Departamento de
Automática y Computación
Automatika eta
Konputazio Saila

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

Práctica 7: Sistemas de detección de intrusiones

1- Introducción

La seguridad total no existe, es más bien un estado mental: te crees que estás seguro, y descuidas la seguridad de tus sistemas, o eres consciente de la realidad y no te lo crees, con lo que pondrás los medios para complicar el trabajo de los posibles atacantes y detectar sus intentos de intrusión.

Si un *hacker* quiere entrar en nuestro sistema, y es lo suficientemente habilidoso y paciente, tarde o temprano lo conseguirá si no hacemos nada para remediarlo. Cuantas más medidas tomemos encaminadas a la protección de nuestro sistema más le costará a un intruso penetrar en él y cuanto más le cueste, en tiempo y en intentos, más posibilidades tendremos de detectarlo y, por extensión, de neutralizarlo.

Una vez hayamos desplegado todas nuestras medidas de protección interna (políticas de contraseñas, control de la información pública, concienciación de los usuarios del sistema, etc.) y perimetral (*firewalls*, *honeypots*, ACLs, etc.) deberemos dar un paso más y monitorizar el comportamiento del sistema, para poder detectar los posibles ataques. Aquí es donde entran en juego los sistemas de detección de intrusión. Su finalidad no es evitar que se produzcan los ataques o intrusiones, sino detectarlos e informar al responsable de seguridad (si es posible en tiempo real).

Los *honeypots*, que se utilizan para engañar al atacante y confundirlo, pueden funcionar también como detectores de intrusiones. Si incluimos en nuestra red una máquina física dedicada en exclusiva a ejecutar un *honeypot*, sin ninguna otra funcionalidad, cualquier tráfico que entre o salga de dicha máquina será casi con total seguridad un ataque. Además, este tipo de IDS presenta algunas ventajas frente a los IDSs convencionales, ya que los *honeypots* nos permiten investigar las técnicas empleadas por los atacantes, y mientras les tenemos ocupados hackeando las máquinas del *honeypot* no comprometen nuestras máquinas físicas.

Sin embargo un *honeypot*, por sí solo, no cubre todas las funcionalidades de los IDSs. Un Sistema de Detección de Intrusión se emplea para intentar detectar todo el tráfico dañino que no pueda ser identificado por un cortafuegos convencional. Esto incluye ataques contra servicios vulnerables, ataques orientados a aplicaciones concretas, *logins* no autorizados, acceso a ficheros con información sensible, la presencia de *malware* (virus, troyanos y gusanos), etc.

Un IDS suele tener sensores virtuales (por ejemplo, un *sniffer* de red) con los que el núcleo del IDS puede obtener datos externos, generalmente sobre el tráfico de red. Este tráfico pasa por un analizador donde es comparado con patrones (firmas) de ataques conocidos, u otros comportamientos sospechosos, como pueden ser el escaneo de puertos, paquetes malformados, etc. El IDS no sólo analiza a qué tipo de tráfico pertenece cada paquete, sino que también revisa el contenido y su comportamiento.

Sin embargo, un IDS es incapaz de detener los ataques por sí solo, por lo que una vez detectada una posible intrusión, existen dos opciones: mandar una señal de alerta, que se almacena en una base de datos, y avisar al responsable de seguridad, como hacen los *sistemas pasivos*, o bien, poner en marcha las contramedidas que frustren el ataque (alterar en tiempo real las reglas de los cortafuegos, enviar paquetes RST para terminar las conexiones TCP, etc.), como hacen los *sistemas reactivos*.

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

2- Sistemas de Detección de Intrusión basados en Red (NIDS)

Se basan en dos tipos de técnicas. La primera de ellas consiste en la búsqueda de patrones conocidos y tiene como virtud principal la precisión de detección. Por contra, no es eficaz frente a ataques nuevos o que no estén en la base de datos. La segunda técnica se basa en realizar un análisis estadístico del tráfico con el objeto de intentar descubrir actividades anómalas. Su mayor virtud es que puede detectar ataques desconocidos. El precio a pagar es que no es un método del todo preciso y da lugar a falsos positivos.

La herramienta que vamos a manejar es un NIDS, muy conocido en el mundo de la seguridad informática, denominado *snort* (http://www.snort.org). Trabajaremos con *snort* en los PCs A, B ó C.

Para familiarizarnos con él, podemos comenzar leyendo los documentos USAGE, README y FAQ de la documentación (en /usr/share/doc). Si lo que deseamos es adquirir un conocimiento más avanzado de la herramienta podemos consultar el manual en formato pdf, ***también*** disponible en el directorio de documentación o ***en la página web***.

Antes de nada vamos a crear un directorio llamado `snort` en nuestra zona de trabajo (`/home/ssi`) de nuestro PCB, donde copiaremos los ficheros del directorio `/etc/snort` y la documentación del programa:

```
$ cp -r /etc/snort/* /home/ssi/snort
$ cp -r /usr/share/doc/snort-2.8.1 /home/ssi/snort
```

En dicho directorio guardaremos los ficheros de logs y de alertas que se generen durante la práctica. Para conseguir esto, a la hora de ejecutar *snort* tendremos que utilizar la opción -l para cambiar el directorio por defecto, y la opción -u para que el usuario efectivo del proceso sea ssi y, de esta forma, tener permisos para leer los ficheros mencionados. Además, tendremos que ajustar el valor de la variable RULE_PATH, para poder obtener los privilegios que nos permitan editar cualquier fichero de reglas.

Snort necesita un fichero de configuración para trabajar en modo de detección de intrusiones. Encontraremos en nuestro directorio una plantilla ampliamente comentada que constituye un buen punto de partida, no sólo para trabajar, sino también para aprender a activar los distintos componentes del programa. Dicha plantilla es snort.conf. Tendremos que editarla y realizar las modificaciones que consideremos necesarias.

A la hora de definir un escenario en el que realizar nuestras pruebas, lo más interesante será configurar una red con tres máquinas (PCs A, B y C) .

3- Snort como sniffer de red

Antes de utilizar snort como NDIS, para lo que necesitará editar su archivo de configuración snort.conf(lo hará en el siguiente apartado), vamos primero a comprobar su funcionamiento como sniffer de red.

```
$ sudo snort -v
```

Con esta opción -v iniciamos snort en modo sniffer visualizando en pantalla las cabeceras de los paquetes TCP/IP (cabeceras IP, TCP, UDP y ICMP). Si queremos, además, visualizar los campos de datos que pasan por la interface de red, añadimos -d.

```
$ sudo snort -vd
```

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

Si consigue cerrar snort mediante "Ctrl + C", podrá ver un interesante listado de estadísticas correspondientes al tráfico capturado.

Checkpoint 7.1: Pruebe snort como sniffer de red, capturando el tráfico, de una comunicación entre PCA y PCC, desde PCB.

Nota: en PCB no deberá tener configurada ip alguna, tan solo activar la interfaz que desee utilizar, asegúrese de ello. Para PCA y PCC utilice el espacio de direcciones 10.3.armario.0/24

También es posible utilizar filtros definiendo qué puertos escuchar o que tráfico escuchar basado en ip origen o ip destino, por ejemplo:

```
$ sudo snort -vd host 10.3.armario.51 and dst port 8080
```

Mediante el comando `nc(man nc)` ponga a PCA escuchando en el puerto 8080 y conéctese, también empleando el comando `nc`, desde PCC a dicho puerto de PCA.

En PCB debería ver la conexión entre PCA y PCC.

4- Trabajando con snort

Ahora, familiarícese con la herramienta *snort* y configúrela en modo de detección de intrusiones.

Para ejecutar snort en modo detección de intrusiones tenemos que definir qué tipo de alertas generar, existen varios tipos: -A fast (alertas mostrando información básica) –A console (se envian al syslog de la máquina) –A full (el máximo de información) y –A none (desactiva las alertas).

Ejemplo de ejecución de snort.

```
$ sudo snort -A fast -u ssi -l /home/ssi/snort/log -c snort.conf
```

Al lanzar este ejemplo, encontrará varios errores relacionados con archivos de reglas, incluidos por defecto en `snort.conf`, que no existen, y que deberá de comentar en dicho archivo de configuración. Así como la biblioteca libsf_engine.so y el preprocesador ssl. Realizada esta edición inicial vuelva a lanzar el ejemplo.

Podrá comprobar que con la configuración por defecto no se detectan los escaneos de puertos.

- ¿Qué cambios hay que realizar al fichero de configuración plantilla `snort.conf` para que sí se detecten los escaneos de puertos?
- ¿Qué escaneos `nmap` de tipo TCP son detectados con éxito? Verifíquelo mediante:

```
$ nmap -v <direcciónIP>
```

Checkpoint 7.2: Muestre al profesor que snort genera una alerta cuando se produce un escaneo de puertos.

Realice los cambios necesarios en el fichero de configuración para que snort detecte arp spoofing(ataque que analizaremos en la próxima práctica). Para probarlo, además de cambiar el comportamiento de snort, tiene que crear el siguiente escenario:

1. Sitúa los equipos A, B y C en el switch0 (en la misma VLAN)
2. Configura dirección ip a PCA y PCB, utilizando el espacio de direcciones 10.3.armario.0/24
3. Asigna también ip a PCC dentro del mismo espacio de direcciones que PCA y PCB.

Campus de Arrosadía
Arrosadiko Campusa
31006 Pamplona - *Iruñea*
Tfno. 948 169113, Fax. 948 168924
Email: ayc@unavarra.es

4. Abre un socket en PCB con el comando: `sudo nc -l 8080`
5. Conéctate desde PCA al puerto 8080: `nc <ipPCB> 8080`
6. Así tendrá algo parecido a un chat entre PCB y PCA. Compruébelo.
7. Ejecuta en PCC el comando ettercap: `sudo ettercap -T -M arp // //`
8. Podrás observar que PCC es capaz de ver lo que PCA y PCB se envían.

La actividad del PC C debe haber sido detectada por `snort`(en PCB) si has activado correctamente la detección arp spoofing.

Checkpoint 7.3: Mostrar al profesor que snort genera una alerta cuando se produce un ataque basado en arp spoofing.

Vamos a suponer ahora que su jefe quiere saber si sus empleados hablan sobre él en algún chat y que le ha encargado que lo averigüe. Una forma de hacerlo es añadir reglas en snort para que se detecte la cadena de caracteres “jefe” en cualquier paquete TCP o UDP. Especifique qué reglas hay que añadir para conseguir lo que su jefe le ha planteado.

Checkpoint 7.4: Mostrar al profesor que snort genera una alerta cuando en alguna conexión TCP se envía la palabra “jefe”.