalo.
- ...

NIDS: Cortafuegos

Ejemplos de eventos significativos

- Acceso a puertos 'anómalos':

```
block in log quick on elxl0 from any to any port = 79
block in log quick on elxl0 from any to any port = 31337
```

- Orígenes falseados (p.e. dirección interna que proviene de red externa):

```
block in log quick on elxl0 from 127.0.0.0/8 to any
block in log quick on elxl0 from 192.168.0.0/16 to any
```

- Violaciones de protocolo:

```
block in log quick proto tcp from any to any flags SF/SF
block in log quick proto tcp from any to any flags SR/SR
```

- ...

NIDS: Red

IDS en la red

- Es difícil detectar cierto tipo de ataques en el cortafuegos...
- ...pero interesante hacerlo antes de que lleguen a la máquina.
- Cuestiones de diseño:
  - Número de sensores.
  - Ubicación de los sensores: ¿qué hacer en una arquitectura basada en *switches*?

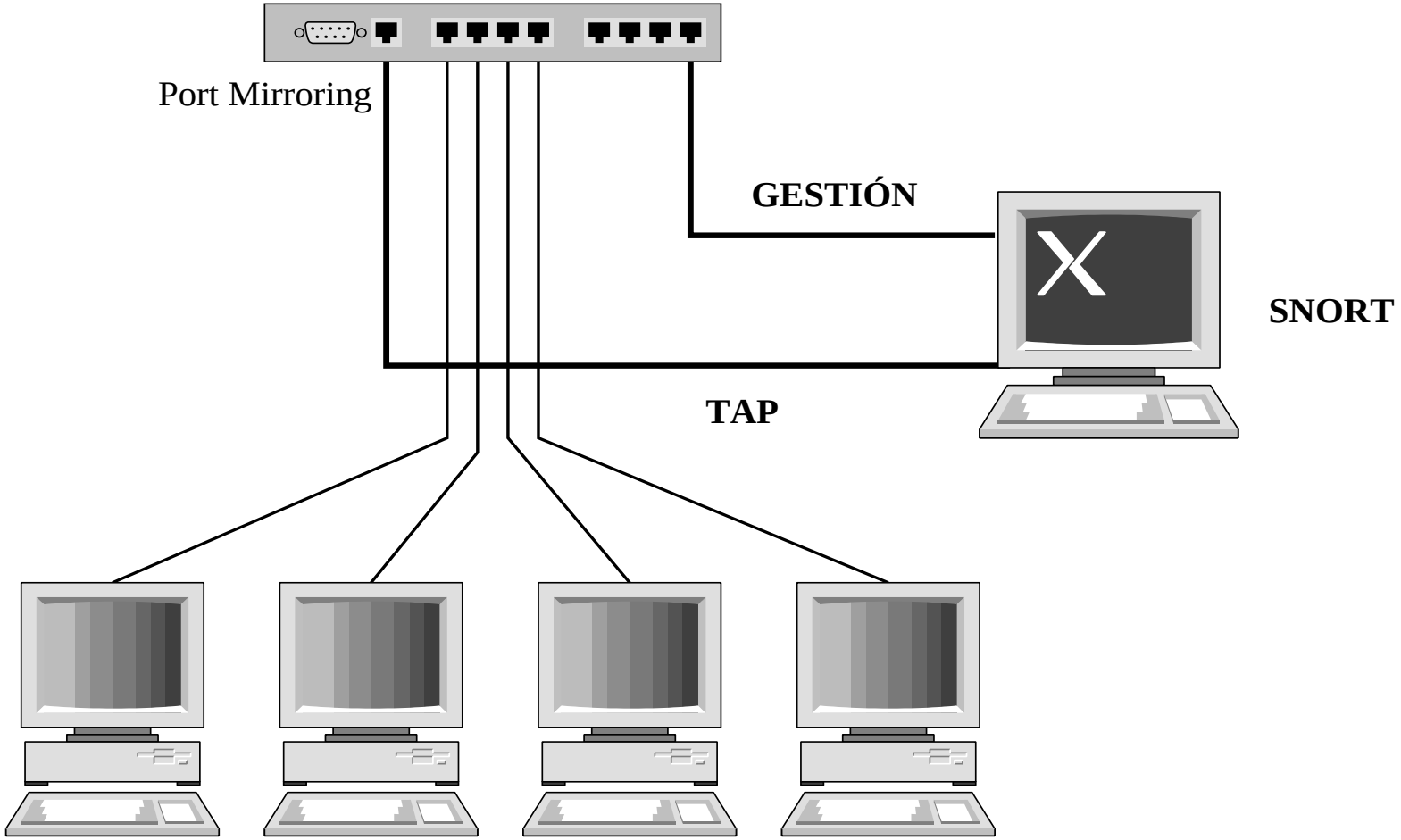
NIDS: Red

¿Cómo hacerlo?

- *Hardware.*
  - PC de gama media + 2 tarjetas de red (TAP + Gestión).
  - *Switch* con soporte para *port mirroring.*
- *Software.*
  - Sistema operativo.
  - *Software* de detección:
    - ISS RealSecure.
    - Dragon IDS.
    - ...
    - **SNORT.** $\Leftarrow$

NIDS: Red

Gráficamente...



NIDS: Red

Un ejemplo: SNORT

- Sistema de detección de intrusos basado en red.
- Detección de usos indebidos (*pattern matching*).
- Uso de **preprocesadores**: reensamblado TCP, normalización de tráfico HTTP, detección de escaneos de puertos...
- Sistema multiplataforma (Unix principalmente, también Windows).
- Más información: `http://www.snort.org/`

SNORT: Modos de funcionamiento

- *Sniffer*: Captura tráfico y lo imprime en pantalla.
- Registro de paquetes: Guarda los paquetes en ficheros con un determinado formato (posibilita la reproducción posterior).
- NIDS: Compara las tramas con un conjunto de patrones predeterminado, registrando en archivos las coincidencias.

SNORT: Reglas

- SNORT proporciona un **elevado** número de reglas genéricas (patrones de ataques).

  $\Rightarrow$ ¿Cuáles utilizar?

- A las pocas horas de descubrirse un nuevo ataque o método de explotación se distribuyen los patrones asociados.
- Podemos automatizar la actualización, aunque no es recomendable.

  $\Rightarrow$ Una nueva regla no tiene porqué ser una buena regla.

- Problemas: Los habituales en detección de usos indebidos:
  - Personalización de la base de datos de patrones (especificación de lo 'anormal').
  - Camuflaje del ataque.
  - ...

SNORT: Reglas (ejemplos)

- Dos partes diferenciadas:
  - Cabecera (define **quién** activa la regla):

    ```
    alert tcp $EXTERNAL_NET any -> $HOME_NET 80
    ```

  - Opciones (definen **qué** activa la regla):

    ```
    (msg:"IDS128 - CVE-1999-0067 - CGI phf attempt";\
    flags:PA; content:"/phf";flags:AP; nocase;)
    ```

- Ejemplo anterior:
  - La regla genera una alerta cuando llega tráfico TCP de una red externa al puerto 80 de la red interna...
  - ...y ese tráfico contiene la cadena `'/phf'` y los bits ACK o PSH activos, generando el mensaje correspondiente.

NIDS: Red

Ventajas de SNORT

- Sistema probado y fiable.
- Correcto soporte y actualización.
- Funciona sobre diferentes operativos y arquitecturas.
- Productos de terceros (AR, BBDD...).
- Barato (¡Libre!)
- ...

Honeynets

- Idea similar al *honeypot*: una red señuelo es la simulación completa de una red en producción, con el fin de atraer la atención de los atacantes y recopilar información sobre los mismos.
- **Todo** el tráfico que entra en la red es sospechoso: trataremos de controlarlo, monitorizarlo, capturarlo y analizarlo.
- Aumenta las ventajas de un sistema de decepción, pero también los problemas.
- Costes de implantación y gestión considerables.

Honeynets: aspectos clave

- **Control de datos**

  $\Rightarrow$ Garantía de que la *honeynet* no va a ser utilizada como plataforma de ataque.

- Captura de datos.

  $\Rightarrow$ Captura del tráfico que entra y sale de la *honeynet*, de forma transparente al atacante.

- Recolección de datos (*honeynets* distribuidas).

  $\Rightarrow$ Envío seguro de la información recopilada a un punto de análisis centralizado.

Adicionalmente...

- Análisis de datos.

  $\Rightarrow$ ¿Qué hacer con la información obtenida? ¿Cómo hacerlo?

Honeynets: control de datos (requisitos)

- Tanto manual como automático.
- Al menos dos niveles de protección ante fallos.
- Fallos en el control provocan la negación de acceso y salida al o desde la red señuelo
- Mecanismos de control altamente configurables en cualquier momento.
- Debe ser difícil de detectar por el atacante.
- Alertas automáticas ante una intrusión.

Honeynets: captura de datos (requisitos)

- Los datos capturados nunca se almacenan dentro de los propios sistemas de decepción.
- Se debe mantener almacenada al menos un año la siguiente información:
  - Conexiones de entrada y salida a la red (cortafuegos).
  - Capturas compleas del tráfico de red.
  - Actividad de cada sistema.
- Posibilidad de monitorización en tiempo real.
- Garantías de seguridad de los sistemas de captura.
- Otros: sincronización de relojes UTC, registros según estándar...

Honeynets: recolección de datos

- Identificador único para cada red señuelo, con nombre y tipo asociados.
- Seguridad de las comunicaciones (cifrado).
- Opciones de anonimización de los datos confidenciales de la organización.
- Sincronización de relojes (NTP).

Honeynets: análisis de datos

- Correlación de los datos recibidos desde la red señuelo.
- Nivel de trazabilidad aceptable.
- Aplicación de técnicas de análisis forense.
- ...

Honeynets: consideraciones generales

- Elemento que redirige al atacante a una red controlada.
  $\Rightarrow$ Transparencia.
- Marcado de tramas... ¿Cuáles redirigir y cuáles no?
- Securización de los sistemas señuelo.
- Trazabilidad de las actividades.
- Filtrado del tráfico saliente.
- Rendimiento: ¿afecto al del sistema real?

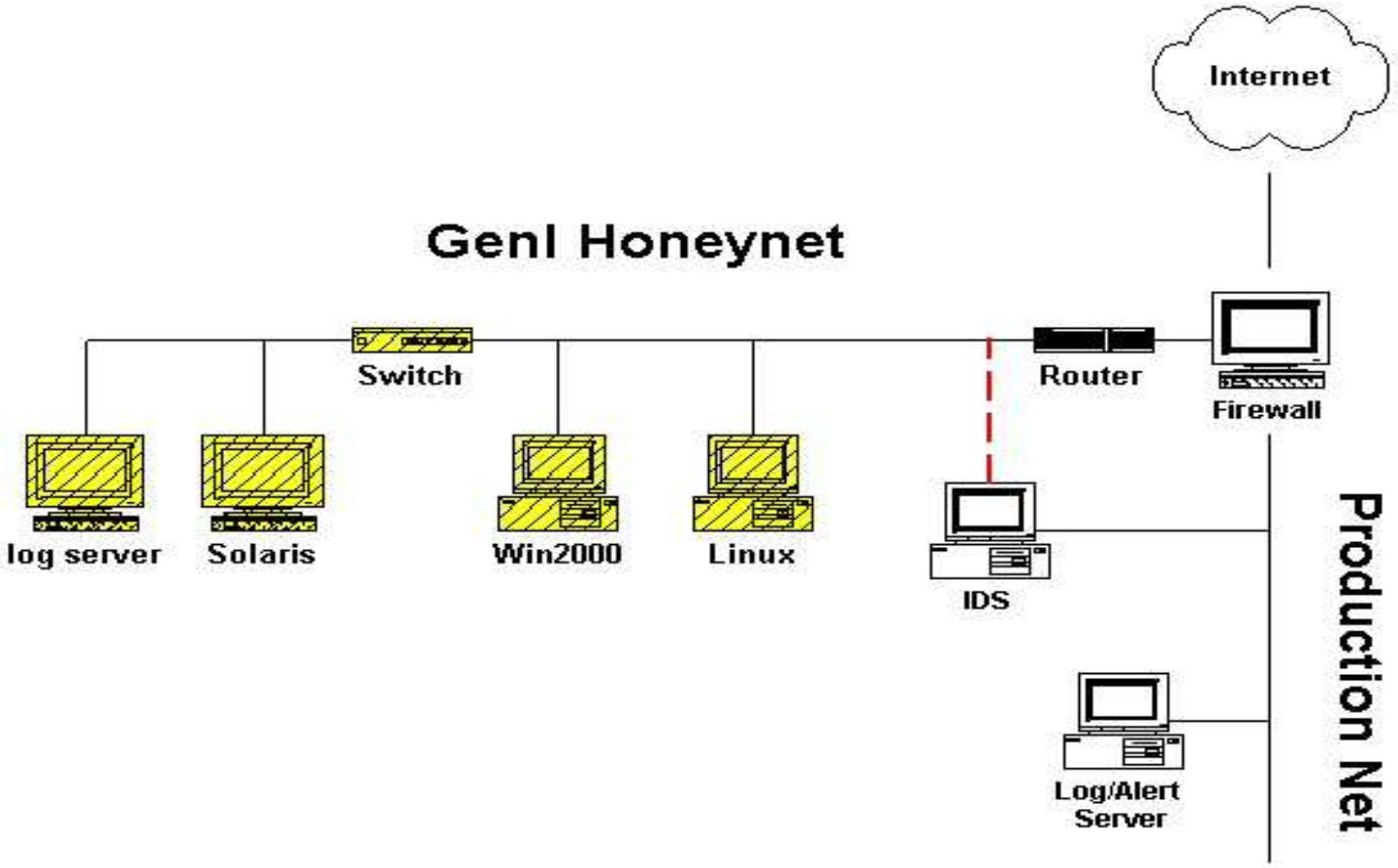
Honeynets: GenI

- Cortafuegos de nivel 3 para control de datos.

  $\Rightarrow$ Conexiones salientes limitadas (límite depende de la funcionalidad requerida).

- *Router* (nivel 2) para control adicional: bloqueo ICMP, filtros *antispoofing*, ...
- NIDS para captura de datos de red (y alerta ante problemas).
- `syslog` para captura de datos de máquina (y alerta ante problemas).

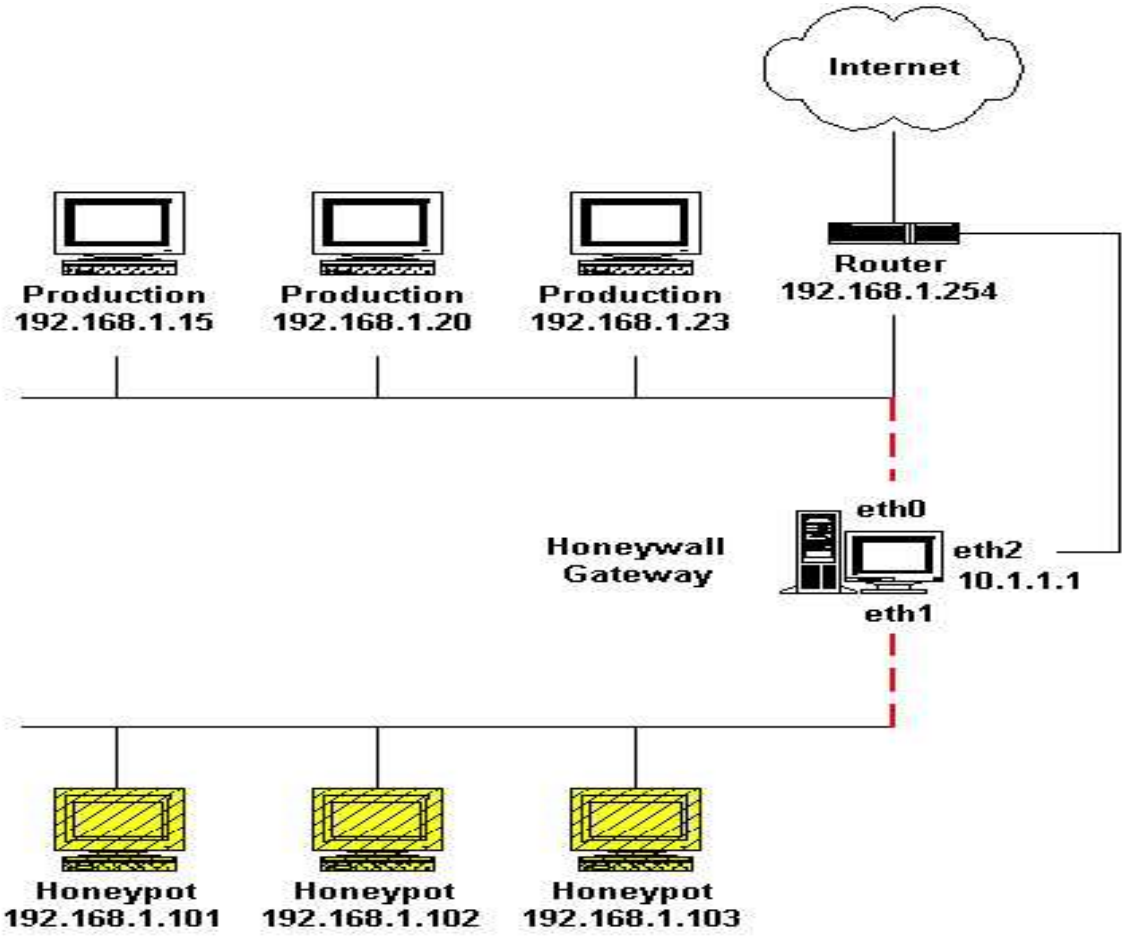
Honeynets: GenI (gráficamente)



Honeynets: GenII

- Más fáciles de desarrollar y más difíciles de detectar.
- Mayor interacción del atacante: más información recopilada.
- Elemento clave: *honeywall* (control y captura). Pasarela de nivel 2, con interfaz de nivel 3 para administración.
- Control de datos: aumento de la inteligencia, detectando la actividad no autorizada y respondiendo ante la misma (IPS).
- Captura de datos: principalmente en el núcleo del operativo.
  $\Rightarrow$ Detección de actividad cifrada.

Honeynets: GenII (gráficamente)



Honeynets: peligros

- Mismos peligros que sistemas de decepción: legales, coste, imagen...
- Además (o *especialmente*):
  - Daño: uso de la red señuelo para atacar redes en producción.
  - Detección: una vez el atacante es consciente de la trampa, ésta pierde su valor.
  - Deshabilitación: supresión o falseamiento de los datos capturados en la red señuelo.
  - Violación: actividades ilegales desde la red señuelo sin atacar a nadie: distribución de música, *software* protegido, pornografía infantil...

NIDS: Honeynets

Honeynets: *The HoneyNet Project*

- Organización sin ánimo de lucro que desde 1999 trabaja por la seguridad en Internet a través de la investigación y el desarrollo.
- Red de *honeynets* distribuidas por todo el mundo.
- *Know your enemy.*
- *Challenge of the month.*
- `http://www.honeynet.org/`

Gran parte de la información sobre redes señuelo presentada en este curso ha sido extraída del proyecto *HoneyNet*.

IDSes distribuidos

- Introducción
- Arquitectura
- Consideraciones:
  - Comunicación.
  - Capacidad.
  - Inteligencia.
- Algunos ejemplos

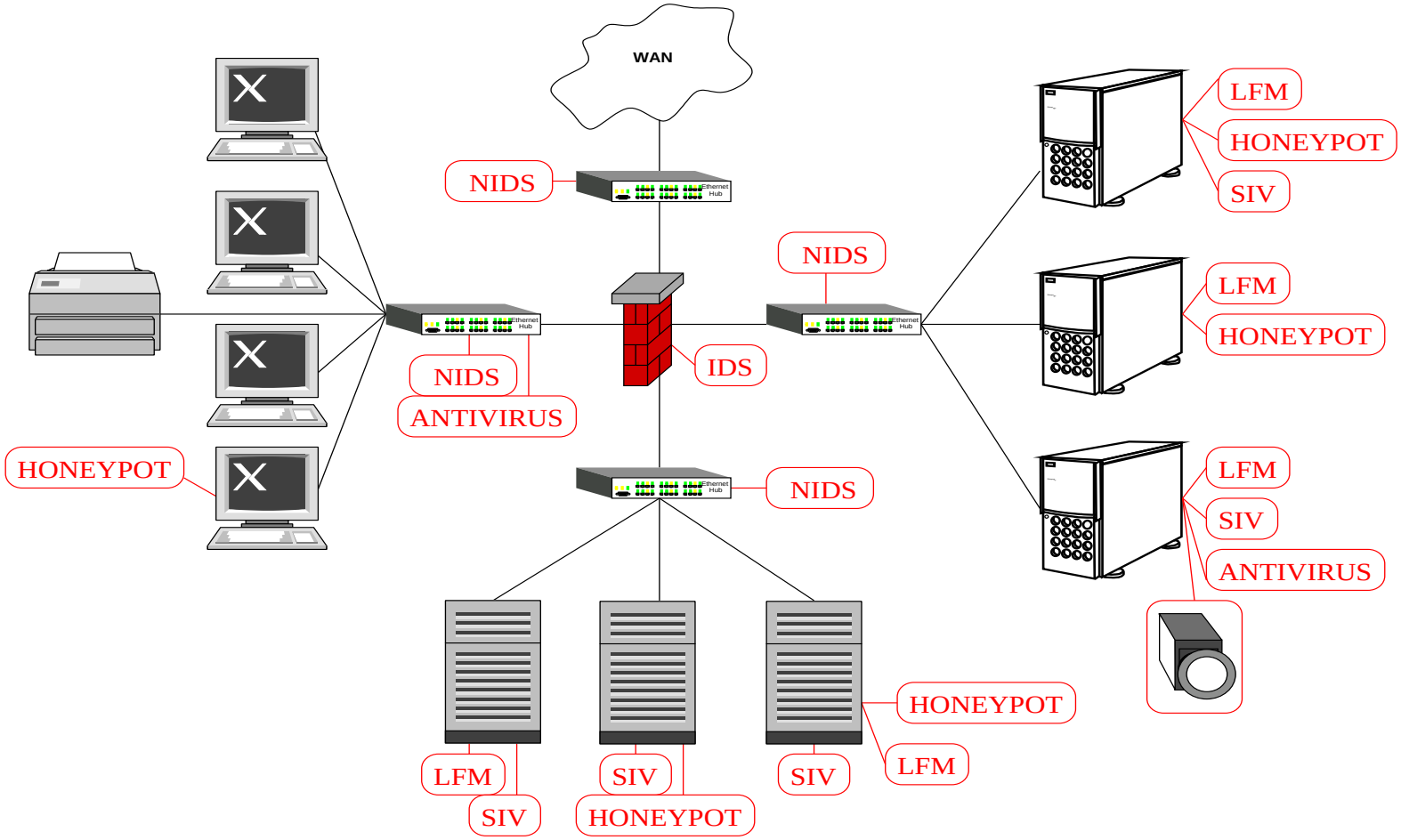
IDSes distribuidos: Introducción

Definiciones

- **Sistema distribuido:** Conjunto de elementos autónomos enlazados entre sí mediante una red, que aparecen frente al usuario como un único sistema.
- **Evento:** Ocurrencia de un conjunto determinado de circunstancias.
- **Sistema distribuido de detección de intrusos (DIDS):** Sistema de detección capaz de agregar eventos generados por diferentes fuentes, proporcionando así una imagen más amplia y detallada de las actividades maliciosas en un determinado entorno.
  - *. . . diferentes fuentes. . .*
  - *. . . capaz de agregar eventos. . .*
  - *. . . imagen más amplia y detallada. . .*

IDSes distribuidos: Introducción

El enfoque clásico...



IDSes distribuidos: Introducción

Problemas del enfoque

- Visión 'local': dificultad para detectar ciertos ataques.
- Gestión compleja ⇒ DESUSO.
- Escasa tolerancia a fallos.
- Escalabilidad nula.
- Costes de mantenimiento y operación elevados.
- **Muchos datos, poca información.**
- ...

IDSes distribuidos: Introducción

Algunas ideas...

- Entorno con múltiples IDSes potencialmente diferentes entre sí.
- Elementos que reciben información (eventos) de estos subsistemas...
- ...y aportan inteligencia adicional.
- Comunicación entre componentes del esquema.
- Humano que ve 'resumen' del estado en cada momento...y toma decisiones en función del mismo.
- Generalización del entorno: visión global de la seguridad.

IDSes distribuidos: Arquitectura

Componentes habituales de un DIDS

- 3+1 elementos básicos en un DIDS:
  - Agentes (monitorizan actividad).
  - Transceptores (comunicación).
  - Maestro/s (centralizan datos).
  - Consola de eventos (interfaz con operador).
- En entornos extremadamente sencillos $\not\exists$ transceptores: comunicación unidireccional ('transmisores').
- Pueden definirse más componentes: generadores, *proxies*, actores, *clusters*...
- Nomenclatura confusa.

IDSes distribuidos: Arquitectura

Agente

- Entidad independiente que monitoriza algún tipo de actividad 'interesante'.
- Ejecutado de forma contínua o bajo demanda.
- No se comunica con otros agentes.
- Diferentes propósitos y lenguajes.
- De 0 a N en cada máquina.
- Nivel más bajo de la jerarquía.

IDSes distribuidos: Arquitectura

Transceiver

- 'Transceptor': transmisor + receptor.
- Un transceptor por cada máquina donde existe algún agente.
- Interfaz de comunicación entre agentes y maestros.
- Transmite información a un maestro (o a varios).
- Recibe órdenes de los maestros.

IDSes distribuidos: Arquitectura

Maestro

- Cerebro del esquema: alta importancia y complejidad.
- Recibe eventos de los transceptores (o de los agentes, en entornos simples).
- Detecta ataques incluso no notificados por transceptores.
- Control de agentes (y transceptores).
- Centralización de la información.
- Nivel máximo de inteligencia 'automática'.

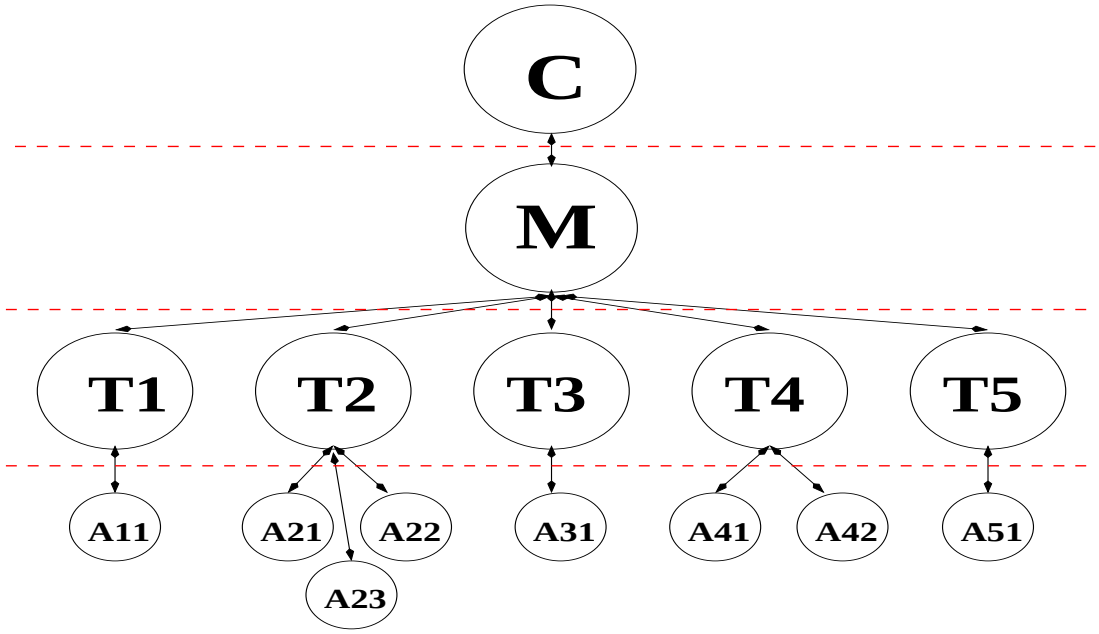
IDSes distribuidos: Arquitectura

Consola de eventos

- Elemento superior de la jerarquía.
- Interfaz del DIDS con un humano.
- Ayuda en la toma de decisiones.
- Quizás no es parte técnica del esquema pero...
- ...¿se podría eliminar del mismo?

IDSes distribuidos: Arquitectura

Un ejemplo gráfico...



- Un maestro (podríamos ampliar el esquema).
- Cinco subsistemas (cinco transceptores).
- N agentes en cada subsistema.
- Estructura **jerárquica** (lo habitual, aunque existen propuestas basadas en modelos no jerárquicos).

IDSes distribuidos: Consideraciones

PROBLEMA: Todos los elementos del esquema deben 'hablar' un lenguaje común.

- Definición de estándares para el tratamiento de los datos de todos los elementos:
  - Vocabulario común.
  - Formato de la información.
  - Intercambio de datos entre elementos.
  - ...
- Dos ejemplos de formalización: CIDF e IDEF.
- SUBPROBLEMA: Fabricantes de sistemas de detección.

IDSes distribuidos: Consideraciones

PROBLEMA: Gran cantidad de información a procesar.

- Incremento de los requerimientos *hardware* (tanto a nivel de máquina como de red) y *software*.
- Optimización de algoritmos de procesamiento (maestro/s).
- Replicación de la jerarquía: elementos intermedios que aportan inteligencia (y reducen los datos intercambiados).
- Aumento de la inteligencia de los agentes: extracción y transmisión de los datos más significativos.

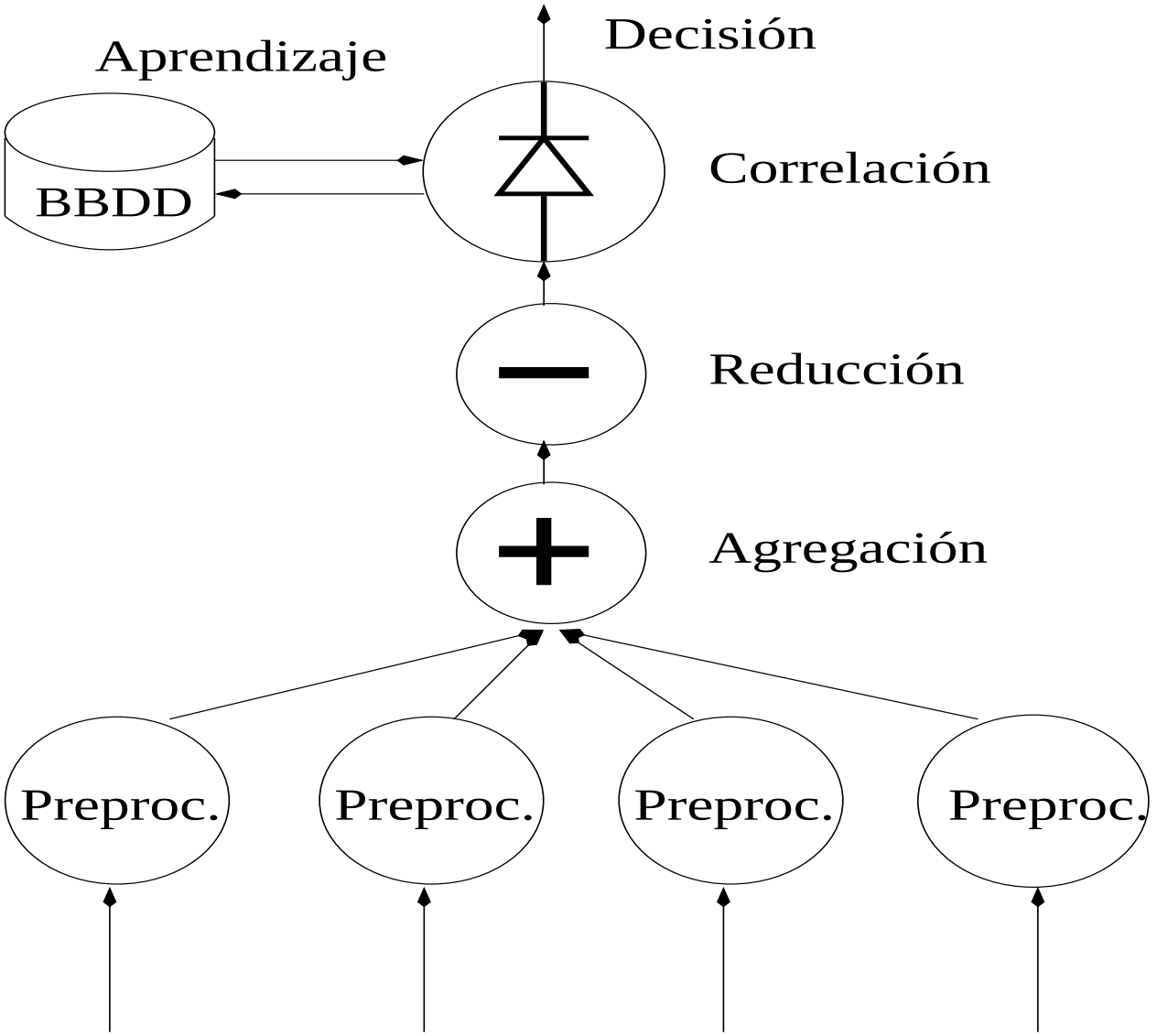
IDSes distribuidos: Consideraciones

PROBLEMA: ¿Cómo tratar los datos para generar 'inteligencia'?

- Preproceso: normalización y control de errores de los datos recibidos de los agentes.
- Agregación: agrupación y ordenación de alertas.
  $\Rightarrow$ Sincronización de relojes en cada elemento.
- Reducción: detección de alertas duplicadas.
  $\Rightarrow$ Proyecto: CIEL (*Common Intrusion Event List*).
- Correlación: detección de las relaciones entre eventos.
- Aprendizaje: realimentación del maestro para determinar nuevas relaciones y estados.
- Decisión: alerta a un operador, respuesta automática...

IDSes distribuidos: Consideraciones

Gráficamente...



IDSes distribuidos: Ejemplos

Algunos productos del mercado:

- IBM Tivoli Risk Manager
- NetForensics Security Information Management
- ArcSight Enterprise Security Management
- Tenable Network Security Lightning Console
- Prelude (*Open Source*)

Problemas habituales:

- Gestión compleja.
- Integración con productos de terceros.
- Correlación.
- €, \$, £, ¥...

Respuesta automática

Respuesta automática

- Introducción
- Esquemas de respuesta
- Consideraciones

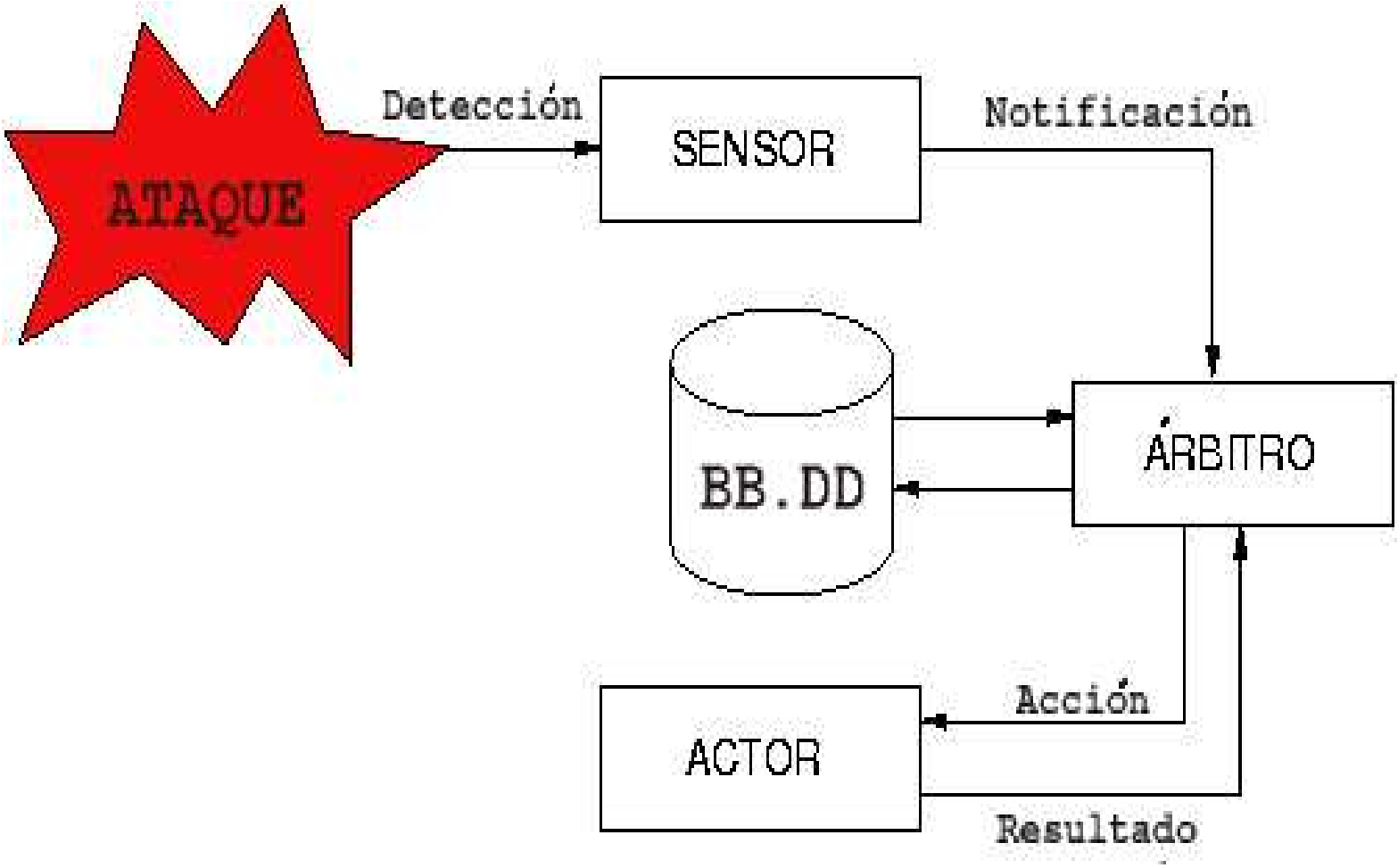
Respuesta automática: Introducción

Introducción

- Un IDS es un mecanismo de seguridad **pasivo**.
- Nos interesa responder ante un ataque, transformando el esquema de detección en un elemento **activo**.
- Dos tipos de respuesta: manual y automática.
- **Respuesta automática (AR):** Conjunto de acciones que se ejecutan sin intervención humana al detectar un evento, generalmente con el objetivo de salvaguardar la integridad, disponibilidad o confidencialidad de un determinado recurso.

Respuesta automática: Introducción

Esquema de funcionamiento



Respuesta automática: Introducción

Respuesta automática vs. manual

- Rapidez.
- Escalabilidad.
- Registro.
- Integración.
- Precio.
- ¿Fiabilidad?

Respuesta automática: Introducción

Ejecución de la AR

- Una intrusión – una respuesta.
- Una intrusión – varias respuestas.
- Varias intrusiones – una respuesta.
- Varias intrusiones – varias respuestas.

Respuesta automática: Esquemas

Esquemas de respuesta automática

- Registro
- Bloqueo
- Ataque
- Recuperación
- Decepción

Respuesta automática: Esquemas

Registro

- Activación de registros adicionales.
- Todavía es un mecanismo pasivo.
- Útil para monitorización.
- Ejemplo: activación del sistema de auditoría de Unix ante un acceso sospechoso a `/etc/passwd`

Respuesta automática: Esquemas

Bloqueo

- Suprime interacciones entre atacante y atacado.
- El esquema más habitual.
- Mecanismos **activos**.
- Ejemplos: bloqueo en cortafuegos intermedio, suspensión de trabajos en un *host*...

Respuesta automática: Esquemas

Ataque

- Acciones agresivas contra el pirata.
- Mecanismos **activos**.
- ¿Es ética esta respuesta?
- Ejemplo: lanzamiento de negación de servicio contra el pirata ante un intento de intrusión.

Respuesta automática: Esquemas

Recuperación

- Detectan cambio en el estado de un recurso atacado y lo devuelven a su estado anterior.
- Mecanismos **activos**.
- Ejemplo: sistema que restaura el contenido de una página *web* si detecta una modificación.
- En muchas ocasiones, difíciles de implantar (p.e. actualización frecuente de páginas *web*).

Respuesta automática: Esquemas

Decepción

- 'Decepcionan' al atacante.
- Mecanismos **activos**.
- Partidarios vs. detractores: ¿Sirve de algo la decepción?
- Ejemplos: '*Sonría a la cámara oculta*', PHF...

Respuesta automática: Consideraciones

Riesgos de la AR

- Subversión del sistema.
- Respuestas inadecuadas.
- ...

$\Downarrow$

Negación de servicio

Respuesta automática: Consideraciones

Minimización del riesgo

- Es **vital** minimizar los riesgos que asumimos al ejecutar la AR.
- Diferentes aproximaciones para lograrlo.
- Tres de ellas:
  - Limitación de respuestas por u.t.
  - Actores protegidos.
  - Ponderación de ataques.

Respuesta automática: Consideraciones

Limitación de respuestas por u.t.

- IDEA: Determinar un número máximo de respuestas por unidad de tiempo, superado el cual el sistema no actúa.
- OBJETIVO: Evitar negaciones de servicio masivas.

¡$\nexists$ umbral universal!

Respuesta automática: Consideraciones

Actores 'protegidos'

- IDEA: Establecer elementos contra los que **nunca** se actúa.
- OBJETIVO: No dañar recursos controlados o vitales para el correcto funcionamiento de una plataforma.

Respuesta automática: Consideraciones

Ponderación de ataques

- IDEA: No todos los eventos son igual de 'sospechosos'.
- OBJETIVO: Ponderar cada actividad sospechosa en función de su probabilidad de ser un ataque real.
- PROBLEMA: ¿Quién determina la probabilidad de que una actividad 'sospechosa' sea realmente un ataque?

Respuesta automática: Consideraciones

Algoritmo de activación

ESTADO: {Detección ataque}

SI umbral de respuestas superado: REGISTRAR && FIN

SI NO

    Comprobación actor atacante

    SI es actor protegido: REGISTRAR && FIN

    SI NO

        Ponderación histórica de gravedad

        SI umbral de gravedad superado: **ACTUAR**

        SI NO

            REGISTRAR

        FSI

    FSI

FSI

Estándares de detección de intrusos

- Introducción
- CIDF
- IDEF

Estándares: Introducción

Introducción

- Cada día es más habitual implantar sistemas de detección de intrusos de diferentes fabricantes y basados en tecnologías diversas.
- Las víctimas de un ataque son múltiples, y cada una de ellas utilizará sus propios IDSes: es de gran ayuda un formato común.
- Los ataques distribuidos son cada día más habituales, y por tanto los sistemas de detección de intrusos distribuidos más necesarios.
- Es interesante que los sistemas de detección puedan compartir datos ('hablar el mismo lenguaje'): necesidad de estándares.
- Los estándares permiten no sólo compartir datos, sino también que estos sean analizados de forma similar, independientemente de su fuente.

Estándares: CIDF

CIDF (*Common Intrusion Detection Framework*)

- Promovido por DARPA (*Defense Advanced Research Projects Agency*).
- Orientado a proyectos de investigación en detección de intrusos.
- Escasa aceptación comercial: proyecto finalizado en 1999.
- Más información: `http://gost.isi.edu/cidf/`

Estándares: CIDF

CIDF define...

- Arquitectura genérica para la detección de intrusos.
- Protocolos de comunicación entre elementos.
- Lenguaje para la representación de datos: CISL (*Common Intrusion Specification Language*).
- Interfaces de programación.

CIDF: Arquitectura

Se definen cuatro tipos básicos de equipos:

- **Equipos E.** Sensores: detectan eventos 'interesantes' y emiten alertas (informes).
- **Equipos A:** Analizadores: reciben los informes de los equipos E y los analizan, emitiendo una 'recomendación'.
- **Equipos D:** Componentes de bases de datos: correlación, estadísticas, análisis históricos, etc.
- **Equipos R:** Equipos de respuesta: responden al evento a partir de los datos generados por los equipos anteriores.

CIDF: Comunicación

- Los diferentes componentes definidos anteriormente intercambian información en forma de **gidos** (*Generalized Intrusion Detection Objects*).
- Un *gido* puede representar:
  - La ocurrencia de un cierto evento en un momento determinado.
  - Una conclusión extraída de un conjunto de eventos.
  - Instrucciones para ejecutar una acción (respuesta).
- Los *gidos* son representados en un lenguaje común denominado CISL.

Estándares: CIDF

CISL: *Common Intrusion Specification Language*

- Lenguaje para expresar información acerca de eventos, ataques y respuestas en el entorno de detección.
- Sintaxis basada en expresiones S: agrupaciones recursivas de etiquetas y datos.
- Etiquetas: SIDs (*Semantic IDentifiers*). Indican la interpretación del resto de la expresión, convirtiéndose en el corazón del lenguaje.
- Diferentes tipos de SIDs. Los principales:
  - Verbos. Indican una acción o recomendación ('borrar', 'bloquear', 'ejecutar',...).
  - Roles. Informan de los elementos implicados en el verbo (objetos o actores).
  - ...

Estándares: CIDF

CISL: ejemplos

```
(Account
    (UserName 'toni')
    (HostName 'andercheran.aiind.upv.es')
)
```

```
(Execute
    (Initiator
        (UserName 'toni')
    )
    (Process
        (FileName 'passwd')
    )
)
```

CIDF: API

- La forma lógica del *gido* (expresión S) no es la recomendada para comunicar elementos: es necesario codificarlos.
- Codificación: secuencia no ambigua de *bytes* que representa la información contenida en el *gido*.
- CIDF define un API para construir un *gido*, codificarlo para ser transmitido a otro componente de la arquitectura y decodificarlo en este componente para ser procesado.
- Se proporcionan librerías y código (C) para desarrolladores.

Estándares: CIDF

CIDF: *The End*

- El proyecto CIDF finalizó en 1999 debido principalmente a su escasa aceptación comercial.
- Sus ideas y objetivos en la comunicación y el lenguaje utilizados siguen vigentes:
  - Tolerancia a fallos.
  - Escalabilidad.
  - Expresividad y precisión.
  - Simplicidad.
  - Facilidad de uso.
  - ...
- Trabajo reaprovechado en la actualidad: IDEF.

Estándares: IDEF

IDEF (*Intrusion Detection Exchange Format*)

- Definido por el *Intrusion Detection Working Group*, de IETF.
- Orientación comercial: IDWG está formado por empresas relacionadas con la detección de intrusos, en desacuerdo con parte del trabajo de CIDF.
- Define:
  - Protocolos de comunicación: *Intrusion Detection Exchange Protocol* (IDXP).
  - Modelo de datos: *Intrusion Detection Message Exchange Format* (IDMEF).
- Más información:
  `http://www.ietf.org/html.charters/idwg-charter.html`

Estándares: IDEF

IDXP: *Intrusion Detection eXchange Protocol*

- Basado en protocolo BEEP (*Blocks Extensible Exchange Protocol*, RFC 3080), que establece sesiones para permitir el intercambio de mensajes entre emisor y receptor, codificados en XML.
- Mecanismo:
  - Establecimiento de sesión BEEP.
  - Negociación de un perfil de seguridad aceptable.
  - Transmisión de datos IDXP:
    * *IDXP Greeting.*
    * Opciones.
    * Mensaje IDMEF.

IDXP: Garantías

IDXP (+BEEP) garantiza los requisitos exigidos para el transporte de datos IDMEF:

- Transmisión fiable de mensajes.
- Interacción con cortafuegos.
- Autenticación.
- Confidencialidad.
- Integridad.
- Autenticación unívoca de emisores.
- Resistencia ante ataques DoS.
- Resistencia ante mensajes duplicados.

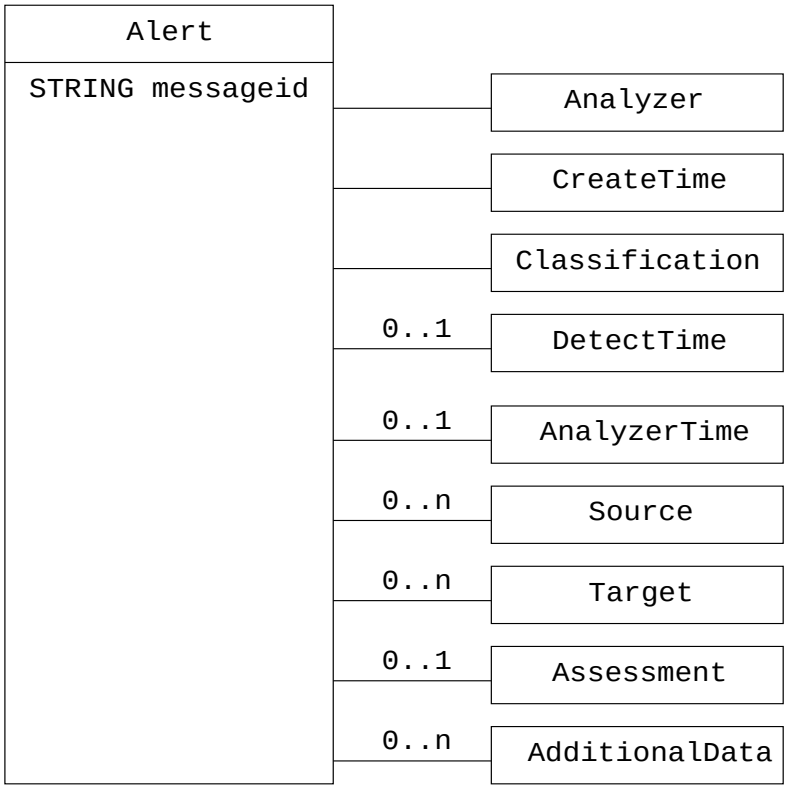
Estándares: IDEF

IDMEF: *Intrusion Detection Message Exchange Format*

- IDMEF define un formato de datos estándar para las alertas con que los diferentes elementos de un esquema de detección de intrusos trabajan.
- Modelo en XML: definición de un DTD (*Document Type Definition*) que establece la sintaxis adecuada para representar la información relativa a intrusiones.

IDMEF: Alertas

- La clase **Alert** es el corazón de IDMEF.
- Cada alerta tiene un identificador (STRING) y una serie de clases agregadas:



IDMEF: Alertas. Clases agregadas

- *Analyzer:* Identificador del analizador que origina la alerta.
- *CreateTime:* Momento en que la alerta es creada.
- *DetectTime:* Momento en que se detecta el evento que origina la alerta.
- *AnalyzerTime:* Fecha y hora del analizador cuando origina la alerta.
- *Classification:* Nombre de la alerta.
- *Assessment:* Información relativa al impacto del evento, acciones de respuesta emprendidas y confianza de la detección.
- *AdditionalData:* Información adicional de la alerta.
- *Source:* Fuente del evento que origina la alerta.
- *Target:* Objetivo del evento que origina la alerta.

IDMEF: Ejemplo (I)

```
<idmef:IDMEF-Message version="1.0"
                     xmlns:idmef="http://iana.org/idmef">
  <idmef:Alert messageid="abc123456789">
    <idmef:Analyzer analyzerid="bc-sensor01">
    </idmef:Analyzer>
    <idmef:CreateTime ntpstamp="0xbc71e980.0x00000000">
      2000-03-09T08:12:32-01:00
    </idmef:CreateTime>
```

IDMEF: Ejemplo (II). Atacante.

```
<idmef:Source ident="abc123">
  <idmef:Node ident="abc123-001">
    <idmef:Address ident="abc123-002"
                   category="ipv4-addr">
      <idmef:address>192.0.2.200</idmef:address>
    </idmef:Address>
  </idmef:Node>
  <idmef:Service ident="abc123-003">
    <idmef:port>21534</idmef:port>
  </idmef:Service>
</idmef:Source>
```

IDMEF: Ejemplo (III). Objetivo.

```
<idmef:Target ident="xyz789">
  <idmef:Node ident="xyz789-001" category="dns">
    <idmef:name>www.example.com</idmef:name>
    <idmef:Address ident="xyz789-002"
                   category="ipv4-addr">
      <idmef:address>192.0.2.100</idmef:address>
    </idmef:Address>
  </idmef:Node>
  <idmef:Service>
    <idmef:port>8080</idmef:port>
  </idmef:Service>
</idmef:Target>
```

IDMEF: Ejemplo (IV). Clasificación.

```
      <idmef:Classification text="phf attack">
        <idmef:Reference origin="bugtraqid">
          <idmef:name>629</idmef:name>
          <idmef:url>
          http://www.securityfocus.com/bid/629
          </idmef:url>
        </idmef:Reference>
      </idmef:Classification>
    </idmef:Alert>
  </idmef:IDMEF-Message>
```

Estándares: IDEF

IDEF: *The Future...*

- Aceptación 'formal' de IDEF como estándar.
- Aceptación de los fabricantes.
  $\Rightarrow$ Productos que 'hablen' IDEF.
- Procesamiento de alertas: ¿qué hacemos con los datos?

Conclusiones

- Resumen.
- Conclusiones.
- Referencias.

Conclusiones

Resumen

IDSes clasificados en función de...

- Técnica de análisis empleada:
  - Detección de usos indebidos.
  - Detección de anomalías.
- Origen de datos utilizados:
  - Sistemas basados en máquina.
  - Sistemas basados en red.

Conclusiones

Resumen

Dos grandes problemas del esquema:

- Modelo de elementos independientes, insuficiente hoy en día.
  - Sistemas distribuidos de detección de intrusos.
  - Elemento clave: estandarización.
- Sistemas de seguridad pasivos.
  - Respuesta automática: *Intrusion Prevention Systems*.
  - Peligros asociados...
  - ...y salvaguardas aplicables.

Conclusiones

- Importancia de la detección.
- Sistemas actuales: débiles (no detección de ataques nuevos) pero necesarios.
- IDS como mecanismo, no como producto concreto.
- El futuro:
  - Sistemas distribuidos.
  - Detección de nuevos tipos de ataques (*novel attacks*).
  - Refinamiento de la respuesta automática: IPS (*Intrusion Prevention Systems*).
  - Correlación.

Referencias

- Intrusion Detection: Network Security beyond the Firewall. Terry Escamilla. John Wiley & Sons, 1998.
- Network Intrusion Detection: An Analyst´s Handbook. Stephen Northcutt, Donald McLachlan, Judy Novak. New Riders Publishing, 2000 (2nd Edition).
- Intrusion Detection: Generics and State of the Art. NATO Research & Technology Organisation. Technical Report RTO–TR–049. NATO, 2002.
- Sistemas de detección de intrusiones. Diego González Gómez. GNU Free Documentation License, 2003.

Referencias

- COAST IDS pages:
  `http://www.cerias.purdue.edu/coast/intrusion-detection/welcome.html`
- Intrusion Detection Working Group:
  `http://www.ietf.org/html.charters/idwg-charter.html`
- The HoneyNet Project:
  `http://www.honeynet.org/`

¡¡Muchas gracias a tod@s!!

Antonio Villalón Huerta
toni@aiind.upv.es