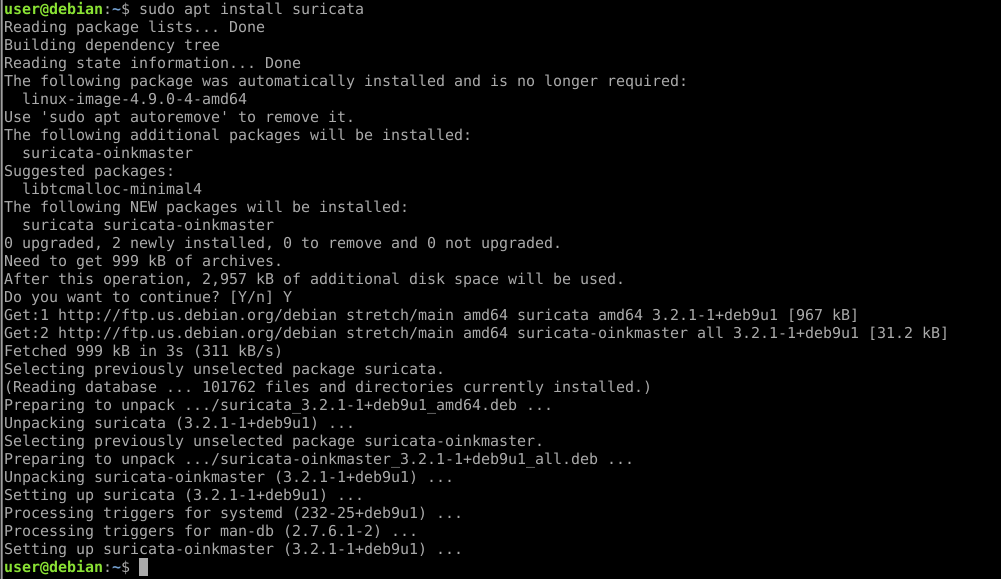
Instalamos Suricata

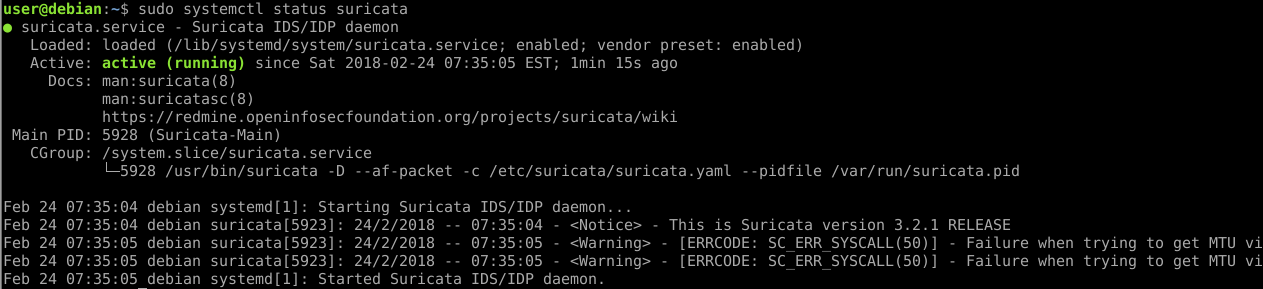
apt-get install suricata



Instalación de Suricata IDS en Debian

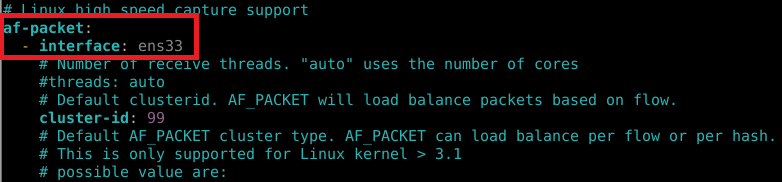
Podemos comprobar que tenemos el servicio corriendo, y por tanto todo ha ido bien, mediante el comando:

systemctl status suricata

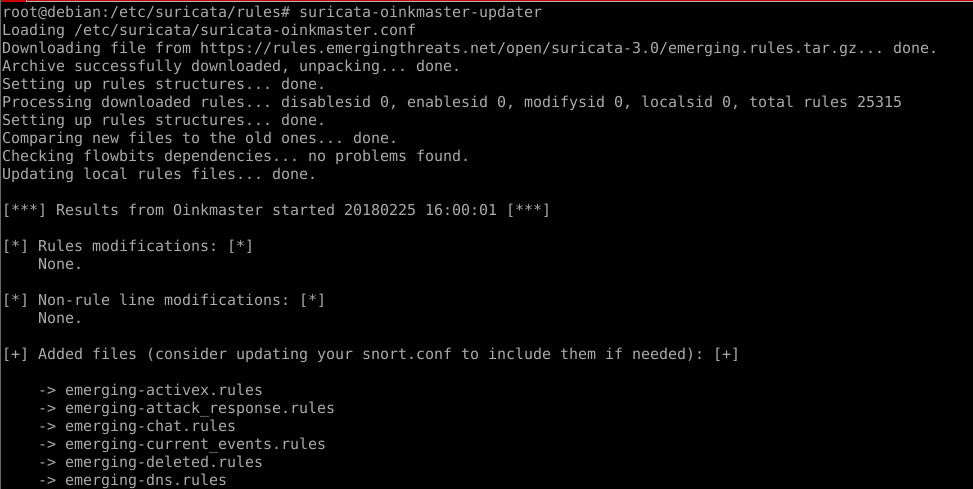
Comprobación del estado del servicio Suricata

Ahora deberemos realizar un par de configuraciones para que todo funcione correctamente. De forma predeterminada, Suricata utiliza el modo de trabajo *AF-PACKET* (lectura de sockets en *raw*) sobre la interfaz *eth0*. Sin embargo, en las últimas versiones de Debian, esta interfaz no existe; ahora, la interfaz de red física predeterminada es *ens33*. Por supuesto, si la interfaz que queremos inspeccionar es otra, será ésa la que configuraremos.

Dentro del fichero de configuración localizado en */etc/suricata/suricata.yaml*, debemos buscar el apartado de configuración del modo *af-packet* y cambiar la interfaz de la siguiente forma:

Edición de suricata.yaml para establecer la interfaz de escucha

¡Y ya tenemos nuestro servicio configurado y corriendo! Pero esto de poco nos va a servir si no sabe qué es lo que tiene que buscar. Para ello, ejecutaremos la herramienta *suricata-oinkmaster-updater* que se incluye con Suricata (al menos en Debian). Lo que hace es descargar las últimas reglas [publicadas](https://rules.emergingthreats.net/) por *Emerging Threats* e instalarlas en nuestro sistema. Nos servirán para tener ya unas cuantas reglas básicas instaladas, que exploraremos en las siguientes entradas

Ejecución de suricata-oinkmaster-updater (parcial)

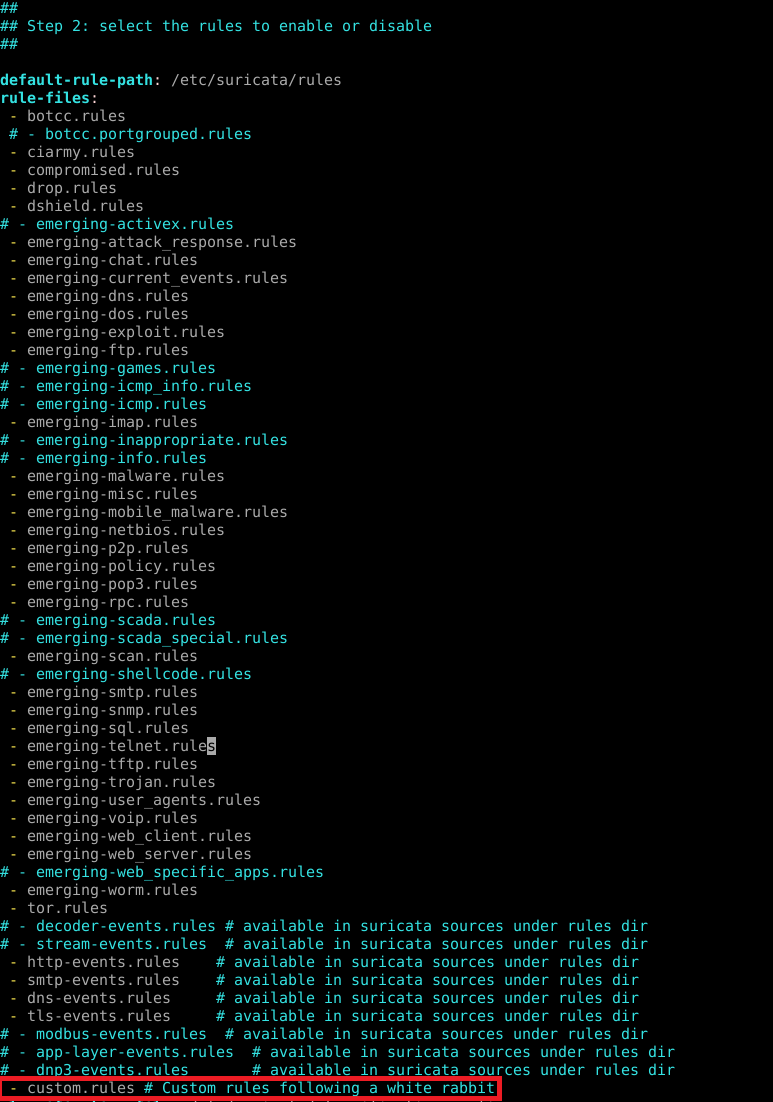
En las siguientes entradas recorreremos las reglas con mayor atención. Sin embargo, para empezar a toquetear, ¡vamos a crear nuestra primera regla! Para ello, crearemos y editaremos con nuestro editor de textos favorito (siempre vim :P) un fichero */etc/suricata/rules/custom.rules* con la siguiente línea de contenido:

alert icmp any any -> any any (msg: "ICMP detected";)

Creación de nuestra primera regla en Suricata

¿Qué es lo que hace, a grandes rasgos, esta regla? Básicamente, sacará una alerta (visible en los logs) cada vez que detecte un paquete ICMP. ¿Ruidoso? ¡Mucho! Pero nos sirve para realizar nuestras primeras pruebas de forma fácil y rápida Además, esto puede llegar a tener su sentido en ciertos entornos donde este tipo de tráfico está restringido (y por tanto, si aparece, ¡algo raro ha pasado!).

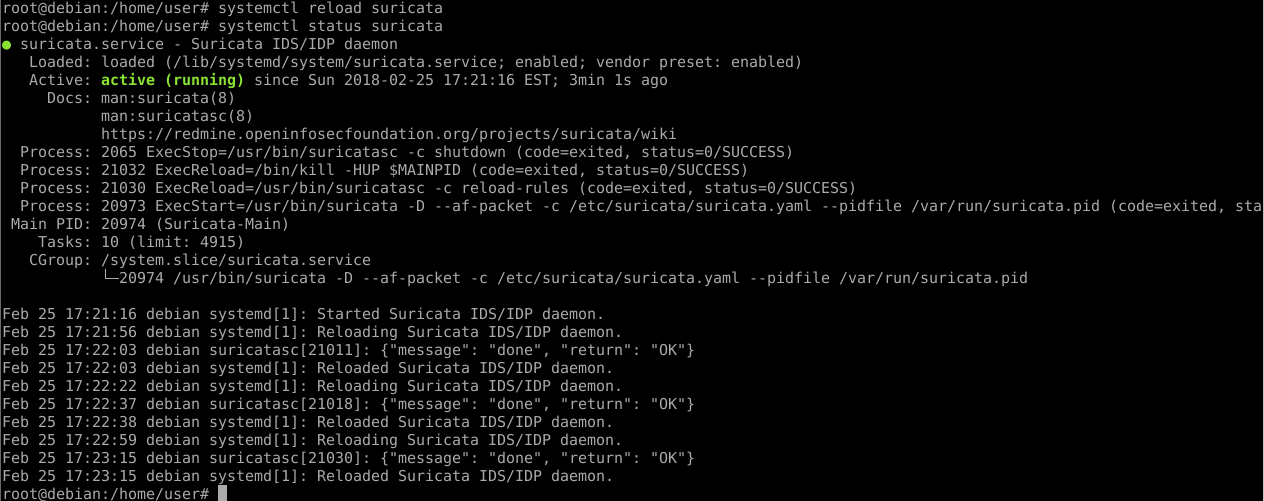
A continuación, debemos agregar nuestro fichero de reglas *custom.rules* a la configuración de Suricata para que pueda aplicarlas sobre el tráfico escuchado. Debemos agregarla al fichero */etc/suricata/suricata.yaml* de la siguiente manera, a continuación de las reglas que aparecen agregadas anteriormente mediante *suricata-oinkmaster-updater*:

Agregando nuestras reglas junto a Emerging Rules en la configuración de Suricata

Hecho esto, debemos recargar el servicio (podría tardar más o menos) para que recoja las nuevas reglas agregadas, y comprobar que todo ha ido bien:

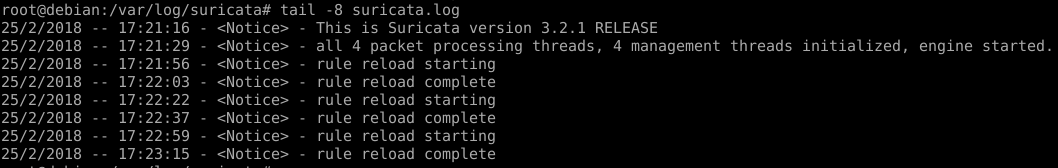
sudo systemctl reload suricata

sudo systemctl status suricata

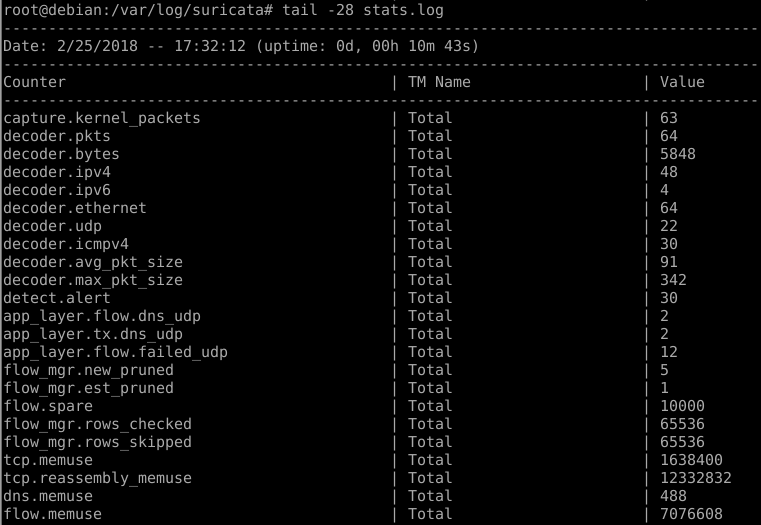
Recarga del servicio tras agregar las reglas

Con esto, y si todo ha ido bien, ¡ya tenemos nuestras reglas funcionando! Podemos hacer una pequeña prueba para comprobarlo. Pero antes, vamos a  realizar un muy breve recorrido por los logs de Suricata, localizados en */var/log/suricata*:

* *suricata.log*: Recoge los eventos del mismo Suricata: inicializaciones, recargas, errores…

Muestra de suricata.log.

* *stats.log*: Recoge estadísticas regulares acerca del tráfico que se ha ido analizando hasta el momento.

Muestra de stats.log.

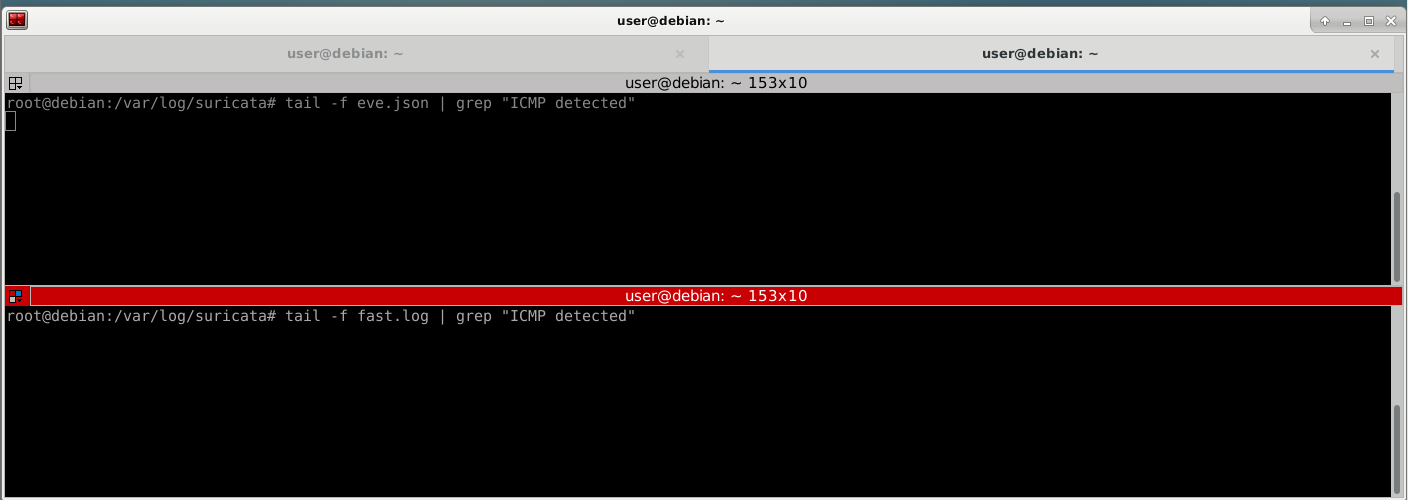
* *fast.log*: Recoge los eventos disparados por las reglas. Tiene como objetivo dar una impresión rápida y directa de los eventos.
* *eve.json*: Recoge, igual que el anterior, los eventos disparados por las reglas, pero lo hace en formato *JSON*, lo que permite que posteriormente pueda ser interpretado de forma mucho más fácil por programas externos.

Enseguida veremos muestras de estos dos Para lograr unos pocos eventos, en una terminal ejecutaremos un *ping* a 8.8.8.8 (los DNS de Google, por poner algo). Si hemos hecho todo bien (si no, tómate un café y revísalo todo, invita @CiberPoliES) deberían entrar eventos tanto a *fast.log* como a *eve.json*. Podemos comprobarlo en directo con un comando *tail -f* en ambos ficheros (que va mostrando las líneas nuevas). Adicionalmente, en ambos utilizaremos *grep* para que solamente muestre aquellas líneas que contengan *«ICMP detected»*, es decir, las que saca nuestra regla.

Por tanto, en dos consolas diferentes:

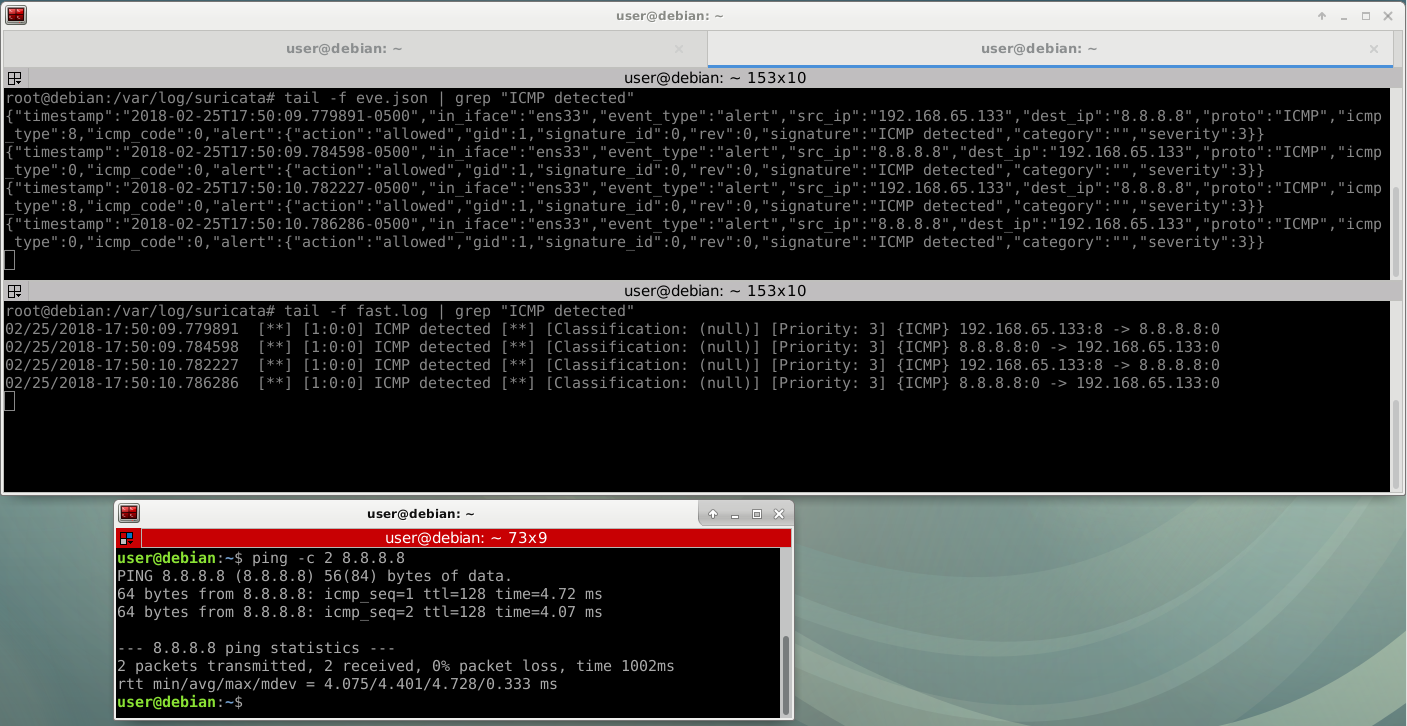
tail -f eve.json | grep "ICMP detected"

tail -f fast.log | grep "ICMP  detected"

Preparando la prueba de generación y visualización de logs.

Cruza los dedos, y en otra terminal ejecuta lo siguiente para lanzar dos *pings*:

ping -c 2 8.8.8.8

Éxito en la prueba de generación y visualización de logs.

¡¡Bingo!! Vemos que correctamente se recogen y visualizan los eventos en ambos logs, en su formato correspondiente y con el mensaje que indicamos: *ICMP detected*. ¡Todo ha salido bien!