Denial of Service

Transcribed on August 1, 2025 at 4:50 PM by Minutes AI

Speaker 1 (00:02)

Bienvenidos a esta nueva sesión, en esta sesión vamos a continuar con la parte práctica de Snort, viendo algún tipo más avanzado de ejercicio y también veremos la gestión de los registros o los logs.

Bien, ahora vamos a simular un ataque de denegación de servicio, es decir, vamos a hacerle muchas peticiones al servidor para ver si realmente es capaz de aguantar todo ese tipo de peticiones, entonces para eso utilizaré una aplicación que se llama HP que lo que hace es simular justamente eso, pues para comprobarlo primero, bueno aquí ya tenemos en el servidor Snort que lo dejaremos ahora funcionando, pero vuelvo a irme otra vez a la máquina atacante que es esta en mi caso que es la máquina 2 y aquí vamos a ver que tiene la IP 17, quedaros con eso, tengo la IP 5 en el servidor y la IP que acaba en 17 en la otra máquina, pero bueno ya vosotros según cómo lo montéis podéis cambiarlo.

Bien pues voy a instalar esta aplicación que se llama HP, mucho cuidado con esta aplicación porque realmente lo que hace es inundar de peticiones a una máquina y eso puede ser un problema, esto hacerlo siempre en un entorno seguro como lo hago yo ahora mismo entre dos máquinas virtuales, en concreto voy a utilizar HPIN 3 y para instalarlo estamos en la máquina atacante y hacemos un sudo apt installado siempre y ponemos hp Bien pues para instalarlo como siempre hacemos sudo apt install h decimos que OK, nuestra clave, lo instalamos, Bien ya lo tenemos, vamos a levantar otra vez el snorkel que hemos parado antes, Bien para levantarlo pues es como antes con la consola porque queremos ver realmente el resultado, lo normal es que no se mostrara y se almacenara en un fichero de configuración que es lo que está pasando por detrás, pero realmente ahora quiero que veáis en tiempo real cómo va apareciendo el ataque 2 que vamos a hacer ahora mismo.

Bien pues vamos a lanzar Snort para que se quede esperando, de momento veis, no hay ninguna actividad con lo cual no está pasando nada.

Bien, para ejecutar el ataque con HP hacemos un sudo, le metemos el parámetro c, que esto establece el número de paquetes ICMP que vamos a enviar, en este caso 5, le ponemos ahora guión i y un parámetro que es 10.000, esto lo que estamos haciendo es establecer el intervalo entre paquetes en microsegundos.

En este caso se envía un paquete cada 10.000 microsegundos o 10 milisegundos.

Después ponemos 1 y la dirección IP de ataque 10.

111.

El 1 especifica que se deben enviar un paquete ICMP de tipo 8, que es un PIN.

Hay que ver la documentación y todo, pero bueno, ya os lo cuento.

Aquí esto es un ataque muy sencillo de denegación de servicio.

Veremos ahora si esto lo detecta nuestra máquina con Ubuntu, que está en el lado del servidor.

Lanzamos el ataque y si vamos al servidor, veis que lo ha detectado.

Como podéis ver aquí arriba, está diciendo que ha habido un intento.

Aquí lo podéis ver desde la dirección IP 17.

Ha habido cinco.

Pues uno, dos, tres, cuatro y cinco.

Justo los cinco ataques que ha lanzado el HP 3.

Entonces todo lo que ha hecho es enviar paquetes ICMP a la dirección IP objetivo.

Cuidado, insisto, esto puede irse de madre si utilizáis muchos paquetes.

Muchos paquetes, mucho tiempo también mucho.

Ampliar los tiempos, bajar los tiempos.

Hagáis lo que hagáis, Esto es una herramienta muy ofensiva, pero nos viene perfecta para ver cómo Snort es capaz de detectarlo.

Entonces, partiendo de aquí, podemos ya actuar.

En este caso solamente está haciendo alertas, pero podemos actuar en caso de un ataque de denegación de servicio.

Podríamos, por ejemplo, activar una arquitectura de backup de refuerzo, para que no caiga, para que sigamos dando un servicio, por ejemplo.

Y eso, Snor, al ser a tan bajo nivel y funcionar en Linux, podríamos interconectarlo con diferentes scripts y con lo que sea, sin ningún tipo de problema.

Bien, dejamos así el servidor y ahora vamos otra vez a la máquina atacante.

Bien, pues de nuevo aquí en la máquina atacante, hacemos un nmap p, que ya sabemos lo que hacía, y le lanzamos a la IP 10.211.55.5.

Esperamos.

Ha escaneado, pero lo que nos interesa no es el resultado en este caso, lo que nos interesa es ver si Snor ha detectado algo.

Así que vámonos otra vez a la máquina servidor.

Y vemos que sí, porque hasta aquí era el ataque del HPIN 3.

Pero a partir de aquí vemos otra cosa.

Fijaros, justo en este punto, aquí abajo, sigue siendo el atackstream, nos dice que hay un intento de ataque de denegación de servicio.

Esto es la parte del HPIN 3, que ya lo ha analizado y después ya vemos la parte del nmap, veis que ha habido un agente de petición, bueno todo esto claro, aquí no te pone nmap directamente sino te está diciendo lo que está usando nmap para intentar sacar información de la máquina, con lo cual vemos que lo ha detectado perfectamente todos los intentos de la máquina desde el nmap.

Bien, paramos de momento nuestro servidor y ahora lo que vamos a intentar hacer es crear nuestras propias reglas.

Bien, pues el paso principal es crear un fichero de texto y en él poner nuestra regla.

Las reglas propias de snor tienen que crearse a partir del número 999999, me refiero al sid, tiene que ser partiendo de ese número porque de ahí para abajo son todas las que ya vienen por defecto, están reservadas las reglas, recordad que están en la carpeta etc snot rules Si hacemos por ejemplo, vamos a ver lo que hay en la carpeta nor, veis, aquí están todos los ficheros con reglas, como podéis ver hay absolutamente de todo, hay reglas ya creadas para virus, ataques relacionados con la web, de servicios concretos, pp, aquí hay muchísimos, tenéis que documentar bien, pero todas estas ya se ponen por defecto en activo cuando tú arrancas snor, porque esta es la carpeta que se define en el fichero configurado como la que tiene que utilizar para las reglas.

Entonces aquí sería donde crearíamos nuestra propia regla, es muy difícil crear una regla que no esté aquí, por no decir prácticamente imposible, tendríamos que hacer algo muy personalizado, concatenando eventos y cosas, así que lo que voy a hacer es simular una conexión ssh pero controlada por nosotros con un mensaje propio, para que veáis cómo podemos llegar incluso a manejar los diferentes flags para ver qué queremos detectar de ese intento de intrusión por ssh.

Así que lo primero será crear un fichero en esa carpeta con la extensión rules.

Bien, pues lo que haremos será, vamos a hacer un sudo nano, le ponemos fc snorles y le vamos a llamar en este caso por ejemplo ssh new rules Ya lo tenemos aquí y ahora simplemente había que montar ese formato que habéis visto en la diapositiva que antes os he contado, en este caso voy a poner la cadena entera y ahora os explico qué es cada cosa

Entonces lo que voy a hacer es un alert, bueno ya sabéis que alert es la acción que va a tomar, en este caso es alertar el protocolo, por supuesto este cp aquí ya sabéis que es el flujo, cualquier puerto hacia, desde Internet, cualquier puerto hacia nuestro equipo, que es el home net, acordáis que era el que busca, pusimos como ip destino, pero esta vez como queremos hacerlo contra ssh no pondremos Any, ponemos 22 porque queremos monitorizar solo los intentos de conexión con ssh y ya ahora vendría la cadena más compleja que es la que se monta toda la reacción y todo lo que tiene que mirar.

Lo primero es el mensaje, aquí en el mensaje le pondríamos por ejemplo, lo pondré en español para que podamos diferenciarlo de las reglas que vienen por defecto, así que puedo poner ojo, prueba de intento ssh por ejemplo, por poner algo y ahora aquí pues ponemos punto y coma y hacemos este comando flow stateless ¿Qué hace el flow stateless?

Lo que hace es, tú puedes mirar si hay un establish o un stateless, un establish es que se ha establecido una conexión y stale son sesiones que no se han producido aún, que es justo este caso, en este caso no ha habido una conexión, estamos en proceso d, con lo cual no es un establish, no se ha establecido, con lo cual tenemos que poner en el flow stateless.

Hay más, tenemos toClient, toServer, from client, hay varias para poder activar aquí, pero en este caso es stateless.

Para esto tenéis que repasar un poco cómo es la cabecera tcp de un paquete de datos y ahí veréis todos los flag y todo lo que podemos activar aquí a este nivel.

Por ese motivo es importante conocer muy bien la arquitectura que tienen los paquetes que estamos analizando.

Bien, lo siguiente es poner después del punto y coma sería poner el valor flags as, en este caso el flag s quiere decir que lo que está haciendo es buscar paquetes con la s, que significa sin flag sin.

Bien, flag s lo que indica es que utilizamos el syn, como he dicho antes, que significa synchronize o sincronización que se establece en uno para indicar una conexión tcp y el indica que el bit sync debe estar activo, lo que significa que se está iniciando una conexión tcp.

Entonces cuando veas el flag s implica que la regla se va a activar si se detecta un paquete tcp que tenga el bitcoin sin establecido, lo que indica un inicio de conexión.

Bien, pues el siguiente paso ya es mucho más directo porque ya sabéis que después lo que hay que poner es el SID, que ya os dije que tenía que ser mayor de 999.999, en este caso le pondremos pues 1 1 2 3, haremos un número grande para que no haya problema, ponemos tin y después el número de televisión, que como es nuestra pues le pondremos el que queramos, así que le ponemos el 3, ya con esto terminamos.

Pues bien, esto sería una regla que lo que hace es detectar conexiones SSH.

Bien, el siguiente paso va a ser añadir esta regla nueva dentro del fichero de configuración para que sepa snorke que tiene una nueva regla que comprobar.

Entonces grabamos y ahora hacemos un sudo nano y nos vamos otra vez a etc snort, esta vez buscamos el fichero snort conf, lo editamos y buscamos una etiqueta que dice site specific rules, que es donde se especifican cuáles son las reglas específicas que vamos a utilizar, ahí tenemos que poner el nombre del fichero que tienen nuestras reglas nuevas o también poner la ruta por defecto que ya sabéis que es la de rules, así que bueno, buscamos esa carpeta o también poner la ruta por defecto que ya sabéis que es la de rules, así que bueno, buscamos esa carpeta.

Bien, aquí lo tenemos referida a etiqueta, buscamos esa etiqueta que es el site specific rule y aquí tenemos que incluir la nueva ruta o el fichero que acabamos de crear nuevo, para eso podemos utilizar ya aquí por defecto coge local rules, pero podemos copiar esto mismo site specific rule y aquí tenemos que incluir la nueva ruta o el fichero que acabamos de crear nuevo, para eso podemos utilizar ya aquí por defecto coge local rules, pero podemos copiar esto mismo, lo copiamos, lo pegamos y le cambiamos el nombre de localrules a la que pusimos que llame ssh new rules.

Bien, con esto ya hemos añadido esta nueva regla que no tenía controlada, en este caso es normal, en este caso fijaros que las específicas son las local rules, pero le hemos dicho vale pues también añade este de aquí, si quisiéramos que cogiera algún otro pues tenemos que ir añadiendo aquí los diferentes ficheros de reglas.

Almacenamos, salimos y ahora levantamos otra vez snort como ya hemos hecho antes.

Bien, pues ya tenemos snor esperando algún tipo de conexión SSH Así que lanzaremos un ssh a la 102.11.155 y lo dejamos aquí y vamos a ver qué es lo que está diciendo el servidor.

Bien pues perfecto, aquí lo veis y de hecho fijaros, poned el texto que pusimos nosotros para confirmar que esta regla es nuestra y directamente está saltando, está diciendo, nos está avisando de un intento de conexión SSH hacia nuestra máquina desde la ip al puerto 22 acabando en la 5 que es la de nuestro servidor.

Como veis es muy sencillo crear tu propia regla y añadirla y aquí viene el potencial porque si llegamos a dominar todo lo que te ofrece de parámetros, de flags, toda esa cantidad de información, podéis automatizarlo al nivel que queráis, lo podéis personalizar a un nivel espectacular.

Un tema importantísimo de Snort es ver los logs del sistema, bueno en Snort y cualquier cosa de ciberseguridad, pero en Snort es particularmente importante, importante tener localizado y ubicado los logs del sistema.

Entonces para eso los logs que vienen con snor están en la carpeta var log snort, vamos a ir a ella, vamos a ir a la carpeta var log snort Bien pues dentro de esta carpeta si hacemos un quingen vemos los diferentes ficheros log, el principal es snort log, este de aquí que es el que tiene todos los eventos.

Si yo por ejemplo intentara hacer un CAD a Snort, fijaos que es binario y no vería el contenido.

Para ver el contenido hay una pequeña aplicación, hay una pequeña herramienta con snorkel que ya viene incluido cuando instalamos snorkel que se llama u spew foo y ahora solamente pondríamos el nombre snor log y esto ya convertiría a un formato legible todo lo que son los paquetes y ya sólo nos quedaría pues analizarlo en el que ya vemos hasta nivel de paquete que es lo que hemos recibido, fijaros qué nivel de detalle, fijaros aquí por ejemplo en este está todo el evento y el paquete totalmente detallado, qué tipo de flag, protocolos, el contenido del paquete, etc.

Con lo cual como veis en la parte de la gestión de log es muy minucioso porque es fundamental, porque todos estos logs son la clave del éxito de la detección de algún tipo de amenazas por parte de Snort.

Pues Snort destaca porque es un sistema altamente configurable, tanto a nivel de reglas como a nivel de actuación, por ejemplo, con las alertas con ejecución, tenemos diferentes formas de actuar ante alguna posible amenaza y también tiene un control de registro o D log muy exhaustivo que facilita un análisis total y un seguimiento de todo tipo de eventos de seguridad.

En un timeline llegamos.