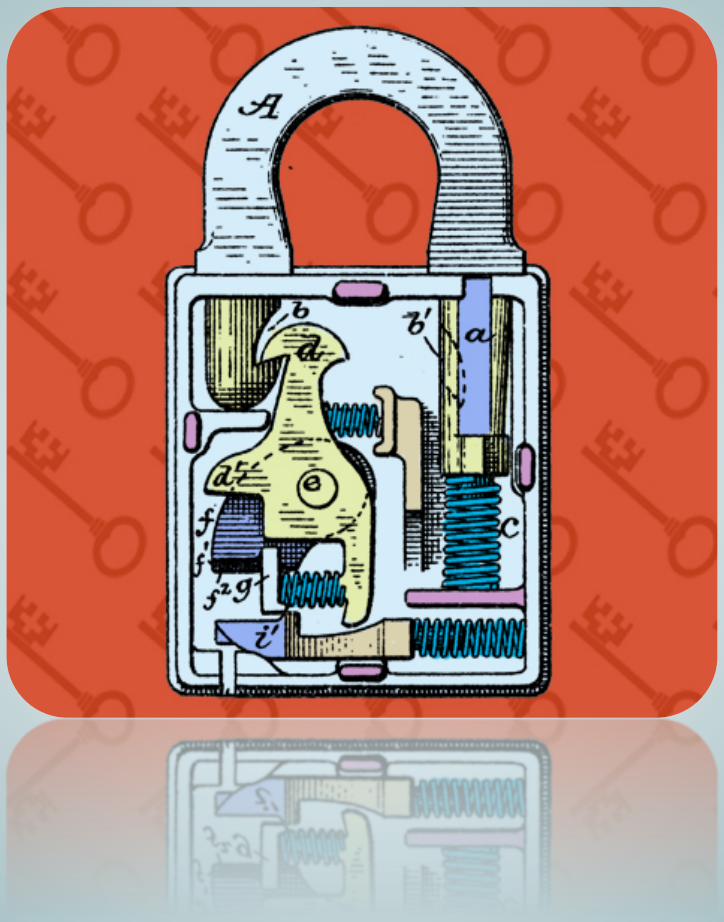
Garantía y Seguridad en Sistemas y Redes

Práctica 8. Detección de Intrusión



Esteban Stafford

Departamento de Ingeniería
Informática y Electrónica

Práctica 8

Detección de Intrusión

1. Introducción

2. Objetivos

Los objetivos fundamentales que persigue esta práctica son los siguientes:

- Conocer la manera de funcionamiento de un IDS sencillo.
- Aprender a crear reglas de IDS propias.

3. Material

Para esta práctica se utilizarán varios programas pero principalmente el IDS `suricata`. Este es un IDS más moderno que el estándar de-facto en el mundo open-source `snort`. Pero que respeta el formato de las reglas de éste. Esto permite migrar de `snort` a `suricata` con un esfuerzo razonable. Más información sobre `suricata` se puede obtener en: http://suricata-ids.org, particularmente en la sección de documentación.

Por otro lado, puede resultar interesante tener una máquina virtual con Kali. Esta es una distribución para los expertos en seguridad. Es una evolución de la antigua distribución Backtrack. Kali tiene una gran cantidad de herramientas destinadas a hacer ataques en el marco de auditorías de seguridad. Más información y descargas en: https://www.kali.org/.

Para hacer pruebas, se va a ejecutar `suricata` sobre trazas de red, en lugar de directamente sobre un interfaz de red. Para ello hay que tener alguna herramienta de captura de trazas como `ngrep`, `tcpdump` o Wireshark, que son capaces de guardarlas en formato `pcap`.

4. Desarrollo

4.1. Instalación de `suricata`

En esta práctica se usará la versión del IDS `suricata` que esta disponible en los repositorios de Ubuntu. Para instalarlo, simplemente se ejecuta:

```
sudo apt-get install suricata
```

Esta versión tiene un arranque de tipo UNIX SysV. Es decir que se controla con el script `/etc/init.d/suricata`. Además, como es típico en Debian se configura con el archivo `/etc/default/suricata`. En este archivo tenemos que hacer varios cambios.

```
RUN=yes
LISTENMODE=pcap
IFACE=eth1
```

En este archivo también se ve donde esta la configuración del demonio `suricata`, que también necesita cambios. Así que antes de arrancar conviene revisar algunos aspectos de `/etc/suricata/suricata-debian.yaml`. Primero se ve que el demonio va a utilizar el directorio `/var/log/suricata` para dejar información acerca de su funcionamiento. Observa la sección `outputs:` para ver qué ficheros van a ser generados en este directorio. Por otro lado, un servicio debería usar `syslog` para notificaciones, para ello hay que cambiar la sección `outputs:` dentro de `logging:` como se indica a continuación.

```
outputs:
- console:
    enabled: no
- file:
    enabled: no
    filename: /var/log/suricata.log
- syslog:
    enabled: yes
    facility: local5
    format: "[%i] <%d> -- "
```

Otro aspecto que especifica este fichero de configuración es el directorio en donde se almacenarán las reglas del IDS (`default-rule-path:`), que conviene cambiar a `/etc/suricata/rules` y descomprimir ahí el archivo https://rules.emergingthreats.net/open/suricata/emerging.rules.tar.gz.

Finalmente aparece una declaración de variables. Obsérvala y modifica el valor de `$HOME_NET` para que solo especifique el interfaz de la máquina virtual.

Con todos estos cambios, se puede arrancar el servicio con

```
/etc/init.d/suricata start
```

4.2. Disparo de reglas del IDS

Con el IDS en marcha, observa los diferentes ficheros de `/var/log/suricata`. Una vez conocida su función, desde la máquina anfitrión trata de disparar alguna regla. Para ello revisa el contenido de algún fichero de `/etc/suricata/rules`. El formato de dichos ficheros se puede encontrar en la documentación de `suricata`. Para hacer ataques sofisticados, también se puede arrancar una máquina virtual con Kali y realizarlos desde ahí. Asegúrate de que puedes disparar diez reglas.

4.3. Creación de reglas

Una vez comprendido la manera en la que se escriben las reglas de `suricata`, conviene saber escribir reglas propias. Así, si se encuentra algún tipo de ataque que el IDS no detecta, se puede crear una regla al efecto y compartirla. Para la creación de reglas es importante poder trabajar en un entorno aislado y repetible, ya el desarrollo de reglas

puede ser un proceso con multiples iteraciones. Como `suricata` se puede ejecutar sobre trazas de red, en lugar de sobre un interfaz de red real, vamos a crear un entorno de trabajo para desarrollo de reglas. Primero se crea un directorio en `$HOME`, por ejemplo, en donde guardar toda la configuración, las reglas y los logs. Dentro de este directorio, hacemos una copia de `/etc/suricata/suricata-debian.yaml` y lo modificamos para que use la consola, en lugar de `syslog`, y los paths de `default-rule-path:` y `default-log-dir:` apuntan dentro del directorio de trabajo. Modifica también la lista de ficheros de reglas para que solo exista uno: `gssr.rules` y crea un fichero vacío con ese nombre dentro del directorio de reglas.

Antes de ejecutar `suricata` con la configuración anterior necesitamos una traza de red del ataque que queremos detectar. En este caso, simplemente vamos a dar la alarma si alguien entra a nuestro servidor Web con `wget`. Entonces con `ngrep`, `tcpdupm` o similar, guarda la traza de un acceso web con `wget` al servidor de la máquina virtual. Guardalo en `wget.pcap`. Del mismo modo guarda una traza en `firefox.pcap` con los paquetes que genera `Firefox` al acceder al servidor.

Ahora puedes ejecutar `suricata` con el siguiente comando.

```
suricata -c suricata-debian.yaml -r wget.pcap
```

Observa que no se produce ninguna alerta. Modifica el fichero `gssr.rules` para incorporar una regla que se dispare cuando se procesa la traza `wget.pcap` pero no cuando se procesa `firefox.pcap`.