INGENIERIA EN SISTEMAS Y CIENCIAS DE LA COMPUTACION

UNIVERISDAD MARIANO GALVEZ DE GUATEMALA

Seguridad de sistemas



***Implementación Suricata***

FREDY FERNANDO FUENTES M.

2890-07-16474

23/11/18

## Instalar Suricata IDS en Opensuse Leap 15

Con incesantes amenazas de seguridad, los sistemas de detección de intrusos (IDS) se han convertido en uno de los requisitos más críticos en entornos de centros de datos de hoy en día. Sin embargo, a medida que más y más servidores de actualización de sus tarjetas de red a 10 GB / 40 GB Ethernet, cada vez es más difícil de poner en práctica la detección de intrusiones informáticas intensivas por hardware a velocidades de línea. Un enfoque para escalar el rendimiento IDS es **IDS** **multi-hilo,** donde la carga de trabajo de inspección profunda de paquetes intensiva de la CPU está paralelizado en múltiples tareas simultáneas. Dicha inspección paralelizada puede aprovechar el hardware multi-núcleo para ampliar IDS throughput fácilmente. Dos esfuerzos de código abierto bien conocidos en esta área son [Suricata](http://list.xmodulo.com/suricata.html) y [Bro](http://list.xmodulo.com/bro.html).

En este tutorial, voy a demostrar **cómo instalar y configurar Suricata IDS en Opensuse leap 15**

Vamos a crear Suricata desde el código fuente. En primer lugar, es necesario instalar varias dependencias necesarias de la siguiente manera.

Ejecutar :

zypper install -n wget tar gcc pkg-config pcre-devel libyaml-devel \

libpcap-devel zlib-devel file-devel make libnetfilter\_queue-devel \

libjansson-devel mozilla-nss-devel libcap-ng-devel lua53-devel

Suricata 4

Una vez que instale todos los paquetes requeridos, seguir adelante e instalar Suricata de la siguiente manera.

En primer lugar, descargar el último código fuente Orcinus de <http://suricata-ids.org/download/>, y construirlo. Al escribir estas líneas, la última versión era suricata-4.0.5

Ejecutar :

wget https://www.openinfosecfoundation.org/download/suricata-4.0.5.tar.gz

tar -xvf suricata-4.0.5.tar.gz

cd suricata-4.0.5

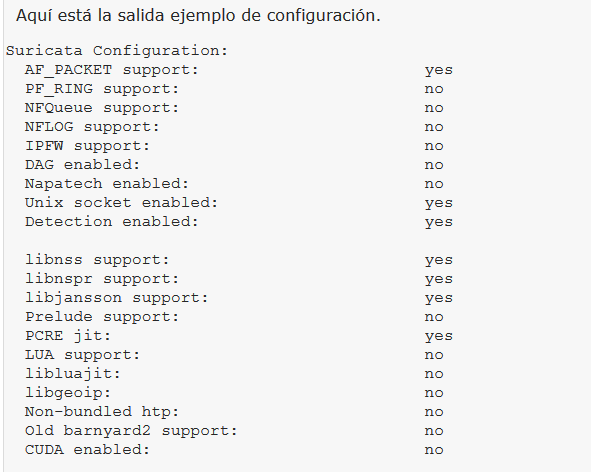
Para Leap 15

Ejecutar :

CPPFLAGS="-I/usr/include/libnetfilter\_queue/ -I/usr/include/libnfnetlink/" ./configure \

--prefix=/usr --sysconfdir=/etc --localstatedir=/var --enable-nfqueue --enable-lua

Aquí está la salida ejemplo de configuración.



Ahora compilarlo e instalarlo.

## Make

Ejecutar :

make

sudo make install

sudo ldconfig

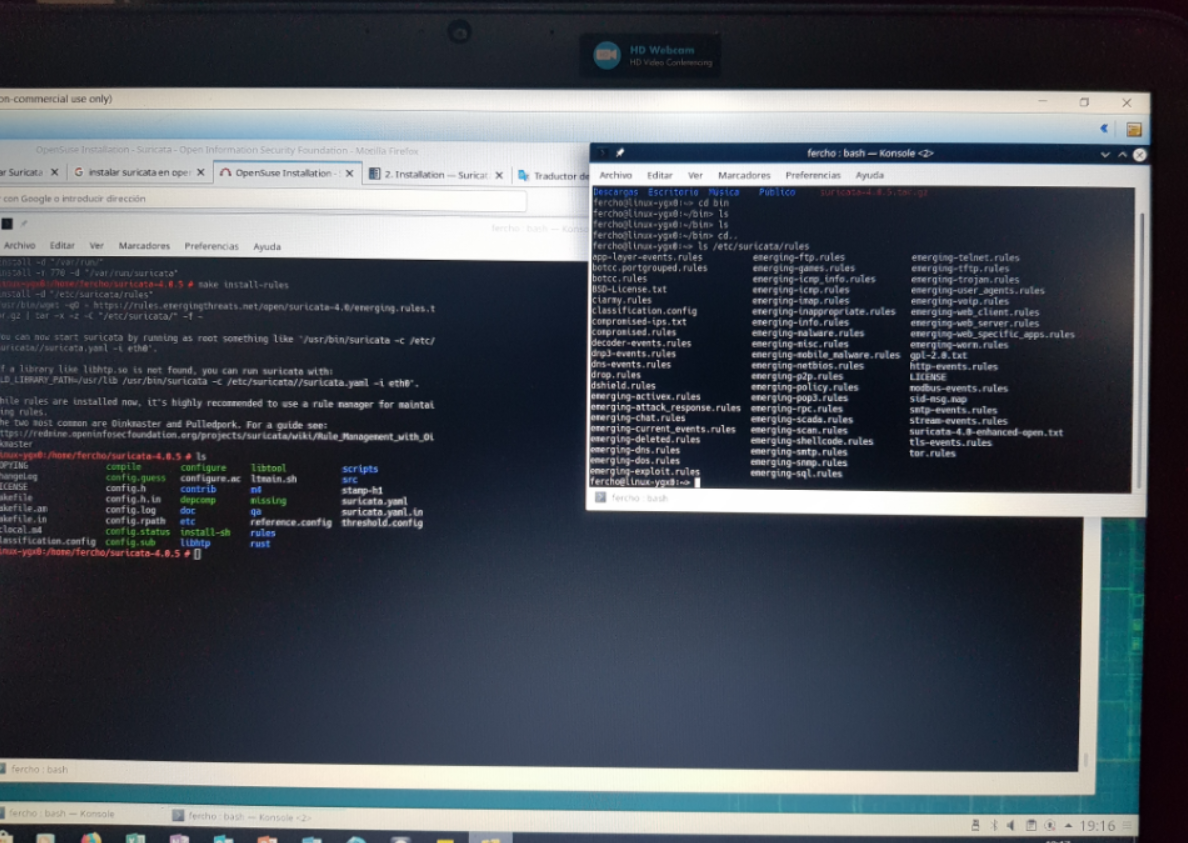
El código fuente de Suricata viene con archivos de configuración por defecto. Vamos a instalar estos archivos de configuración por defecto de la siguiente manera.

make install-conf

Como saben, Suricata es inútil sin los conjuntos de reglas del IDS. Convenientemente, el Makefile trae la opción de instalación con las reglas IDS. Para instalar reglas de IDS, ejecute el siguiente comando.

make install-rules

El comando de instalación de regla anterior descarga la instantánea actual de los conjuntos de reglas disponibles en la comunidad de [EmergingThreats.net](https://translate.googleusercontent.com/translate_c?act=url&depth=1&hl=es&ie=UTF8&prev=_t&rurl=translate.google.es&sl=en&tl=es&u=http://rules.emergingthreats.net/&usg=ALkJrhjyW8_VhdT9M7SM-jJLmb36efGClw) , y los almacena en /etc/suricata/rules



## Configurar Suricata IDS por primera vez

Ahora es el momento de configurar Suricata. El archivo de configuración se encuentra en **/etc/suricata/suricata.yaml.** Abra el archivo con un editor de texto para su edición.

$ sudo vi /etc/suricata/suricata.yaml

Éstas son algunas reglas de configuración básica para que usted pueda comenzar.

La palabra clave "default-log-dir" debe apuntar a la ubicación de los archivos de registro Suricata.

default-log-dir: /var/log/suricata/

Bajo la sección "vars", encontrará varias variables importantes utilizados por Suricata. "HOME\_NET" debe apuntar a la red local para ser inspeccionado por Suricata. "!$ HOME\_NET" (asignado a EXTERNAL\_NET) se refiere a cualquier otra red que la red local. "XXX\_PORTS" indica el número de puerto(s), utilizado por diferentes servicios. Tenga en cuenta que Suricata puede detectar automáticamente el tráfico HTTP independientemente del puerto que utiliza. Por lo tanto, no es crítica para especificar la variable HTTP\_PORTS correctamente.

vars:

HOME\_NET: "[192.168.122.0/24]"

EXTERNAL\_NET: "!$HOME\_NET"

HTTP\_PORTS: "80"

SHELLCODE\_PORTS: "!80"

SSH\_PORTS: 22

La sección de "host-os-policy" se utiliza para defenderse de algunos ataques bien conocidos que explotan el comportamiento de pila de red de un sistema operativo (por ejemplo, el reensamblaje TCP) para evadir la detección. Como contramedida, los IDS modernos recurrieron a la llamada inspección "por objetivos" "target-based", cuando el motor de inspección afina su algoritmo de detección basado en un sistema operativo de destino del tráfico. Por lo tanto, si usted sabe lo que los OS anfitriones locales individuales están ejecutando, se puede alimentar esa información para Suricata para mejorar potencialmente la tasa de detección. Esto es cuando se utiliza la sección "host-os-policy". En este ejemplo, la política de IDS por defecto es Linux; si no hay información del OS es conocido por una dirección IP en particular, Suricata aplicará la inspección basada en Linux. Cuando se captura el tráfico de 192.168.122.0/28 y 192.168.122.155, Suricata se aplicará la política de inspección basado en Windows.

host-os-policy:

# These are Windows machines.

windows: [192.168.122.0/28, 192.168.122.155]

bsd: []

bsd-right: []

old-linux: []

# Make the default policy Linux.

linux: [0.0.0.0/0]

old-solaris: []

solaris: ["::1"]

hpux10: []

hpux11: []

irix: []

macos: []

vista: []

windows2k3: []

Bajo la sección "threading", puede especificar [afinidad de CPU](https://translate.googleusercontent.com/translate_c?act=url&depth=1&hl=es&ie=UTF8&prev=_t&rurl=translate.google.es&sl=en&tl=es&u=http://xmodulo.com/run-program-process-specific-cpu-cores-linux.html&usg=ALkJrhhFcBA0hOlNouLpzIMiIUUc7QR0WQ) para los diferentes hilos Suricata. Por defecto, la afinidad de la CPU está desactivado ( "set-cpu-affinity: no"), lo que significa que las discusiones de Suricata serán programados en cualquiera de núcleos de CPU disponibles. De forma predeterminada, Suricata creará un hilo "detectar" para cada núcleo de CPU. Puede ajustar este comportamiento especificando "detect-thread-ratio: N". Esto creará N \* M detectar hilos, donde M es el número total de núcleos de CPU en el host.

threading:

set-cpu-affinity: no

detect-thread-ratio: 1.5

Con la configuración de roscado anteriores, Suricata creará 1,5 \* hilos de detección M, donde M es el número total de núcleos de CPU en el sistema.

Para obtener más información sobre la configuración Suricata, se puede leer el archivo de configuración por defecto en sí, que está fuertemente comentó para mayor claridad.

## Realizar detección de intrusiones con Suricata

Ahora es el momento de probar a ejecutar Suricata. Antes de su lanzamiento, hay un paso más que hacer.

Cuando se utiliza el modo de captura de pcap, es muy recomendable desactivar cualquier característica de paquetes offloead (por ejemplo, LRO / GRO) en la tarjeta de red, que está escuchando en Suricata, ya que esas funciones pueden interferir con la captura de paquetes en vivo.

Así es cómo puede desactivar LRO/GRO  en la interfaz de red eth0:

$ sudo ethtool -K eth0 gro off lro off

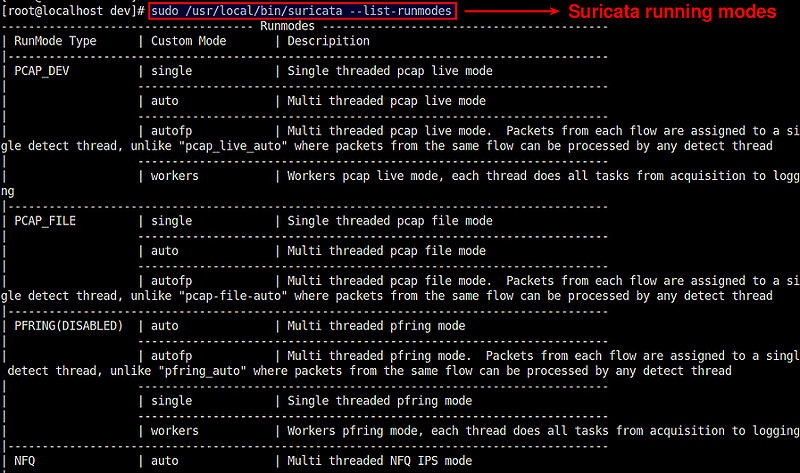
Tenga en cuenta que dependiendo de su tarjeta de red, es posible que aparezca el siguiente mensaje de advertencia, que puede pasar por alto. Simplemente significa que la tarjeta de red no es compatible con LRO.

No se puede cambiar a gran recibir-offload

Cannot change large-receive-offload

Suricata es compatible con varios modos de funcionamiento. Un RunMode determina cómo los diferentes hilos se utilizan para el IDS. El siguiente comando muestra todos los [runmodes disponibles](https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Runmodes) .

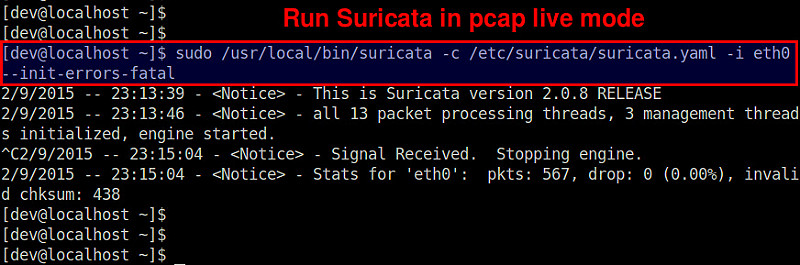
$ sudo /usr/local/bin/suricata --list-runmodes

[](https://translate.googleusercontent.com/translate_c?act=url&depth=1&hl=es&ie=UTF8&prev=_t&rurl=translate.google.es&sl=en&tl=es&u=https://www.flickr.com/photos/xmodulo/20481140934/&usg=ALkJrhiHIaLeDPGos1NWgLR-UfCPGxbM9w)

El RunMode predeterminado utilizado por Suricata es autofp (que significa "auto equilibrio para flujos altos de carga" o" auto flow pinned load balancing"). En este modo, los paquetes de cada flujo distinto se asignan a una sola hilo "detectar". Los flujos se asignan a las discusiones con el menor número de paquetes sin procesar.

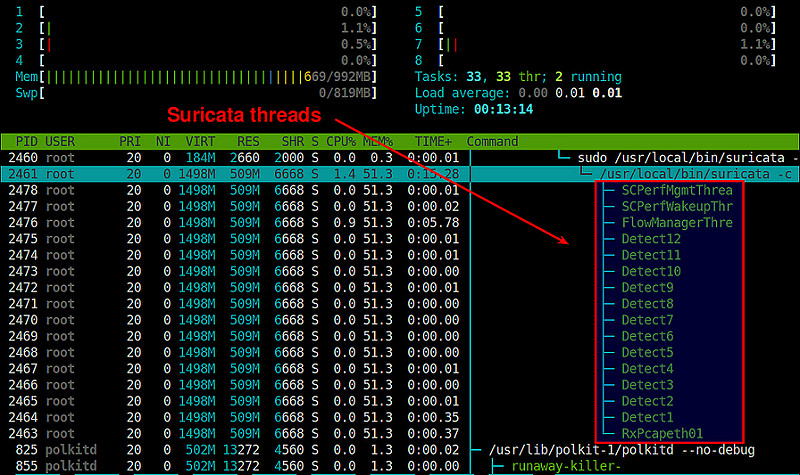
Por último, vamos a comenzar Suricata, y a verlo en acción.

$ sudo /usr/local/bin/suricata -c /etc/suricata/suricata.yaml -i eth0 --init-errors-fatal

[](https://translate.googleusercontent.com/translate_c?act=url&depth=1&hl=es&ie=UTF8&prev=_t&rurl=translate.google.es&sl=en&tl=es&u=https://www.flickr.com/photos/xmodulo/21077552366/&usg=ALkJrhgik8eviHAfaSpn8305rvkq-cCffA)

En este ejemplo, estamos monitoreando una interfaz eth0 red en un sistema de 8 núcleos. Como puede observarse, Suricata crea 13 hilos de procesamiento de paquetes y 3 hilos de gestión. Los hilos de procesamiento de paquetes consisten en un hilo PCAP captura de paquetes, y 12 detectan hilos (igual a 8 \* 1.5). Esto significa que los paquetes capturados por un hilo de captura son de equilibrio de carga de 12 hilos para detectar IDS. Los hilos de gestión son un gestor de flujo y dos contadores / Estadísticas hilos relacionados.

Aquí hay un hilo-vista del proceso de Suricata (tramado por [H TOP](https://translate.googleusercontent.com/translate_c?act=url&depth=1&hl=es&ie=UTF8&prev=_t&rurl=translate.google.es&sl=en&tl=es&u=http://ask.xmodulo.com/view-threads-process-linux.html&usg=ALkJrhiDnRm_36Skc-AQdGdUFHyMiRRaiw) ).

[](https://translate.googleusercontent.com/translate_c?act=url&depth=1&hl=es&ie=UTF8&prev=_t&rurl=translate.google.es&sl=en&tl=es&u=https://www.flickr.com/photos/xmodulo/20482669593/&usg=ALkJrhjzF_K-7JtypJkbrJrLTIYpbDXA1Q)

los registros de detección Suricata se almacenan en / var / directorio / log suricata.

/var/log/suricata/fast.log $ tail-f

01/04 / 2015-15: 47: 12.559075 [\*\*] [1: 2.200.074: 1] SURICATA TCPv4 suma de comprobación no válida [\*\*] [Clasificación: (null)] [Prioridad: 3] {} TCP 172.16.253.158:22 -> 172.16.253.1:46997

01/04 / 2015-15: 49: 06,565901 [\*\*] [1: 2.200.074: 1] SURICATA TCPv4 suma de comprobación no válida [\*\*] [Clasificación: (null)] [Prioridad: 3] {} TCP 172.16.253.158:22 -> 172.16.253.1:46317

01/04 / 2015-15: 49: 06,566759 [\*\*] [1: 2.200.074: 1] SURICATA TCPv4 suma de comprobación no válida [\*\*] [Clasificación: (null)] [Prioridad: 3] {} TCP 172.16.253.158:22 -> 172.16.253.1:46317

Para facilitar la importación, el registro también está disponible en formato JSON:

$ tail -f /var/log/suricata/eve.json

{"timestamp":"2015-04-01T15:49:06.565901","event\_type":"alert","src\_ip":"172.16.253.158","src\_port":22,"dest\_ip":"172.16.253.1","dest\_port":46317,"proto":"TCP","alert":{"action":"allowed","gid":1,"signature\_id":2200074,"rev":1,"signature":"SURICATA TCPv4 suma de comprobación no válida "," categoría: "severidad" "," ": 3}}

{"timestamp":"2015-04-01T15:49:06.566759","event\_type":"alert","src\_ip":"172.16.253.158","src\_port":22,"dest\_ip":"172.16.253.1","dest\_port":46317,"proto":"TCP","alert":{"action":"allowed","gid":1,"signature\_id":2200074,"rev":1,"signature":"SURICATA TCPv4 suma de comprobación no válida "," categoría: "severidad" "," ": 3}}