Sistemas detección de intrusos

Roberto Gómez Cárdenas
rogomez@itesm.mx
http://webdia.cem.itesm.mx/ac/rogomez

¿Qué es un IDS?

- Intrusion Detection System
- Software especificamente diseñado para reconocer los patrones de un comportamiento no deseado.
- Pueden proporcionar un medio para registrar intentos, detener intentos en progreso y cerrar hoyos que coincidan con algunos patrones de ataques, bloqueando la secuencia para que ya no ocurra.
- Conjunto de herramientas para identificar y manejar riesgos.

¿Qué es una intrusión?

- Difícil de definir.
  - No hay un consenso.
  - Esto es un gran problema:
    - Que tal si alguien efectúa un telnet al sistema de usted…
      - Y trata de entrar al sistema como root ?
    - Que ocurre con un escaneo de pings ?
    - O un escaneo de algún producto comercial (ISS, Axent) ?
    - O que tal si alguien prueba phf en su webserver ?
      - Y que pasa sí phf funciona …
        - » Y el atacante se puede cargar al sistema.

Lámina 3 Dr. Roberto Gómez C.

¿Qué es un IDS ideal?

- El sistema ideal de detección de intrusos notificará al administrador de sistema/red la existencia de un ataque en proceso:
  - Con 100% de exactitud.
  - Inmediatamente (t<1min).
  - Con un diagnóstico completo del ataque.
  - Con recomendaciones de como bloquear el ataque.

Lástima que no existe!

Lámina 4 Dr. Roberto Gómez C.

Objetivos: Exactitud

- 100% de exactitud y 0% de falsos positivos.
  - Un falso positivo ocurre cuando el sistema genera una falsa alarma.
    - O lo que es lo mismo el síndrome de Juanito y el Lobo…
  - Generar un 0% de falsos positivos es trivial:
    - No detectar nada.
  - Generar un 0% de falsos negativos es un objetivo adicional:
    - No permitir que ningún ataque pase desapercibido.

Lámina 5 Dr. Roberto Gómez C.

Objetivos: Notificación Oportuna

- Para ser exacto en el diagnóstico, el IDS puede tener necesidad de "sentarse" sobre la información hasta que todos los detalles lleguen.
  - Implicaciones directas acerca de la definición de un IDS de tiempo real.
  - El IDS debe informar al usuario acerca de la demora.

Lámina 6 Dr. Roberto Gómez C.

Objetivos: Notificación Oportuna

- El canal de notificación debe estar protegido.
  - El atacante puede bloquear/inutilizar el mecanismo de notificación.
  - Un IDS que usa e-mail como canal de notificación va a tener problemas al informar que el servidor de correo esta siendo atacado.

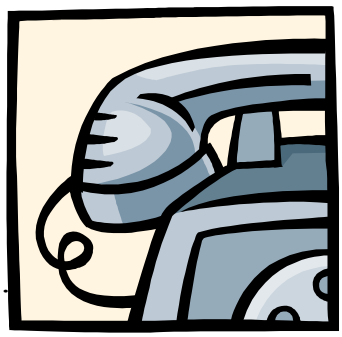


Lámina 7 Dr. Roberto Gómez C.

Objetivos: Diagnóstico

- Idealmente el IDS categorizará/identificará el ataque:
  - Demasiado detalle para el administrador de red, al tener que conocer íntimamente los ataques.
- Difícil de implementar:
  - Especialmente cuando las cosas "se ven raras" y no concuerdan con ataques conocidos.

Lámina 8 Dr. Roberto Gómez C.

Objetivos: el sueño

- El IDS perfecto no solo identificará un ataque, también:
  - Evalúa la vulnerabilidad del blanco.
  - Si el blanco es vulnerable informará al administrador.
  - Si la vulnerabilidad tiene un "parche" conocido, indicará al administrador como aplicarlo.
- Esto requiere cantidades tremendas de conocimiento incorporado al sistema.

Lámina 9 Dr. Roberto Gómez C.

IDS: a Favor

- Un IDS razonablemente efectivo puede identificar:
  - Ataques internos.
  - Ataques externos.
- Permite al administrador medir la cantidad de ataques que está sufriendo la infraestructura.
- Puede funcionar como respaldo de seguridad perimetral en caso de falla de otros sistemas (firewall, filtros).

Lámina 10 Dr. Roberto Gómez C.

IDS: en Contra

- No tienen como característica principal prevenir o bloquear ataques.
  - No son un reemplazo de firewalls, routers, etc.
- Si el IDS detecta problemas en la red interna, que debe hacer el administrador ?
  - Por definición, ya es demasiado tarde.

Lámina 11 Dr. Roberto Gómez C.

Paradigmas uso IDS

- Paradigmas ubicación IDS
  - Detección de ataques.
  - Detección de intrusiones
- Paradigmas de correlación de datos
  - IDES
  - Auditoría
  - Inline
  - Híbrido
- Paradigmas según la fuente de datos
  - host
  - red

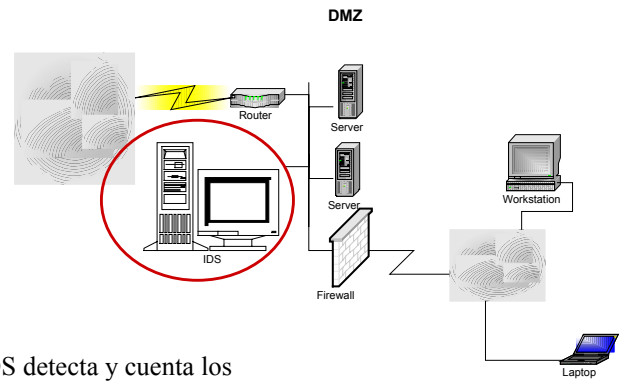
Lámina 12 Dr. Roberto Gómez C.

Paradigmas para el Uso del IDS

- Detección de ataques.
- Detección de intrusiones.

Lámina 13
Dr. Roberto Gómez C.

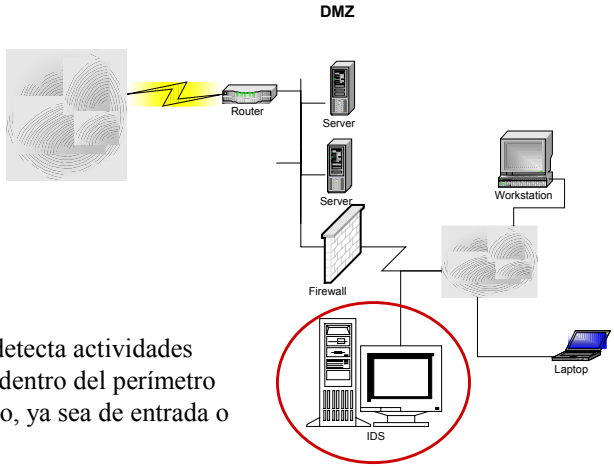
Detección de Ataques



El IDS detecta y cuenta los ataques contra la DMZ y el firewall.

Lámina 14
Dr. Roberto Gómez C.

Detección de Intrusos



El IDS detecta actividades ilegales dentro del perímetro protegido, ya sea de entrada o salida.

Lámina 15
Dr. Roberto Gómez C.

Detección de Ataques vs. Intrusos

- Idealmente efectuar las dos.
- Siendo realista, primero efectuar detección de intrusos y después detección de ataques.
  - O liberar primero la detección de ataque para justificar la decisión ante la directiva, después liberar la detección de intrusos.
- La pregunta importante tiene que ver con los costos de staff para reaccionar ante alarmas generadas por el sistema de detección de ataques.

Lámina 16
Dr. Roberto Gómez C.

Paradigma correlación datos

- IDES
  - define el IDS en términos de sujetos, objetos, perfiles, registros de Auditoría, registros de Anomalías, alarmas
- Auditoria
  - post procesamiento de la información de auditoría.
- Inline
  - procesa datos de auditoría conforme estos se van generando.
- Hibrido
  - explotan ambos esquemas: detección inline de eventos significativos que se envían a una estación de auditoría.

Lámina 17 Dr. Roberto Gómez C.

Paradigma fuente datos

- IDS Basado en Host
  - Tripwire
- IDS Basado en Red
  - Snort

Lámina 18 Dr. Roberto Gómez C.

IDS Basado en el Host

- Se obtienen los datos desde puntos ubicados dentro del sistema operativo.
  - Bitácoras de auditoría C2.
  - Bitácoras de sistema.
  - Bitácoras de aplicación.
- Los datos se recolectan de manera muy compacta.
  - Pero es dependiente del SO y de las aplicaciones.

Lámina 19 Dr. Roberto Gómez C.

IDS Basado en la Red

- Recolecta datos ya sea de la red o de un hub/switch.
  - Reensambla paquetes.
  - Examina encabezados.
- Intenta inferir que es lo que está ocurriendo a partir del contenido del tráfico de la red.
  - Identidades inferidas a partir de las acciones.

Lámina 20 Dr. Roberto Gómez C.

Paradigmas de Operación de los IDS

- Detección de anomalías o el enfoque de IA.
- Detección de mal uso o el enfoque fácil y sencillo.
- Alarmas contra robo o la detección basada en políticas.
- Honey Pots o el enfoque pásale a lo barrido.
- Híbridos

Lámina 21 Dr. Roberto Gómez C.

Detección de Anomalías

- Metas:
  - Analizar la red o sistema e inferir que es normal.
  - Aplicar medidas estadísticas o heurísticas a eventos subsecuentes, para determinar si estos concuerdan con el modelo o estadística de lo que es"normal".
  - Si los eventos se encuentran fuera de una ventana de probabilidad que determine lo que es normal, se genera una alerta.
    - Control configurable de falsos positivos.

Lámina 22 Dr. Roberto Gómez C.

Detección de Mal Uso

- Metas
  - Conocer que es lo que constituye un ataque.
  - Detectarlo
- Implementaciones típicas de mal uso:
  - Network Grep: búsqueda de strings en conexiones de red que puedan indicar que esta ocurriendo un ataque.
  - Reconocimiento de patrones: Codificar series de estados que son intercambiados durante el transcurso de un ataque.
    - E.g. el cambio de dueño de /etc/passwd
    - Open /etc/passwd para W
  - Su forma de operar es muy similar a los antivirus

Lámina 23 Dr. Roberto Gómez C.

Alarmas Contra Robo

- Es un sistema de detección de mal uso que tiene un blanco muy específico.
  - Puede no interesar alguien que escanee firewall desde exterior.
  - Puede no interesar alguien que escanee mainframe desde interior.
- Basadas en políticas, alertan al administrador sobre violaciones a estas.
- Detectan eventos que no necesariamente son violaciones de seguridad, pero si de políticas.
  - Nuevos ruteadores.
  - Nuevas subredes.
  - Nuevos servidores.

Lámina 24 Dr. Roberto Gómez C.

Honey Pots

- Es un sistema que fue configurado para atraer al atacante.
  - ecom.tienda.com.mx
  - transfers.banco.com.mx
  - tacacs.isp.net.mx
- Metas:
  - hacer que se vea atractivo al atacante.
  - hacer que se vea débil y fácil de atacar.
  - hacer que sea posible monitorear todo el tráfico que entra y sale del sistema.
  - alertar al administrador cada vez que alguien logra accesar al sistema.

Lámina 25 Dr. Roberto Gómez C.

IDS Híbridos

- La mayor parte de los IDS comerciales son de este tipo.
- Su fortaleza es en la detección de anomalías.
  - Estadística.
  - Demasiados falsos positivos.
- Los híbridos ideales deben incorporar lógica de los scaners de vulnerabilidades.

Lámina 26 Dr. Roberto Gómez C.

Consideraciones de implementación

- IDS y el WWW
  - fallan en presencia de SSL.
- IDS y Firewalls
  - eventualmente se fusionaran en una sola aplicación.
- IDS y VPNs
  - IDS basados en red tienen problemas en presencia de VPN's.
- IDS y VPNs
  - es difícil para un IDS basado en red grabar el tráfico de un red switcheada.
- IDs y desempeño
  - no operan bien en redes de alta velocidad.

Lámina 27 Dr. Roberto Gómez C.

Ejemplos IDS

- Tripwire
- Snort
- Portsentry
- Logsentry
- Hostsentry

Lámina 28 Dr. Roberto Gómez C.

Cuidando integridad: tripwire

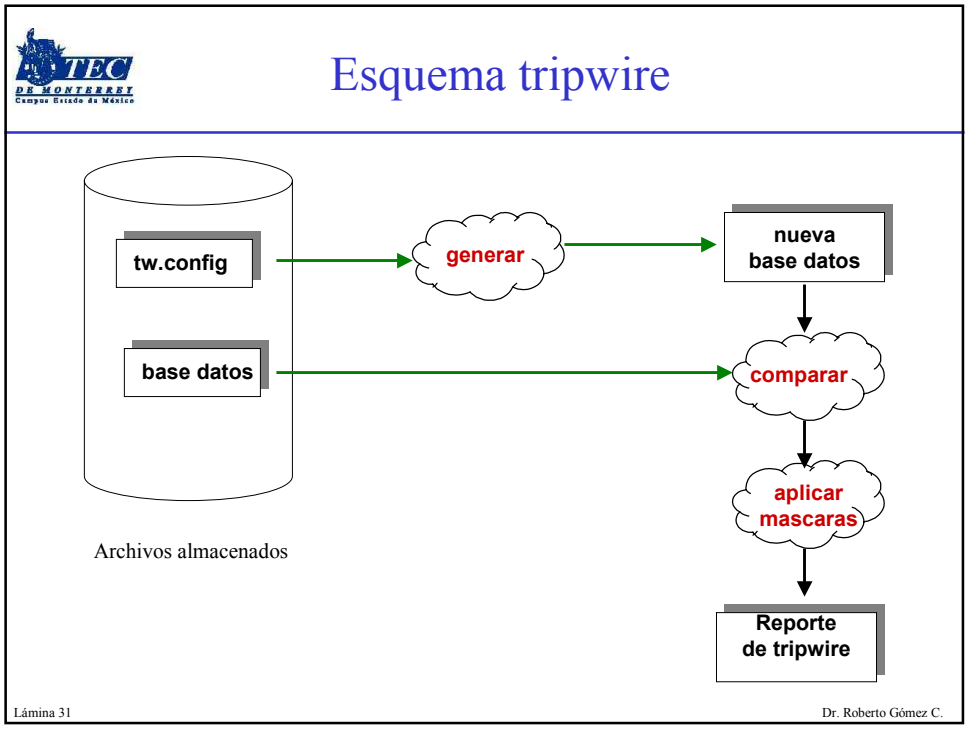
- Sistema verificador de integridad de archivos
  - compara un conjunto de estos objetos con la información sobre los mismos almacenada previamente en una base de datos
  - alerta al administrador en caso de que algo haya cambiado
- Tripwire es un programa que corre en la mayor parte de los sistemas Unix
- Escrito por Gene Kim y Gene Spafford en la Univ. de Purdue.

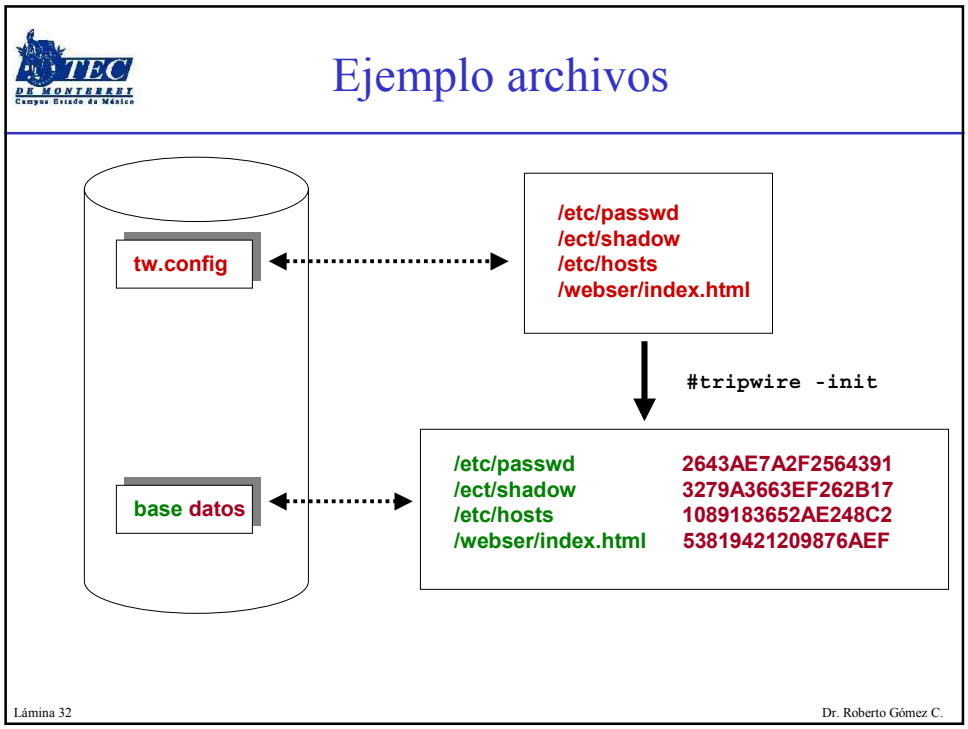
Lámina 29 Dr. Roberto Gómez C.

¿Cómo trabaja?

- Se crea un resumen (huella digital) de cada archivo o directorio importante
- Los resúmenes se almacenan en un medio seguro
  - un CD-ROM o un disco protegido contra escritura
- Si alguno de los archivos es modificado, Tripwire alertará la próxima vez que se lleve a cabo la comprobación
  - sustitución de un programa por una versión troyanizada o añade una entrada en nuestro fichero de contraseñas
- Para generar los resúmenes se usan funciones hash
  - MD2, MD4, MD5, Snefru, CRC-16 y CRC-32.

Lámina 30 Dr. Roberto Gómez C.


Esquema tripwire
tw.config
base datos
generar
nueva
base datos
comparar
aplicar
mascaras
Reporte
de tripwire
Archivos almacenados
Lámina 31
Dr. Roberto Gómez C.


Ejemplo archivos
tw.config
base datos
/etc/passwd
/ect/shadow
/etc/hosts
/webser/index.html
#tripwire -init
/etc/passwd 2643AE7A2F2564391
/ect/shadow 3279A3663EF262B17
/etc/hosts 1089183652AE248C2
/webser/index.html 53819421209876AEF
Lámina 32
Dr. Roberto Gómez C.

Primeros pasos

- Una vez hemos compilado el código fuente de Tripwire se debe inicializar la base de datos;
  - es necesario crear el fichero tw.config en la localización indicada en include/config.h,
  - es en ese archivo donde se especifican los directorios o archivos a verificar
- A continuación se inicializa la base de datos con la orden

tripwire –initialize (- init)

Lámina 33 Dr. Roberto Gómez C.

Ejemplo inicio base datos

```
root@cachafas:47# tripwire -init
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
### Phase 3:   Creating file information database
###
### Warning:   Database file placed in ./databases/tw.db_toto.
###
###          Make sure to move this file file and the configuration
###          to secure media!
###
###          (Tripwire expects to find it in '/usr/local/tw'.)
root@cachafas:48#
```

Lámina 34 Dr. Roberto Gómez C.

Algunos directorios a verificar

- /dev
- /etc (observar por pid files)
- /tmp /var/tmp /proc
- /usr/sbin
- /usr/local/bin
- /lib

Lámina 35 Dr. Roberto Gómez C.

¿Y después?

- Archivo ./databases/tw.db_toto se encuentran las funciones resumen de los archivos y directorios especificados en tw.config,
  - los datos de ese archivo se asumen como fiables,
  - por lo que es recomendable generarlo antes de abrir la máquina a los usuarios, nada más instalar el operativo.
  - si un usuario lo consigue modificar toda la seguridad de Tripwire se rompe
- Con la base de datos inicial ya generada, se puede ejecutar regularmente Tripwire para verificar que no se han presentado cambios

Lámina 36 Dr. Roberto Gómez C.

Ejemplo verificación

```
root@cachafas:48# tripwire &>resultados
root@cachafas:49# head -17 resultados
                ### Phase 1:   Reading configuration file
                ### Phase 2:   Generating file list
                ### Phase 3:   Creating file information database
                ### Phase 4:   Searching for inconsistencies
                ###
                ###                 Total files scanned:          4821
                ###                     Files added:          2
                ###                     Files deleted:        0
                ###                     Files changed:          4413
                ###
                ###                 After applying rules:
                ###                     Changes discarded:      3959
                ###                     Changes remaining:      458
                ###
                added:   -rw------- root          0 May  5 03:46:06 2000 /var/tmp/test
                changed: -rw-r--r-- root        972 May  5 03:49:53 2000 /var/adm/utmp
                changed: -rw-r--r-- root      10044 May  5 03:49:53 2000 /var/adm/utmpx
root@cachafas:50#
```

Lámina 37 Dr. Roberto Gómez C.

Actualizando

- Existen archivos o directorios que van a cambiar habitualmente.
  - archivo de contraseñas cada vez que añadamos a un usuario al sistema
- Dos mecanismos de actualización
  - modo interactivo: (opción –interactive) cada vez que Tripwire detecta un archivo con modificaciones pregunta si se desea actualizar la base de datos
  - modo actualización: el modo update se utiliza para la actualización de un archivo o de un directorio pasado como parámetro al ejecutable

Lámina 38 Dr. Roberto Gómez C.

Ejemplo modo interactivo

```
root@cachafas:50# tripwire -interactive
    ### Phase 1:   Reading configuration file
    ### Phase 2:   Generating file list
    ### Phase 3:   Creating file information database
    ### Phase 4:   Searching for inconsistencies
    ##
    ###                Total files scanned:          4820
    ##                    Files added:            1
    ##                    Files deleted:          0
    ##                    Files changed:           4413
    ##
    ###                After applying rules:
    ###                    Changes discarded:       3958
    ##                     Changes remaining:       457
    ###
    added:   -rw------- toni       32768 May  5 03:55:29 2000 /var/tmp/Rx0000755
            - ---> File: '/var/tmp/Rx0000755'
            - ---> Update entry?  [YN(y)nh?]
```

Lámina 39 Dr. Roberto Gómez C.

Ejemplo modo actualización

```
root@cachafas:51# tripwire -update /etc/passwd /etc/shadow
        ### Phase 1:   Reading configuration file
        ### Phase 2:   Generating file list
        Updating: update file: /etc/passwd
        Updating: update file: /etc/shadow
        ## Phase 3:   Updating file information database
        ###
        ## Old database file will be moved to `tw.db_anita.old'
        ###          in ./databases.
        ###
        ### Updated database will be stored in './databases/tw.db_anita'
        ###          (Tripwire expects it to be moved to '/usr/local/tw'.)
        ###
root@cachafas:52#
```

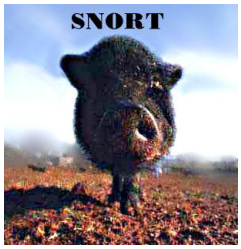
Lámina 40 Dr. Roberto Gómez C.

Recomendaciones uso tripwire

- Usar cron para ejecutar Tripwire en tiempos regulares
  - sistemas conectados a Internet ejecutarlo al menos una vez al día
  - posible enviar resultados por correo
  - asegurarse que se ejecute a intervalos diferentes
- Tripwire permite hacer verificaciones rápidas excluyendo algunos resumenes
  - útil si se ejecuta muy rápido con cron

Lámina 41 Dr. Roberto Gómez C.

Un IDS de red: snort



- Escrito por Martín Roesch (1998)
  - fundador de sourcefire (www.sourcefire.com)
- Es una version ligera de IDS que se basa en *libpcap*, y corre en UNIX
- Gratuito
- Puede lograr búsquedas de contenido en paquetes de IP
  - después a través de un logging, le avisa al administrador de seguridad si ha habido alguna actividad inusual

Lámina 42 Dr. Roberto Gómez C.

SNORT

- Provee a los administradores de seguridad con la información **suficiente** para tomar decisiones adecuadas.
- Puede ayudar cuando se encuentran hoyos de seguridad y no se haya liberado su "parche" o cuando por pólitica de seguridad no se pueda instalar el "parche" sin antes ser probado.
- Está disponible bajo la licencia GNU (General Public License) y su código fuente está disponible.

Lámina 43 Dr. Roberto Gómez C.

Plataformas

- Snort corre en casi cualquier versión de UNIX, incluyendo:
  - Linux (originalmente desarrollado para y en)
  - OpenBSD
  - FreeBSD
  - Solaris
  - HP-UX
  - AIX.
  - Windows (¿porque no?)

Plataformas de hardware como Sparc, Alpha y los x86.

Lámina 44 Dr. Roberto Gómez C.

Donde se obtiene

- Página snort
  - http://www.snort.org
- Las firmas se pueden bajar de
  - http://www.snort.org
  - http://www.whitehats.ca
  - http://www.silicondefense.com
- Requerimientos Windows
  - winpcap: http://netgroup-serv.polito
- Requerimientos Linux
  - libnet: http://www.packetfactory.net/libnet
  - libpcap: http://www.tcpdump.org/release/libpcap

Lámina 45 Dr. Roberto Gómez C.

Archivos principales de snort

- /etc/snort
  - contiene los archivos de configuración y reglas
- /var/log/snort
  - contendrá las bitácoras generadas por snort
- /usr/bin/snort
  - contiene el ejecutable de snort

Lámina 46 Dr. Roberto Gómez C.

Modos operación snort

- sniffer
  - solo lee los paquetes que circulan por la red y los despliega
- packet logger
  - almacena paquetes en el disco
- network intrusion detection system
  - analiza tráfico que coincida con una regla definida y realiza una determinada acción especificada en la regla

Lámina 47 Dr. Roberto Gómez C.

SNORT

- Los 3 elementos básicos de SNORT son:

1. decodificador de paquetes
2. motor de detección
3. subsistema de “logeo” y alerta

Lámina 48 Dr. Roberto Gómez C.

Elementos configuración snort

- Existen tres formas de indicarle a Snort como actuar
  - *snort.conf* configura variables, preprocesadores, salidas y conjuntos de reglas activas
  - archivos *.rules* define las huellas (signatures) actuales
  - opciones a nivel línea de comandos, las cuales sobreescriben las opciones en el archivo snort.conf
- Snort correrá con el archivo *snort.conf* por default
  - se aconseja configurarlo de acuerdo a la actividad de la red monitoreada
  - de esta forma se puede ahorrar tiempo al examinar los resultados

Lámina 49 Dr. Roberto Gómez C.

Ejemplo archivo snort.conf

```
# Step #1: Set the network variables:
# var HOME_NET $eth0_ADDRESS
var EXTERNAL_NET $HOME_NET

# Step #2: Configure preprocessors
preprocessor frag2
preprocessor bo

# Step #3: Configure output plugins
#
# output log_tcpdump: snort.log
# output trap_snmp: alert, 7, trap -v 2c -p 162  myTrapListener myCommunity

# Step #4: Customize your rule set
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
```

Lámina 50 Dr. Roberto Gómez C.

Comentarios snort.conf

- La indica a snort:
  - cual es la red local
  - que herramientas usar para preprocesar paquetes
    - paquetes fragmentados/defragmentados
  - que herramientas usar para formatear las bitácoras de salida
    - por ejemplo: habilitar XML
- Mayoría opciones: dejar la opción por default y solo ajustar la dirección red local, por ejemplo:

var HOME_NET [192.168.1.0/24,10.120.0.0/16]

Lámina 51 Dr. Roberto Gómez C.

Ejemplo archivo .rules

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP ISS Pinger";
content:"|495353504e475251|";itype:8;depth:32; reference:arachnids,158;
classtype:attempted-recon; sid:465; rev:1;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP L3retriever Ping";
content: "ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"; itype: 8; icode: 0; depth: 32;
reference:arachnids,311; classtype:attempted-recon; sid:466; rev:1;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP Nemesis v1.1 Echo";
dsize: 20; itype: 8; icmp_id: 0; icmp_seq: 0;
content: "|00000000000000000000000000000000|"; reference:arachnids,449;
classtype:attempted-recon; sid:467; rev:1;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize: 0;
itype: 8; reference:arachnids,162; classtype:attempted-recon; sid:469; rev:1;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP icmpenum v1.1.1"; id: 666;
dsize: 0; itype: 8; icmp_id: 666 ; icmp_seq: 0; reference:arachnids,450;
classtype:attempted-recon; sid:471; rev:1;)
```

Lámina 52 Dr. Roberto Gómez C.

Archivos de reglas

- bad-traffic.rules
- finger.rules
- smtp.rules
- dos.rules
- web-coldfusion.rules
- web-frontpage.rules
- web-attacks.rules
- icmp.rules
- backdoor.rules
- porn.rules
- attack-responses.rules

exploit.rules
ftp.rules
rpc.rules
ddos.rules
web-cgi.rules
web-iis.rules
sql.rules
netbios.rules
shellcode.rules
info.rules
local.rules

scan.rules
telnet.rules
rservices.rules
dns.rules
tftp.rules
web-misc.rules
x11.rules
misc.rules
policy.rules
icmp-info.rules
virus.rules

Lámina 53 Dr. Roberto Gómez C.

Comentarios reglas

- Si un paquete coincide con el criterio especificado por una regla
  - Snort lo almacena en un archivo de bitácora
- Si un paquete no coincide con ninguna regla
  - Snort no hace nada
  - no es necesario preocuparse por el tráfico que snort no esta configurado para reconocer
- Posible añadir las capacidades de snorts
  - incorporar reglas nuevas de los diferentes sitios que proporcionan una regla
  - posible crear sus propias reglas

Lámina 54 Dr. Roberto Gómez C.

Reglas de snort

- Flexibles y fáciles de modificar.
- Un ejemplo de regla:

```
alert tcp any any  -> 192.168.1.0/24 111
        ( content:"|00 01 86 a5|";
          msg "mountd access"; )
```

- Elementos antes paréntesis comprenden el encabezado de la regla.
- Elementos dentro paréntesis son las opciones de la regla
  - palabras antes del caracter "**:**" son keywords
  - esta sección no es forzosa para todas las reglas

Lámina 55 Dr. Roberto Gómez C.

Encabezados de las reglas

- Contiene información que define
  - el quien, donde y que de un paquete
  - la acción a tomar si un paquete cumple con la regla
- Formato:

```
accion protocolo  IP  puerto   -> IP  puerto
```

- El primer elemento en una regla es la acción de la regla.
- Se tienen definidas acciones por default

Lámina 56 Dr. Roberto Gómez C.

Acciones por default

accion protocolo IP puerto -> IP puerto

- alert
  - genera una alerta usando el método de alerta seleccionado y envía a una bitácora el paquete
- log
  - envía a una bitácora el paquete
- pass
  - ignora el paquete
- activate
  - alerta y activa otro regla dinámica
- dynamic
  - permanecer inactiva hasta activarse por otra regla, entonces actuar como una regla de tipo log

Lámina 57 Dr. Roberto Gómez C.

Protocolos

accion protocolo IP puerto -> IP puerto

- Existen cuatro protocolos que snort analiza buscando un comportamiento sospechoso
- Los protocolos actuales son:
  - TCP
  - UDP
  - ICMP
  - IP
- Se piensa incluir en un futuro
  - ARP, IGRP, GRE, OSPF, RIP, IPX

Lámina 58 Dr. Roberto Gómez C.

Las direcciones IP

accion protocolo IP puerto -> IP puerto

- Información dirección IP entrada y salida
- Palabra **any** puede definir cualquier dirección
- Operador negación **!**
  - cualquier dirección excepto la especificada
- Lista de direcciones **[** IP1, IP2, ... IP3 **]**
- Formadas por dirección numérica IP y bloque CIDR
  - bloque CIDR indica el netmask que debe aplicarse a las direcciones de las reglas
  - cada paquete que entra es probado contra la regla
  - /24: red clase C, /16 red clase B, /32 dirección específica

Lámina 59 Dr. Roberto Gómez C.

Ejemplos direcciones

- Ejemplo dirección con bloque CIDR

  192.168.1.0/24

  - bloque direcciones 192.168.1.0 a 192.168.1.255
- Ejemplo negación

```
alert tcp !192.168.1.0/24 any  -> 192.168.1.0/24 111 \
   ( content: “|00 01 86 a5|”; msg:”peligro”; )
```

- Ejemplo rango direcciones:

```
alert tcp ![ 192.168.1.0/24, 10.1.1.0/24] any  ->
          [ 192.168.1.0/24, 10.1.1.0/24] 111 \
   ( content: “|00 01 86 a5|”; msg:”peligro”; )
```

Lámina 60 Dr. Roberto Gómez C.

Numeros de puerto

accion protocolo IP puerto -> IP puerto

- Especificación del puerto de entrada y de salida.
- Pueden especificarse de varias formas
  - any, puertos estáticos, rangos y negación
- Keyboard **any**: cualquier número de puerto
- Puertos estáticos: un solo número de puerto, (21 para ftp, 80 para http, 23 para telnet)
- Rangos puertos: operador **:**
- Negación puerto: operador **!**

Lámina 61 Dr. Roberto Gómez C.

Ejemplos números de puerto

log udp any any -> 192.168.1.0/24 1:1024

- tráfico proveniente de cualquier puerto y cuyo puerto destino varía entre 1 y 1024

log udp any any -> 192.168.1.0/24 :6000

- tráfico proveniente de cualquier puerto dirigido a puertos menores o iguales a 6000

log udp any :1024 -> 192.168.1.0/24 500:

- tráfico proveniente de puertos privilegiados menores o iguales a 1024 dirigido a puertos mayores o iguales a 500

Lámina 62 Dr. Roberto Gómez C.

El operador de dirección

accion protocolo IP puerto -> IP puerto

- Indica la orientación o dirección del tráfico de la regla que se esta aplicando.
- Las direcciones y puertos del lado izquierdo corresponden a tráfico proveniente del host origen y del lazo derecho al host destino
- Existe un operador bidireccional <>
  - considerar los pares dirección/puerto tanto en una orientación destino como origan
  - usado para analizar una conversación de los dos lados (telnet, ftp, pop3)
  - ejemplo **log !192.168.1.0/24 any <> 192.168.1.0/24 23**

Lámina 63 Dr. Roberto Gómez C.

Opciones de las reglas

- Forman el corazón de la máquina de detección de intrusión.
- Todas las opciones están separadas por el caracter "**;**"
- Las keywords de las opciones están separadas por el carácter "**:**"
- Cuenta con 35 keywords

Lámina 64 Dr. Roberto Gómez C.

Keywords de las opciones

- msg
- logto
- ttl
- tos
- id
- ipoption
- fragbits
- dsize
- flags
- seq
- ack
- itype
- icode icmp_id
- icmp_seq
- content
- content-list
- offset
- depth
- nocase
- session
- rpc
- resp
- react
- reference
- sid
- rev
- classtype
- priority
- uricontent
- tag
- ip_proto
- sameip
- stateless
- regex

Lámina 65 Dr. Roberto Gómez C.

Ejemplos reglas snort

- Un ejemplo de content

```
alert tcp any any -> 192.168.1.9/24 143
  ( content: "|90C8 C0FF FFFF|/bin/sh"; msg: "IMAP buffer overflow");
```

- Un ejemplo de ack

```
alert tcp any any -> 192.168.1.0/24 any
    ( flags: A; ack: 0, msg: "NMAP TCP ping")
```

- Un ejemplo de rpc

```
alert tcp any any -> 192.168.1.0/24 111
    ( rpc: 100000,*,3;  msg: "RPC getport (UDP)";)
```

Lámina 66 Dr. Roberto Gómez C.

Los formatos de las alertas

- Formato tcpdump

```
11:53:49:869667 eth0 > 192.168.0.231.12242 > 192.168.1.10.www:
   s  6373380:6373380(0) win 8192
   <mss 1460, nop, nop, sackOK>  (DF)
```

- Formato snort

```
[**] IDS024 – RPC – portmap-request-ttdbserv [**]
07/27 – 13:33:58.314512 10.0.0.69:896 -> 192.168.38.15:111
UDP TTL:64 TOS: 0x0 ID:33481   Len 64

[**] rwwwshell CGI access attempt [**]
06/10 – 07:55:01.284025 62.0.183.93:1526  ->  208.237.191.52:80
TCP     TTL:64    TOS: 0x0   ID:4816       DF
 *****PA*   Seq: 0xF3156AC9    Ack: 0x9B63081     Win:  0x7078
47  45  54  20  2F  63   67  69  2D  62  69  6E  2F  72  77  77   Get /cgi-su
77  73  68  65  6C  6C  2E  70  6C  20  48  54  54  50  2F  31   bin/rww
2E  30  0A 0A  02  00   00  00                                    wshell.pl
                                                                  HTTP/1
```

Lámina 67 Dr. Roberto Gómez C.

Formato TCPdump

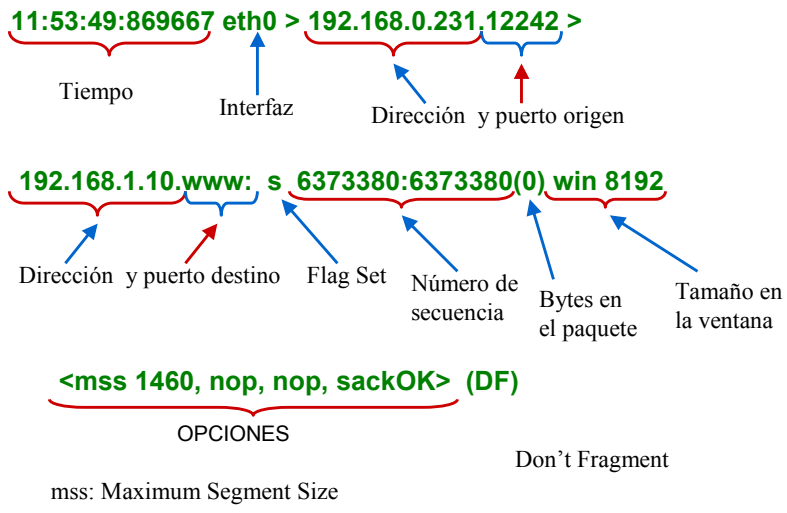


Lámina 68 Dr. Roberto Gómez C.

Formato Snort (1)

[**] IDS024 – RPC – portmap-request-ttdbserv [**]

Nombre de la alerta

07/27 – 13:33:58.314512 10.0.0.69:896 -> 192.168.38.15:111

Fecha y tiempo

FUENTE
Dirección y puerto origen

orientación tráfico

DESTINO
Dirección y puerto destino

UDP **TTL:64** **TOS: 0x0** **ID:33481**

tipo protocolo | time to live | tipo de servicio | identificador de sesión

Len 64

longitud

Lámina 69
Dr. Roberto Gómez C.

Formato Snort (2)

[**] rwwwshell CGI access attempt [**]

Nombre de la alerta

06/10 – 07:55:01.284025 62.0.183.93:1526 -> 208.237.191.52:80

Fecha y tiempo | FUENTE | DESTINO

TCP **TTL:64** **TOS: 0x0** **ID:4816** **DF**

protocolo | time to live | servicio | id sesión | don't fragment

*******PA*** **Seq: 0xF3156AC9** **Ack: 0x9B63081** **Win: 0x7078**

banderas | # de secuencia | # acknowledgement | tamaño ventana

```
47 45 54 20 2F 63 67 69 2D 62 69 6E 2F 72 77 77   Get /cgi-su
77 73 68 65 6C 6C 2E 70 6C 20 48 54 54 50 2F 31   bin/rww
2E 30 0A 0A 02 00 00 00                           wshell.pl
                                                  HTTP/1
                                                  .0 .....
```

hex payload

Formato comprensible por el humano

Lámina 70
Dr. Roberto Gómez C.

Salidas de una sesion telnet

Salida TCPdump de una sesión telnet

```
20:59:49.153313 0:10:4:b:d:a9:66 0:60:97:7:c2:8e0800 125:192.168.1.3.23 >
  192.168.1.4.1031: P76:147(71) ack 194 win 17514 (DF) [tos 0x10] (ttl 64,
  id 660) 4510 006f 0294 4000 4006 b48d c0a8 0103 c0a8 0104 0017 0407
  df4a 6536 b3a6 fd01 5018 446a d2ad 0000 fffa 2203 03e2 0304 820f 07e2
  1c08 8204 09c2 1a0a 827f 0b82 150f 8211 1082 13ff f00d 0a46 7265 6542
  5344 2028 656c 7269 632e 686f 6d65 2e6e 6574 2920 2874 7479 7030
  290d 0a0d 0a
```

Salida snort de una sesión telnet

```
20:59:49.153313 0:10:4B:D:A9:66 -> 0:60:97:7:C2:8E type: 0x800 len: 0x7D
  192.168.1.3:23 -> 192.168.1.4:1031 TCP TTL:64 TOS:0x10 DF ***PA*
  Seq: 0xDF4A6536 Ack: 0xB3A6FD01 Win: 0x446A
FF FA 22 03 03 E2 03 04 82 0F 07 E2 1C 08 82 04   ..".............
09 C2 1A 0A 82 7F 0B 82 15 0F 82 11 10 82 13 FF   ................
F0 0D 0A 46 72 65 65 42 53 44 20 28 65 6C 72 69   ... FreeBSD (elri
63 2E 68 6F 6D 65 2E 6E 65 74 29 20 28 74 74 79   c.home.net)(tty
70 30 29 0D 0A 0D 0A                              p0)....
```


Ejemplo contenido /var/log/snort/alert

```
# snort  -c snort.conf –s –h 192.168.1.0/24 [1]

[**] [1:469:1] ICMP PING NMAP [**][Classification:
Attempted Information Leak] [Priority: 2]03/28-
09:48:40.739935 192.168.1.2 -> 192.168.1.3ICMP
TTL:46 TOS:0x0 ID:61443 IpLen:20
DgmLen:28Type:8  Code:0  ID:10629   Seq:0
ECHO[Xref =>
http://www.whitehats.com/info/IDS162]

[**] [1:469:1] spp_portscan: PORTSCAN DETECTED
from 192.168.1.2 (THRESHOLD 4 connections
exceeded in 0 seconds) [**]03/28-09:48:41.052635

[**] [100:2:1] spp_portscan: portscan status from
192.168.1.2: 183 connections across 1 hosts:
TCP(183), UDP(0) [**]03/2809:48:45.007501
```

[1]Almacenar bitácoras en /var/log/snort (default) y enviar alertas al archivo por default /var/log/snort/alert

Contenido parcial de /var/log/snort/portscan.log

snort –c /etc/snort/snort.conf –l /var/log/snort/

```
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:106 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:193 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:138 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:128 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:156 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:35 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:48 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:16 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:173 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:72 SYN ******S*
Mar 28 09:48:41 192.168.1.2:45061 -> 192.168.1.3:65 SYN ******S*
```

Lámina 73 Dr. Roberto Gómez C.

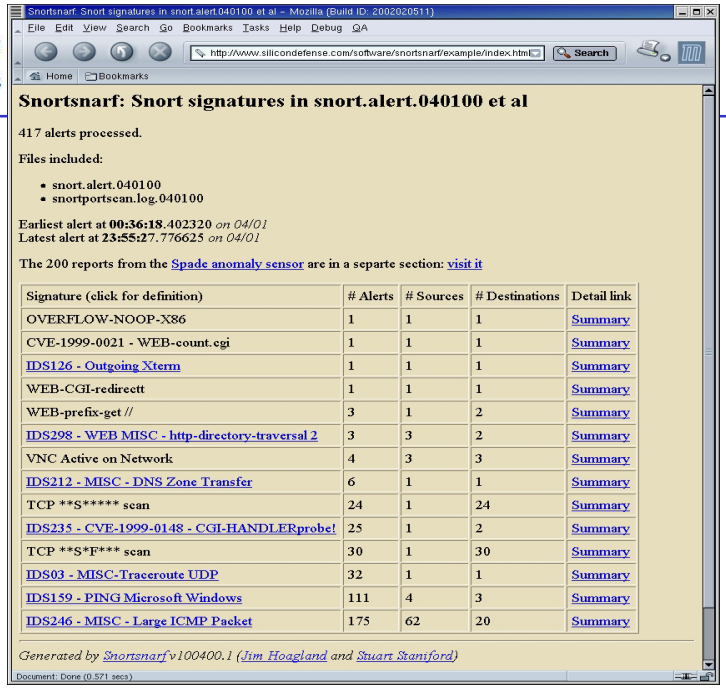
Software relacionado con snort y producido por terceros

- SnortSnarf
- ACID
- Demarc
- Ethereal
- SQL

Lámina 74 Dr. Roberto Gómez C.

SnortSnaff

- Producido por Silicon Defense
- Es un programa en Perl que toma archivos de alertas de Snort y produce reportes en formato HTML
- La salida esta dirigida a una inspección de diagnóstico
- Silicon Defense también proporciona sensores con soporte comercial
- Página:
  - http: www.silicondefense.com/software/snortsnarf/



ACID

- Analysis Console for Intrusion Databases (ACID)
- Máquina de análisis basada en PHP
- Busqueda y procesamiento en un base de datos de eventos de seguridad generados por diferentes IDSes, firewalls y herramientas de monitoreo de red.
- Generador de queries e interfaz de busqueda, visualizador de paquetes (decodificador), administrador de alertas, generador de gráficas y estadísticas
- Página:
  - http://acidlab.sourceforge.net/

Lámina 77 Dr. Roberto Gómez C.

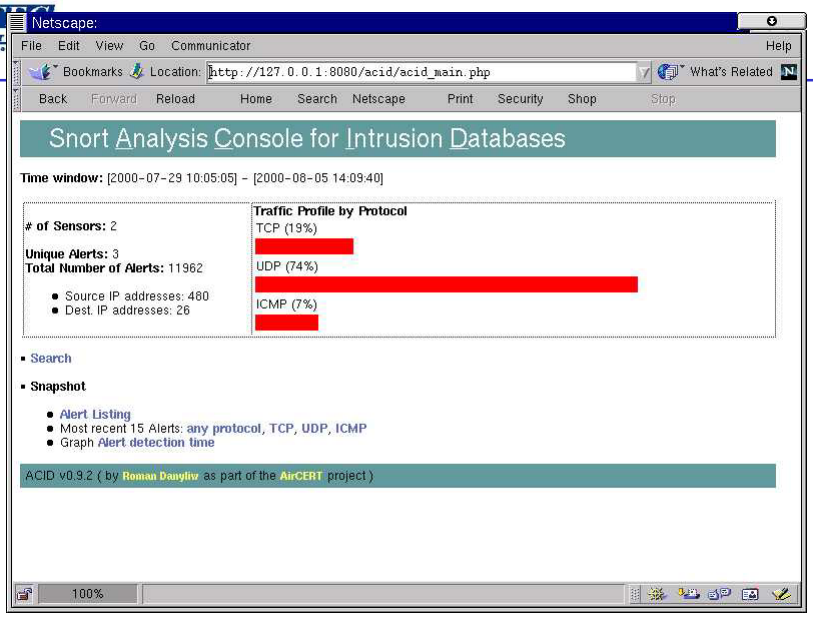


Lámina 78 Dr. Roberto Gómez C.

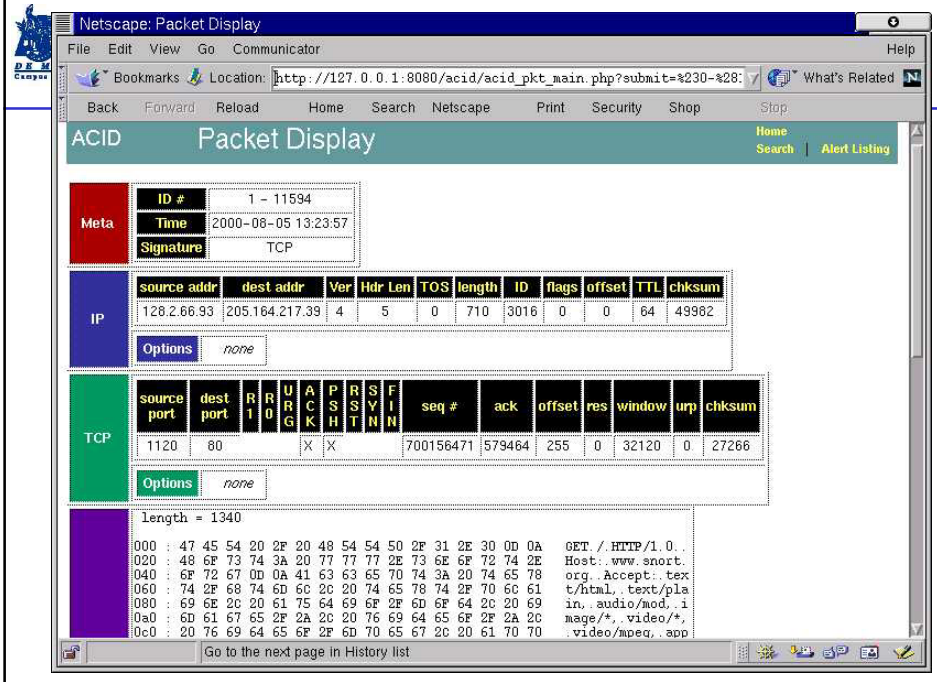


Lámina 79 Dr. Roberto Gómez C.

Demarc

- Consola de administración de NIDS
- Integra Snort con el poder y conveniencia de una interfaz centralizada para todos los sensores de red.
- Monitorea todos los servidores/hosts para asegurarse que servicios de red como correo y web permanezcan accesibles todo el tiempo.
- Monitorea bitacoras de sistema en busca de entradas anormales que pueden indicar intrusos o malfuncionamiento del sistema
- Página
  - http://www.demarc.com

Lámina 80 Dr. Roberto Gómez C.

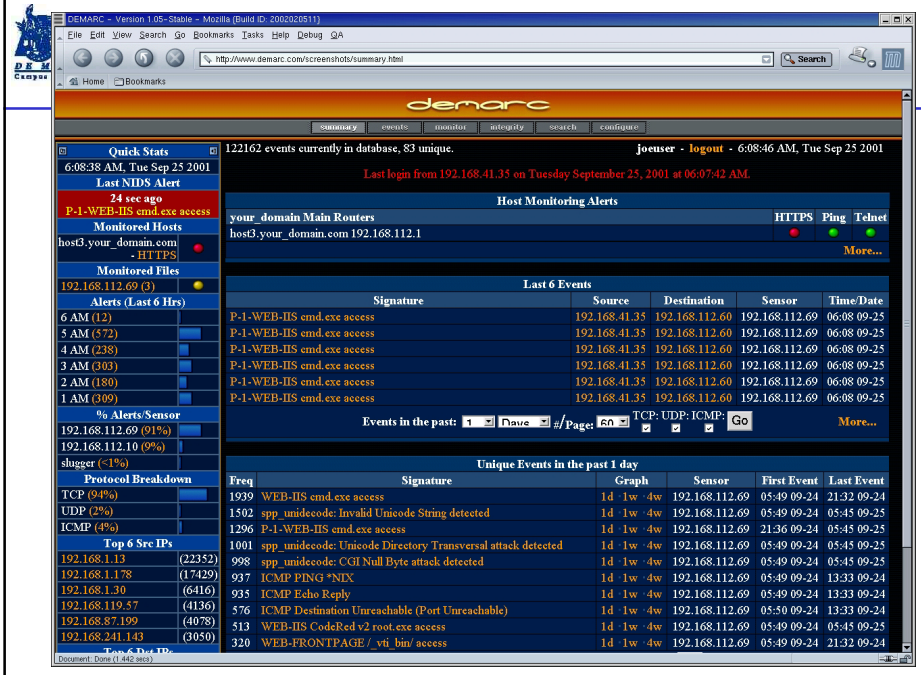


Lámina 81 Dr. Roberto Gómez C.

Lámina 82 Dr. Roberto Gómez C.

Ethereal

- No es un elemento de Snort
- Es uno de los mejores GUI, open source, para visualizar paquetes
- Paquetes
  - http://www.ethereal.com
  - Para windows: www.ethereal.com/distribution/win32/ethereal-setup-0.9.2.exe
  - Para Unix: www.ethereal.com/download.html
  - Red Hat Linux RPMs: ftp.ethereal.com/pub/ethereal/rpms/

Lámina 83 Dr. Roberto Gómez C.

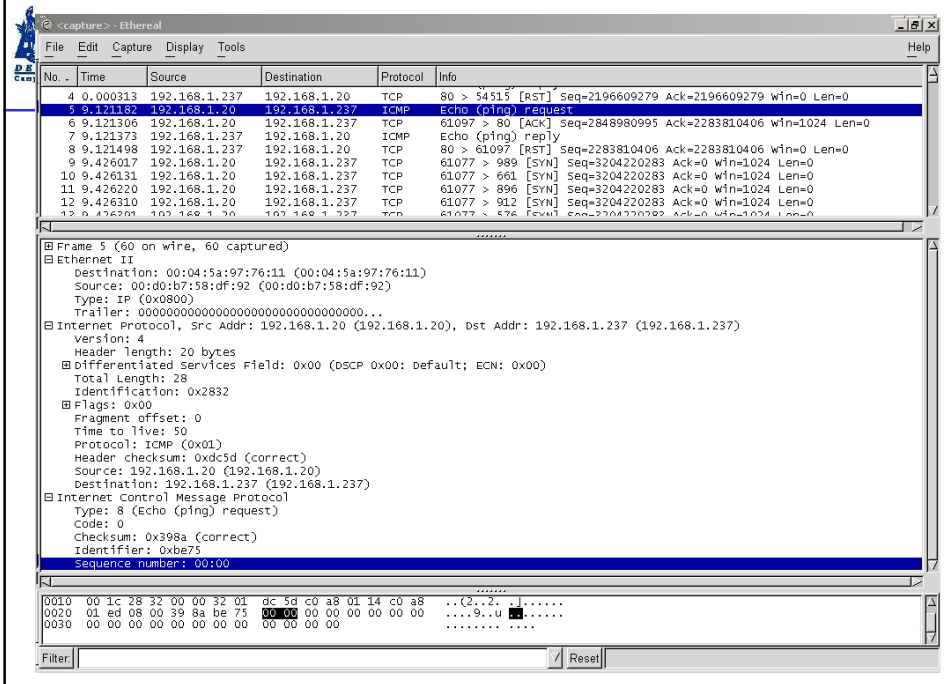


Lámina 84 Dr. Roberto Gómez C.

SQL

- Módulo para enviar salidas a una variedad de bases de datos SQL.
- Módulo desarrollado por Jed Pickel.
- Después es posible consultar la base de datos a través de queries
  - los resultados dependen de la base de datos
- El formato es

  databese: <log | alert>, <database type>, <parameter list>

Lámina 85
Dr. Roberto Gómez C.

Psionic PortSentry

- Tercer componente de la suite Abacus
  - logcheck y hostsentry son los otros dos
- Detecta y guarda un log de los escaneos de puertos,
- Posible configurar para que bloquee la máquina atacante haciendo difícil el completar un escaneo de puertos.
  - no se recomienda ya que se podría utilizar para generar un ataque de denegación de servicio en hosts legítimos
- Disponible en:
  - http://www.psionic.com/abacus/portsentry/

Lámina 86
Dr. Roberto Gómez C.

Logsentry

- Oiginalmente conocido como LogSentry
- Basado en programa auditoria frequentcheck.sh
- Automatiza auditoria auditoria archivos bitácoras a través de búsqueda de keywords predefinidos dentro de bitácoras de mensajes y de correos
  - puede reportar entradas que contienen algún keyword en específico y entradas que no contienen otros keywords
- Dos partes: script logcheck.sh y archivo binario logtail
  - lee mensajes recientes de bitácoras, buscando mensajes de syslog o PortSentry
  - binario guarda última posición de cada archivo revisado

Lámina 87 Dr. Roberto Gómez C.

Hostsentry

- Herramienta detección intrusos basada en host que proporciona Login Anomaly Detection (LAD).
- Permite detectar
  - logins no usuales, dominios y directorios sospechosos, así como intentos de logins desconocidos
- Usa base datos basada en Python para dar seguimiento a la actividad del usuario.
- La base de datos “aprende” el comportamiento de login de los usuarios.
  - comportamiento usado por firmas modulares para detectar eventos inusuales

Lámina 88 Dr. Roberto Gómez C.

Conclusión: IDS

- Son una herramienta que hay que saber utilizar.
- No son un silver bullet.
- Requieren de mucho soporte por parte de los administradores.
- Requieren de conocimiento profundo del comportamiento de la red/sistemas.
- Es importante hacer una buena evaluación de la ubicación del IDS.
- Sigue siendo una tecnología joven.

Lámina 89 Dr. Roberto Gómez C.

Bibliografía

- Network Intrusion Detection, Northcutt et al, New Riders.
- Intrusion Signatures and Analysis, Northcutt et al, New Riders.
- Internet Security and Firewalls, Cheswick and Bellovin, Addison Wesley.
- TCP/IP Illustrated Vol. 1, Stevens, Addison Wesley.

Lámina 90 Dr. Roberto Gómez C.