UNIVERSIDAD DE GRANADA

Honeypots: una herramienta para conocer al enemigo.

Administración de Sistemas y Seguridad

Máster Profesional en Ingeniería Informática

Curso académico 2022/2023

Autores

José Alberto Gómez García

Lucas Gutiérrez Durán

Contenidos

1. Introducción a la ciberseguridad en la red

Una red es cualquier conjunto de computadores, smartphones, impresoras y, en definitiva, dispositivos electrónicos conectados entre sí para poder compartir datos e información, recursos o servicios entre ellos utilizando unos protocolos de comunicación comunes. Esta comunicación se puede realizar tanto a través de medios físicos como de dispositivos inalámbricos y está compuesta, desde una perspectiva muy general, de un emisor, un mensaje y un receptor.

Cuando hablamos de redes, las primeras en aparecer son de telefonía y telegrafía; más adelante aparecería, a finales de la década de los sesenta, la primera red de computadores como tal: ARPANET. Aunque hoy en día pueda parecer imposible e incluso irónico, en un principio no se pensaba en el concepto de seguridad como un criterio importante dentro del ámbito de las redes de computadores. No obstante, si bien cualquier sistema informático o de comunicaciones puede sufrir un ataque, siempre es más sencillo que esto ocurra si ese sistema está conectado remotamente a otros (es más sencillo encontrar una brecha de seguridad en una conexión de red a copiar el disco duro de un ordenador a un disco duro externo sin que el propietario se dé cuenta). Por esta razón, no es hasta que se produce un auge en el campo de las redes cuando se producen los primeros hackeos.

Como se ha dicho, la seguridad no era un tema primordial y esto queda patente en lo que se considera el primer ataque a una red, con las manipulaciones de la telefonía en EE.UU. a principios de los años 70. John Draper consiguió modificar un sencillo silbato que venía de regalo en las cajas de cereales de la marca estadounidense “Cap’n Crunch” para engañar a la central telefónica y poder hacer llamadas gratis. Esto remarca los bajos niveles de seguridad existentes por aquel entonces.

A partir de ahí, con la expansión de todo tipo de comunicaciones entre sistemas informáticos, el mundo del hackeo también se expande y evoluciona. Así, con el nacimiento de empresas como Microsoft y Apple y la popularización de Internet empiezan a surgir los primeros ataques importantes y los aspectos relativos a la ciberseguridad se empiezan a tornar más cruciales. Con todo ello, se han terminado estableciendo unos términos esenciales en la seguridad de las conexiones de red.

- **Confidencialidad**. El mensaje llega únicamente al destino especificado, sin que otros usuarios no autorizados lo intercepten por el camino, es decir, que el mensaje sólo es comprensible por las entidades autorizadas.
- **Integridad**. El mensaje no es modificado en el camino del origen al destino. Si se produjese alguna modificación, se debe detectar para poder solucionarla lo más rápido posible.
- **Autenticación y control de accesos**. Identificar los usuarios autorizados y solo permitirle el acceso a el servicio a ellos.
- **No repudio.** Un extremo de la comunicación no puede retractarse después de haber hecho el envío de información, esto es, impedir la renuncia de la autoría de una acción.

- **Disponibilidad.** Capacidad del sistema de mantener las prestaciones independientemente de la demanda.

La seguridad en la red es, por tanto, el conjunto de políticas que rigen el funcionamiento de ésta y cuyo objetivo es asegurar que todos los aspectos y propiedades anteriores se puedan cumplir sin brechas durante la actividad del servicio. No sólo se ha de pretender proteger la conexión en sí misma, sino todos los datos asociados a los integrantes de la red.

Hay que tener en cuenta, además, que no todos los ataques son activos (ataque directo a los recursos, datos, información o servicios del sistema) sino que también existen los ataques pasivos. Con el primer tipo de ataques es con el que más familiarizados estamos: incluye virus, escuchas, suplantación de DNS, man-in-the-middle, entre muchísimos otros. El segundo tipo de ataques es quizá del que menos nos damos cuenta: escaneo de puertos, sniffing, etc.

Se podría pensar que estos ataques son del todo imprevisibles, pero el hecho es que no lo son (al menos en parte) ya que los cibercriminales se van adaptando conforme se van actualizando las medidas de seguridad. En 2022, muchas empresas fueron víctimas de ciberataques y brechas de datos. Se notificaron mediante la AEPD (Agencia Española de Protección de Datos) y los formularios que tiene habilitados para dicho fin, 1647 brechas de seguridad. Un ejemplo de estas brechas de seguridad, son los ataques por ransomware sufridos por los hospitales del Vall d'Hebron y Lucena, en los que se robaron (y quedaron expuestos) datos clínicos de miles de usuarios. Los perpetradores del ataque intentaron chantajear a los hospitales. Fuera del territorio nacional, tanto NVIDIA como Samsung sufrieron un ataque también por ransomware. En el caso de la primera empresa se filtró información patentada y código fuente relativos a la tecnología DLSS; mientras que en el caso de Samsung se filtró parte del código fuente de los smartphones de la saga Galaxy. Curiosamente, estos dos ataques fueron perpetrados por el mismo grupo, Lapsus$.

Como se puede imaginar, actualmente no existe ningún sistema que esté totalmente blindado contra ataques. Por ello, para proteger los sistemas se implementan una serie de medidas de seguridad que suelen variar en función del uso que tengan los dispositivos. Por ejemplo, la mayoría de los ordenadores implementan un cortafuegos y cuentan con un antivirus para detectar y encargarse de amenazas potenciales, pero la naturaleza del cortafuegos o el antivirus varía dependiendo del sistema para el que ha sido diseñado: no será el mismo en un PC de uso diario que en un servidor.

Con todo esto, surgen numerosas soluciones de seguridad de redes, muchas para evitar que el atacante consiga entrar al sistema, muchas para solucionar las brechas una vez el atacante está dentro del sistema y muchas con el fin de prevenir ataques al sistema, fortaleciéndolo y haciéndolo menos vulnerable (y por tanto menos atractivo para el atacante y menos propenso a sufrir un hackeo). Nosotros nos vamos a centrar en una de las estrategias más potentes de prevención: los honeypots.

2. Breve historia de los honeypots

La historia de los honeypots se remonta a finales de la década de 1980 y principios de la década de 1990. Fue en este período cuando se comenzó a tomar conciencia de la creciente amenaza de los ataques cibernéticos y se buscaban nuevas formas de defenderse. En ese momento, Clifford Stoll, un administrador de sistemas en el laboratorio nacional de física Lawrence Berkeley en California, fue uno de los primeros en utilizar un honeypot para rastrear a un hacker.

En 1988, Stoll, intrigado por una serie de intrusiones en sus sistemas, decidió crear un señuelo atractivo que simulara una máquina militar altamente valiosa. Configuró un sistema llamado "The Cuckoo's Egg" (El huevo del cuco) que aparentaba ser vulnerable, pero que en realidad estaba cuidadosamente monitoreado. Con esta trampa, logró rastrear y seguir las actividades de un hacker que resultó ser un espía informático patrocinado por una agencia de inteligencia extranjera. El incidente y su monitorización duró cerca de 3 años.

Este evento marcó el inicio del uso de honeypots como una estrategia de seguridad. A medida que la conciencia sobre los ataques cibernéticos aumentaba, más investigadores y profesionales de seguridad comenzaron a desarrollar honeypots para estudiar y comprender mejor las técnicas de los atacantes. Estos primeros honeypots a menudo se implementaban en sistemas Unix y se centraban en el monitoreo de actividades sospechosas en los registros de red. Para ello, fue vital la figura de Fred Cohen, que en 1997 publicó uno de los precursores de los actuales honeypots de baja interacción (que trataremos más adelante), el “Deception Toolkit” (DTK)

En 1998 se lanzaron los dos primeros honeypots comerciales, llamados “Cybercop Sting” y “NetFacade”. El primero corría bajo Windows NT y podía simular diferentes tipos de dispositivos, mientras que el segundo podía emular una red IP de clase C (254 equipos) y permitía simular equipos con 7 sistemas operativos y multitud de servicios.

En 1999, se lanzó el grupo sin ánimo de lucro “Honeypot Project”, dedicado a investigar la comunidad de hackers de sombrero negro. Se lanzaron otros softwares como “ManTrap” (1999), posteriormente renombrado a “Decoy Server”, “Tiny Honeypot” (2002) o Google Hack Honeypot (2002). Posteriormente, en 2004 se lanza una herramienta que permite implementar honeypots mediante CD-ROM booteable, “Roo”.

Con el tiempo, los honeypots evolucionaron para adaptarse a diferentes necesidades y escenarios de seguridad. Se crearon distintos tipos de honeypots, los cuales se tratarán más adelante en esta memoria. Su desarrollo y uso continuo son fundamentales para mantenerse al tanto de las últimas técnicas de ataque y salvaguardar la integridad de las redes y los sistemas informáticos.

3. Definición y características de los honeypots

Los honeypot nacieron para proteger la información vital y son un sistema cuyo fin es el de actuar como un señuelo, para así atraer a los potenciales atacantes. Cuando un atacante penetra en el honeypot, éste activa una alerta que informa del ataque a los administradores, lo que permite que se pueda mitigar (hasta cierto punto) la intrusión: permiten la monitorización del sistema recopilando información sobre los ataques y permitiendo reconocer la procedencia y naturaleza de las posibles amenazas a las que nos enfrentamos. El honeypot es una parte aislada de nuestro sistema informático, un sistema falso, que realmente no desempeña ninguna función real más allá que la de aumentar la seguridad. De aquí en adelante nos referiremos al sistema que se quiere proteger como sistema real y al honeypot como señuelo, sistema falso o sistema ficticio.

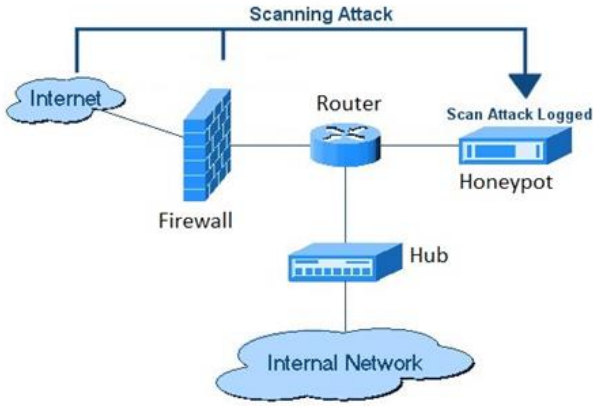
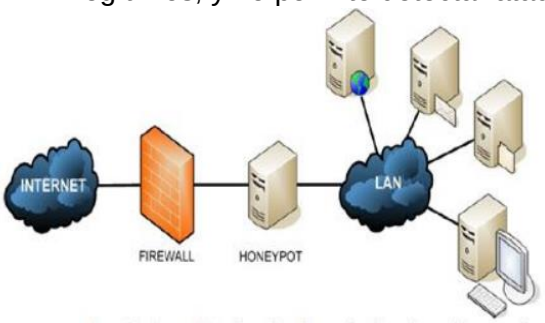


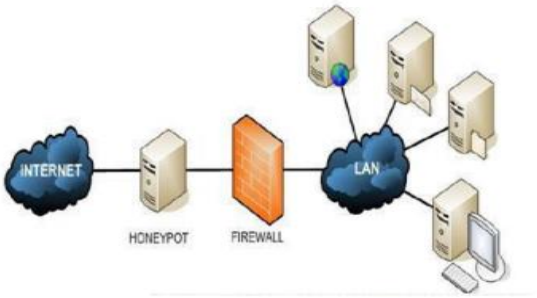
Imagen 1. Diagrama de un honeypot.

Los honeypots, por tanto, tienen varias funcionalidades: distraer al atacante del sistema real, recabar información sobre los ataques para poder proteger mejor el sistema real, alertar de un ataque cuando se produce y, una vez detectado un ataque, ralentizarlo, para conseguir implementar algún parche de seguridad en el sistema real y evitar que el atacante penetre en él. Gracias a la flexibilidad que ofrecen los honeypots, estas funcionalidades se pueden implementar individualmente en varios honeypots diferentes o se pueden combinar todas para aprovecharlas al máximo. Además, por la propia definición de honeypot, se puede considerar un ataque toda interacción con el sistema ficticio, ya que no hay ningún motivo para que algún otro usuario aparte de los administradores (que accederán a él para ver los datos de monitorización) interactúe con dicho sistema.

La instalación del honeypot se puede realizar en tres puntos diferentes de nuestro sistema real. Pueden instalarse:

- **En el interior del firewall**: en este caso, nos permite detectar ataques tanto internos como externos. Requiere una configuración específica para dejar acceso al honeypot, pero no a la red. De no configurarlo correctamente, pueden producirse filtraciones de tráfico.
- **En el lado externo del firewall**, para detectar ataques externos que todavía no han conseguido penetrar en nuestro sistema real. Hace que la seguridad de la red interna no se vea comprometida en ningún momento, pero conlleva dos desventajas; consume ancho de banda que podríamos emplear para dar servicio a los clientes legítimos, y no permite detectar ataques que tengan lugar desde dentro de la red.





Imágenes 2 y 3. Honeypot detrás y delante de un firewall, respectivamente.

- **En una zona desmilitarizada (DMZ)**. Así, será posible la separación del honeypot de la red interna y servidores. Nos permite detectar ataques tanto internos como externos, pero requiere modificar el firewall.

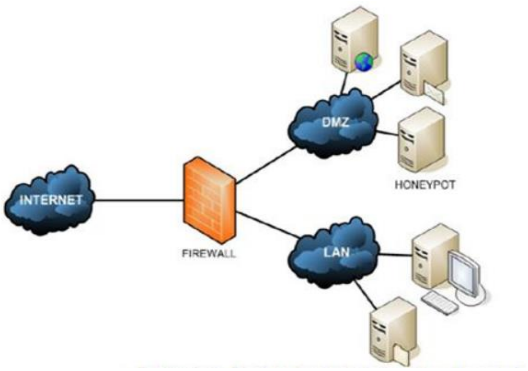


Imagen 4. Honeypot en zona desmilitarizada de la red.

Además, hay dos tipos de implementaciones: la física y la virtual. En la física dispondremos un computador específico y exclusivo para el honeypot, integrado en la red del sistema con una IP propia. En la virtual creamos un sistema virtualizado dentro de uno de los equipos físicos de los que ya disponemos en nuestro sistema real.

Además, cabe destacar que es posible ubicar varios honeypots (tanto físicos como virtuales, o combinando ambos tipos) en la red de nuestro sistema real y crear así lo que se conoce como honeynet. Esto nos permite monitorizar de una forma más completa y una parte más extensa de nuestro sistema, incluyendo diversos recursos que con un único honeypot no podríamos llegar a cubrir.

4. Tipos de honeypots

Una de las razones por las que los honeypots son una técnica de prevención tan importante es porque se pueden emplear para proteger todo tipo de recursos del sistema. Esto significa que podemos implementar honeypots con un diseño específico para cada tipo de amenaza a la que queremos que se enfrente. Vamos a destacar los principales y más comunes tipos de honeypots, identificándolos con la amenaza a la que se corresponden.

Trampas de correo electrónico (spamtraps). Se basa en direcciones de correo electrónico que no existen o que ya no están en uso. Hay tres tipos básicos de estas direcciones de correo ficticias.

- Las denominadas “puras”, creadas por organizaciones anti-spam con el único fin de detectar ataques de spam.
- Las que han expirado o han sido recicladas, alguna vez pertenecieron a una persona, pero ya no tienen propietario. A veces son rescatadas para utilizarlas para detectar ataques de spam.
- Las creadas por un error tipográfico. Por ejemplo, por haberse registrado en una web con un correo de dominio “gmai.com” en vez de “gmail.com”.

Este tipo de honeypots son del todo seguros, ya que no utilizan direcciones auténticas y permiten detectar e identificar de forma bastante sencilla a los atacantes. Una vez llega un correo a esta dirección, se añade el remitente a una lista negra (blacklist) y así se puede evitar que envíe correos a las direcciones auténticas de nuestro sistema.

Base de datos señuelo. Se utilizan para detectar ataques sobre bases de datos: se utiliza una que no contenga información sobre el sistema real. Los ataques a bases de datos incluyen inyección SQL, vulnerabilidades a nivel de validación, utilización indebida de los servicios ofrecidos, vulneración de privilegios, etc.

Señuelo de malware. Consiste en la creación de aplicaciones software o APIs que no tengan funciones ni conexión auténtica con el sistema real. El atacante, sin embargo, la detectará como una aplicación normal y cuando intente introducir algún tipo de malware en ella, éste se podrá monitorizar para ser analizado posteriormente. De este análisis se pueden obtener las características esenciales del malware para poder desarrollar software antimalware para implementar en las aplicaciones reales o corregir vulnerabilidades en las APIs del sistema.

Señuelo araña. Las arañas web son un tipo de procesos automatizados (generalmente bots) que se utilizan en la World Wide Web para rastrear masivamente páginas web a través de sus enlaces. Aunque las empresas que ofrecen servicios de búsqueda web utilizan esta técnica con fines “lícitos”, como indexar webs y optimizar los resultados de las búsquedas, otras organizaciones las utilizan con fines malintencionados: desde recopilar datos hasta obtener formas de contacto masivas. El honeypot se basa en la creación de páginas web (y sus respectivos enlaces) cuyo acceso no es abierto, a excepción de para las arañas web que sí que podrán acceder. Cuando se produce un acceso a esa web, se puede obtener información acerca del usuario que entró en ella y añadirlo a una blacklist de bots maliciosos y rastreadores web.

Aparte de ser específicos para el área que se quiere proteger, también puede variar el grado de interacción que los honeypots tienen con el atacante. Dentro de esta clasificación podemos encontrar dos grandes grupos:

Honeypots de baja interacción. Implementan servicios básicos de red y los protocolos TCP/IP, con lo que el contacto con el atacante es menor (al ofrecer servicios elementales es muy probable que el atacante se dé cuenta de que es un señuelo y abandone el ataque rápidamente). Por ello, la información recolectada es mínima, únicamente grado y tipo de amenaza y, posiblemente, la procedencia de ésta (de una forma poco detallada). La contraparte es que son los más seguros y sencillos de implementar en nuestro sistema.

Honeypots de alta interacción. Implementan servicios reales, imitando al sistema real, tienen una red local, bases de datos y procesos. El hecho de poseer todos estos servicios tiene la desventaja inherente de hacer que el honeypot sea, a su vez, una posible puerta de entrada al sistema real, por lo que la seguridad en este tipo de implementación es fundamental: el honeypot ha de estar aislado completamente de la red real del sistema. Además, consumirá más recursos y será más difícil de implementar y de supervisar. No obstante, las ventajas que ofrece respecto a los de baja interacción son bastante grandes. Al tener mayor número de servicios, los atacantes pasan más tiempo dentro del sistema ficticio y es más difícil que se den cuenta de que es un señuelo. Esto hace que se pueda obtener mucha más información, no solamente la naturaleza del ataque y el tipo de amenaza, sino también sobre la procedencia de una forma más detallada, ya que se pueden implementar herramientas de rastreo dentro del propio honeypot.

Honeypots puros. El honeypot no es parte de otro sistema real con el fin de mejorar la seguridad de este, sino que es un sistema real completo independiente de cualquier otro. Este es el tipo que más información proporciona, ya que tiene las funcionalidades plenas de un sistema normal, pero también es el que más cuesta implementar, mantener y supervisar (ya que se debe disponer de otro sistema distinto que esté completamente protegido con el que se pueda acceder de forma limpia al honeypot). Este tipo es el menos común y solo lo utilizan organizaciones de ciberseguridad o investigación cuyo fin es detectar y analizar las tendencias de los atacantes.

Más allá del propio diseño, los honeypots también se pueden clasificar dependiendo de la intencionalidad y el fin con el que se implementan.

Production honeypots. Son aquellos utilizados para detectar amenazas y protegerse frente a ellas. Son implementados principalmente por empresas con el fin de reducir los riesgos de sus sistemas. Normalmente este tipo de honeypots son de baja interacción, debido a la sencillez de su implementación.

Research honeypots. Son utilizados con el propósito de obtener información sobre las motivaciones de los ataques y la psicología de los atacantes. Son usados por organizaciones sin ánimo de lucro, militares o gubernamentales con el fin de investigar y desarrollar técnicas más potentes e innovadoras de protección.

5. Ventajas y desventajas del uso de honeypots

Como hemos visto, los honeypots se basan en un concepto simple, pero la complejidad de su diseño e implementación puede variar dependiendo del tipo y clase de señuelo que queramos utilizar. Aunque anteriormente se han visto varios tipos de clasificaciones, nos gustaría destacar sobre todo las ventajas e inconvenientes de la categorización en función del grado de interacción con el atacante: honeypots de baja y alta interacción.

Honeypots de baja interacción. Son relativamente sencillos de implementar, fáciles de usar y casi no presentan riesgos para nuestro sistema real cuando el señuelo sufre un ataque. No obstante, tienen claras desventajas como que la información que se obtiene de su uso es muy limitada y, además, son fácilmente detectables por los atacantes.

Honeypots de alta interacción. Permite obtener mucha más información, que servirá para conocer cómo el atacante accedió al sistema y las acciones que llevó a cabo en él. Sin embargo, también tiene desventajas: existe un mayor compromiso y riesgo del sistema real, ya que el señuelo tiene parte de (o incluso todos) los servicios que los demás computadores de la red ofrecen.

Dejando ahora a un lado los beneficios y perjuicios de los diferentes tipos de honeypots, también nos gustaría poder destacar las virtudes e inconvenientes del uso de los honeypots.

Una de las utilidades fundamentales de los honeypots es poder monitorizar un sistema (el propio señuelo) que recibe ataques que sabemos de antemano que no van a conllevar ningún riesgo al sistema real. Esto implica que durante los ataques podemos generar una serie de registros que posteriormente se podrán analizar con el fin de prevenir y combatir las amenazas en el sistema real. Esto supone una gran ventaja en seguridad, ya que nos permite ir por delante de los ciberdelincuentes que intenten atacar nuestro sistema real, siempre que contemos con un personal experto en la materia, capaz de extraer toda la información que dichos registros pueden proporcionarnos.

Otra ventaja importante de los señuelos es que pueden servir como una herramienta de pentesting. El pentesting (o prueba de penetración) es una técnica que permite hacer un examen del sistema con el fin de descubrir los puntos débiles de seguridad en el mismo. Si disponemos de un honeypot de alta interacción o un honeypot puro, llevando a cabo este tipo de pruebas en él podremos exponer sus puntos débiles, que serán parecidos a los del sistema real (dada la similitud entre estos tipos de señuelos y el sistema real) y, por tanto, podremos implementar mejoras en el sistema real para mejorar su rendimiento en términos de seguridad.

Un problema de los honeypots es que sólo detecta una intrusión si el atacante interactúa con él. Por tanto, no nos avisan ni nos sirven de protección contra ataques a otros equipos de la red, ya que el señuelo ni tiene accesos a esos computadores ni será capaz de detectar estos ataques para monitorizarlos.

Además, como el objetivo fundamental de los honeypot es atraer los ataques, el señuelo se debe parecer lo máximo posible a un equipo real. Si esto no se consigue, el atacante evitará atacar al honeypot (porque se dará cuenta de que no es un computador real del sistema) y éste perderá toda su utilidad. Esto deriva una nueva desventaja: como el señuelo debe imitar al sistema real, es muy importante que esté correctamente configurado porque, de lo contrario, será una amenaza para la seguridad de la red. Se pretende que sea el principal objetivo de los ataques, por ello, si no está bien protegido y aislado del sistema real podrá ser una puerta de entrada a la red, dejándola expuesta.

6. Aplicaciones y herramientas

En cuanto a herramientas que permitan la creación de honeypots, la oferta es enorme. Además, hay que destacar que hay varios tipos de herramientas y dos formas principales de generar un honeypot.

Creación "desde cero". Podemos crear un honeypot por nuestra cuenta, sin software de terceros específico para ello. Seremos nosotros los encargados de tomar una máquina, física o virtual, instalar y configurar el sistema operativo y firewall. Con esto ya tendríamos el honeypot "base" creado; ahora falta añadirle herramientas de monitorización y detección de amenazas.

Entre éstas podemos encontrar también mucha variedad, desde algunas básicas y simples que nos permiten detectar intentos de inicio de sesión, como ELMAH, hasta sistemas de detección y prevención de intrusión en la red como Snort. Si hemos decidido crear un honeypot por nuestra cuenta, será de vital importancia hacer varias pruebas para comprobar que el señuelo funciona correctamente y tiene suficiente seguridad antes de ponerlo en funcionamiento dentro del sistema real.

Creación a partir de herramientas de terceros. En Internet existen multitud de aplicaciones de terceros que tienen implementados muy diversos honeypots y que podemos instalar en nuestro sistema para hacer uso de ellos. Podemos destacar varios:

- KFSensor es un sistema de honeypots con IDS (sistema de detección de intrusión) para sistemas operativos Windows que incluye, entre otras facilidades, herramientas de control remoto con interfaz gráfica.
- Glastopf permite crear honeypots de baja interacción para aplicaciones web, permitiendo la detección de spam e inyección SQL, entre otras amenazas.
- T-pot, que reúne otras tantas herramientas de honeypots (incluye, por ejemplo, Glastopf) y permite desplegar hasta 11 tipos diferentes de honeypots y monitorizarlos todos cómodamente con una interfaz muy detallada, es quizá la herramienta más completa en este ámbito.
- Cowrie es un honeypot de alta interacción para SSH y Telnet, diseñado para registrar los ataques de fuerza bruta realizados por el atacante. Dispone de un modo en que emula a un sistema UNIX en Python, y actúa como proxy de SSH y Telnet, permitiendo observar el comportamiento del atacante al intentar comprometer otro sistema.

Otros softwares que permiten crear honeypots, un tanto más "especializados", y aunque interesantes puede que más obvios para un atacante, son los siguientes:

- Gridpot: permite simular un SCADA de red eléctrica de forma realista.
- GasPot: diseñado para simular un medidor de tanque modelo Guardian AST del fabricante Veeder Root, común en la industria petrolera.
- iHoney: permite simular una planta de tratamiento de aguas y todos sus posibles elementos. Su desarrollo se enmarcó en un proyecto de investigación realizado en 2017 por el Ministerio de Industria, Energía y Turismo de España.

7. Demostración práctica

En esta sección del trabajo vamos a intentar mostrar el funcionamiento de un honeypot de una forma más visual. Para ello, vamos a crear un honeypot sencillo en un ambiente "virtualizado" haciendo uso de un sistema Ubuntu 22.04 a través del módulo Windows Subsystem For Linux (WSL2) de nuestro computador con Windows 11 Pro. Este honeypot será de baja interacción: no vamos a hacer que ofrezca ningún servicio como servidor, simplemente lo añadiremos a nuestra red como si fuese un computador más. Técnicamente, este sistema Ubuntu sólo es visible para la máquina Windows anfitriona, y no puede ser accedido desde fuera de este (ni siquiera en la red local).

La configuración de red del honeypot es la siguiente:

```
modejota@MSI:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.19.58.175  netmask 255.255.240.0  broadcast 172.19.63.255
        inet6 fe80::215:5dff:fe4b:21dc  prefixlen 64  scopeid 0x20<link>
        ether 00:15:5d:4b:21:dc  txqueuelen 1000  (Ethernet)
        RX packets 69  bytes 7313 (7.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 6  bytes 516 (516.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Bucle local)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Imagen 5. Configuración de red del Honeypot.

Solamente se dispone de una interfaz de red, que lo conecta con la máquina anfitriona, y que es emulada como un puerto Ethernet. Según la documentación de WSL esta dirección no es completamente estática, lo cual deberemos tener en cuenta cuando configuremos Snort.

Por defecto, las distribuciones Linux instaladas mediante WSL2 no disponen de servidor SSH. Dado que lo necesitaremos en pruebas posteriores, lo instalamos (*apt install openssh-server)* e iniciamos (*service ssh start*). Se mantendrá en el puerto por defecto (22), aunque podría ser interesante moverlo a un puerto no estándar para evitar ataques sistemáticos como el escaneo de puertos que realizaremos posteriormente.

Además, instalamos Snort (*apt install snort)*, lo que nos permitirá tener un honeypot de baja interacción. Durante la instalación, este software nos preguntará por la dirección IP donde deberá monitorizar el tráfico recibido. Dado que la dirección es cambiante, dejamos el dialogo de la imagen 6 vacío y realizaremos la configuración posteriormente en el fichero correspondiente.

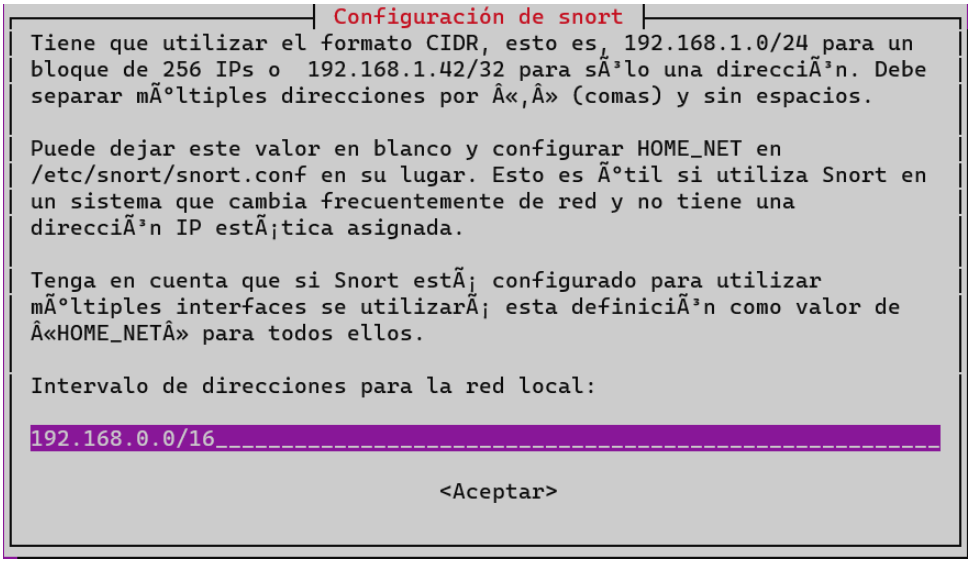


Imagen 6. Configuración inicial de Snort.

Este software tiene bastante potencial porque, además de venir con una serie de reglas predefinidas por la comunidad, permite añadir nuevas reglas para monitorizar las partes del sistema que se deseen. Dispone de un conjunto de reglas adicionales, que puede obtenerse previo pago de una suscripción de 30$ por sensor si se trata de uso personal.

La configuración de Snort es muy sencilla, simplemente se debe modificar el archivo */etc/snort/snort.conf* (la ruta exacta puede variar en función del sistema operativo, se muestra la de Linux). Dentro de este archivo tenemos una serie de variables de entorno que debemos cambiar. Las dos primeras son *HOME_NET* y EXTERNAL_NET; que corresponden a la dirección IP de la máquina honeypot y a lo que vamos a considerar direcciones fueras de la red local. La dirección IP de la máquina local se mostró anteriormente, 172.19.58.175/20; mientras que consideraremos equipos externos todos aquellos que no sean la propia máquina honeypot. Este segundo filtro podría facilitar el no tratar (o tratar de forma diferente) según que amenazas en función de si provienen de la red local o no. Otra variable definida es *RULE_PATH,* que determina la ruta hasta el directorio donde se alojan las reglas, y cuyo valor mantendremos por defecto. En la siguiente imagen se mostrará el fichero de configuración con las variables de entorno configuradas:

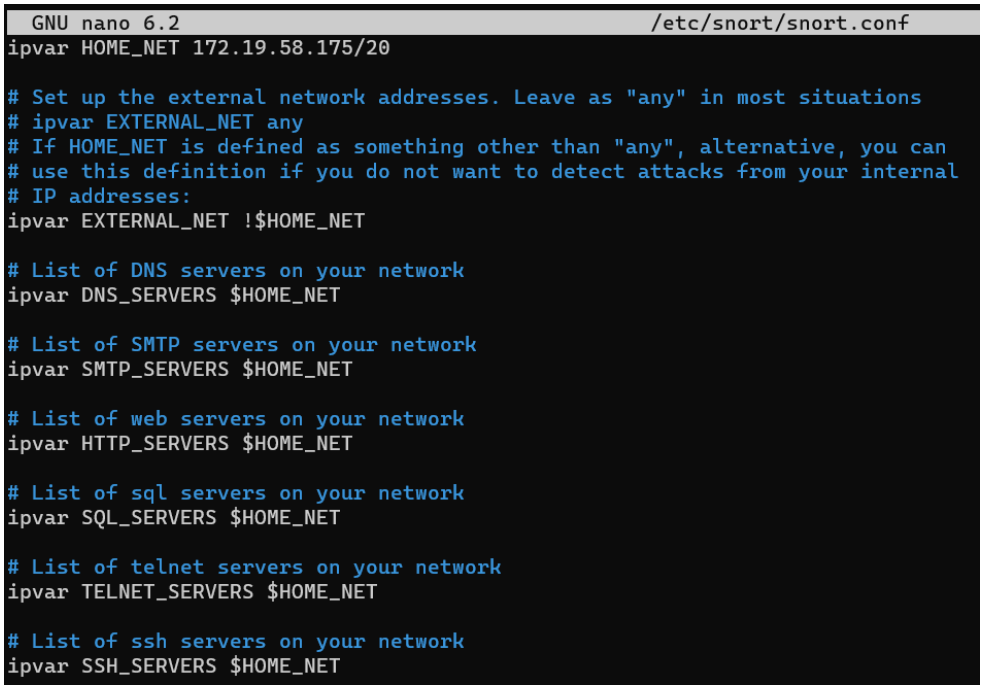


Imagen 7. Parte de las variables de configuración de Snort.

Snort hace uso de reglas que permiten personalizar las diferentes alertas de las que nos notificará el honeypot y sus condiciones de actvación. Estas se encuentran en el directorio */etc/snort/rules/*, donde podremos usar unas reglas por defecto ya instaladas (mostradas en la imagen 8,) y entre las que podemos encontrar reglas para ataques web, SQL, FTP, etc; descargar más desde la web oficial de Snort; o bien podremos crear nuestro propio archivo *.rules* y definir nuestras propias reglas. Por defecto se tiene un fichero *local.rules* para que definamos nuestras propias reglas.

Imagen 8. Ficheros de reglas por defecto de Snort.

Para lanzar Snort haremos uso del siguiente comando:
sudo snort -A console -i eth0 -c /etc/snort/snort.conf

Con este comando especificamos que la salida se muestre por consola, se monitorice la interfaz eth0 y se utilice el fichero de configuración por defecto. Debe ejecutarse con permisos de superusuario en tanto que la herramienta captura y analiza tráfico de red. Opcionalmente se podría usar el flag *-q* para ocultar el registro del programa, el cual es bastante largo.

Si ejecutamos esto, con los ficheros de reglas que se cargan por defecto, Snort nos notifica de que se cargan 3385 reglas. El registro nos detalla algo más sobre el número de reglas para cada diferentes protocolos, cuales afectan a tráfico de entrada, salida o ambos.

```
4059 Snort rules read
    3385 detection rules
    0 decoder rules
    0 preprocessor rules
3385 Option Chains linked into 951 Chain Headers
+++++++++++++++++++++++++++++++++++++++++++++++++

+-------------------[Rule Port Counts]---------------------------------------
|             tcp     udp    icmp      ip
|     src     151      18       0       0
|     dst    3307     126       0       0
|     any     383      48      53      22
|      nc      27       8      16      20
|     s+d      12       5       0       0
+-----------------------------------------------------------------------------
```

Imagen 9. Carga de reglas por defecto de Snort.

Con esta configuración, y por probar algo simple, si intentamos hacer un escaneo de puertos con NMAP obtendremos una alerta de SNMP indicando que puede que haya un "leak" de información. NMAP por su parte sólo detecta como abierto el puerto 22 (lo cual es intencional)

```
C:\Program Files (x86)\Nmap>nmap 172.19.58.175
Starting Nmap 7.94 ( https://nmap.org ) at 2023-06-10 20:27 Hora de verano romance
Nmap scan report for 172.19.58.175
Host is up (0.00033s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:15:5D:FB:1B:41 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds
```

```
modejota@MSI:~$ sudo snort -A console -q -i eth0 -c /etc/snort/snort.conf
06/10-20:27:45.184860  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Informa
tion Leak] [Priority: 2] {TCP} 172.19.48.1:63927 -> 172.19.58.175:161
06/10-20:27:45.194231  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted
Information Leak] [Priority: 2] {TCP} 172.19.48.1:63927 -> 172.19.58.175:705
^C*** Caught Int-Signal
```

Imágenes 10 y 11. Ejecución de NMAP y aviso de Snort por una petición SNMP.

En este momento, acudimos al fichero */etc/snort/snort.conf* y comentamos la parte del mismo que indica la carga de las reglas por defecto presentes en el directorio *rules* a excepción del fichero *local.rules.* A este fichero le añadiremos un par de reglas personalizadas.

La primera de ellas permite detectar la IP origen de un ping. La otra regla que hemos añadido alerta de posibles ataques de fuerza bruta mediante SSH. Para el caso de la segunda regla generaremos un mensaje si se intenta acceder mediante SSH (puerto 22 sobre TCP) sin éxito 2 veces en menos de 60 segundos. Los valores SID son identificadores únicos que utiliza Snort para distinguir a cada regla (se suele usar valores superiores al millón para reglas personalizadas).

- *alert icmp any any -> $HOME_NET any (msg:"ICMP TEST"; sid:10000001; rev:001;)*
- *alert tcp any any -> $HOME_NET 22 (msg:"Possible SSH brute forcing!"; flags: S+; threshold:type threshold, track by_src, count 2,seconds 60; sid:10000002; rev:001;)*

Ahora que ya tenemos todo configurado en nuestro honeypot, iniciamos Snort para que empiece a recabar datos; si desde la máquina host (Windows 11 Pro) hacemos un ping a la máquina virtual, veremos cómo se registra en el log. Esto mismo ocurre con las conexiones SSH; intentaremos acceder sin éxito dos veces para que nos alerte de una posible intrusión.

```
modejota@MSI:~$ sudo snort -A console -q -i eth0 -c /etc/snort/snort.conf
06/11-11:26:37.552144  [**] [1:10000001:1] ICMP TEST [**] [Priority: 0] {ICMP} 172.19.48.1 -> 172.19.58.175
06/11-11:26:37.552170  [**] [1:10000001:1] ICMP TEST [**] [Priority: 0] {ICMP} 172.19.58.175 -> 172.19.48.1
06/11-11:26:38.640496  [**] [1:10000001:1] ICMP TEST [**] [Priority: 0] {ICMP} 172.19.48.1 -> 172.19.58.175
06/11-11:26:38.640526  [**] [1:10000001:1] ICMP TEST [**] [Priority: 0] {ICMP} 172.19.58.175 -> 172.19.48.1
06/11-11:26:39.746596  [**] [1:10000001:1] ICMP TEST [**] [Priority: 0] {ICMP} 172.19.48.1 -> 172.19.58.175
06/11-11:26:39.746632  [**] [1:10000001:1] ICMP TEST [**] [Priority: 0] {ICMP} 172.19.58.175 -> 172.19.48.1
06/11-11:26:40.848791  [**] [1:10000001:1] ICMP TEST [**] [Priority: 0] {ICMP} 172.19.48.1 -> 172.19.58.175
06/11-11:26:40.848824  [**] [1:10000001:1] ICMP TEST [**] [Priority: 0] {ICMP} 172.19.58.175 -> 172.19.48.1
```

Imagen 12. Mensajes generados por Snort en el honeypot al detectar el ping de la otra máquina.

```
C:\Program Files (x86)\Nmap>ssh modejota@172.19.58.175
modejota@172.19.58.175's password:
Permission denied, please try again.
modejota@172.19.58.175's password:
Permission denied, please try again.
modejota@172.19.58.175's password:
modejota@172.19.58.175: Permission denied (publickey,password).

C:\Program Files (x86)\Nmap>ssh modejota@172.19.58.175
modejota@172.19.58.175's password:
Permission denied, please try again.
modejota@172.19.58.175's password:
Permission denied, please try again.
modejota@172.19.58.175's password:
modejota@172.19.58.175: Permission denied (publickey,password).
```

Imagen 13. Intentos de conexión con SSH al honeypot desde Windows.

```
modejota@MSI:~$ sudo snort -A console -q -i eth0 -c /etc/snort/snort.conf
06/10-20:36:09.741625  [**] [1:1000000002:1] Possible SSH brute forcing! [**] [Priority: 0] {TCP} 172.19.48.1:33524 -> 172.19.58.175:22
```

Imagen 14. Mensaje generado por Snort al detectar las conexiones SSH fallidas.

Como se puede ver en las anteriores imágenes, podemos ver la IP origen y el puerto desde el que se intenta acceder al honeypot y simplemente hemos implementado dos reglas en un sencillo honeypot de baja interacción. Aunque nosotros hemos intentado “atacar” el honeypot desde dentro de la propia red, cualquier otro ataque externo que se produzca también sería detectado por Snort y mostraría las respectivas interacciones.

Además de mostrar logs con información resumida, Snort es capaz de proporcionar información más detallada de los paquetes que recibe y envía el sistema. Puede actuar como un sniffer de cierto tipo de tráfico. En el anexo se adjuntan imágenes adicionales donde queda patente este comportamiento para tráfico SSH, HTTP e ICMP. Adicionalmente, podríamos configurar, por ejemplo, que se nos envíe un correo electrónico cuando se produzcan determinadas alarmas, lo cual podría ayudarnos a frenar un ataque real rápidamente. Aunque no se realiza en esta práctica, dentro de */etc/snort/snort.conf* deberíamos tener configurado: *output alert_email: <SMTP_SERVER_IP>, <SENDER_EMAIL>, <RECEIVER_EMAIL>, <EMAIL_SUBJECT>* o bien podríamos utilizar servicios de terceros como Swatch.

8. Conclusiones

Como hemos visto, los honeypots son una herramienta bastante potente y flexible, que se puede utilizar en prácticamente cualquier sistema informático. Por una parte, son potentes por la capacidad que tienen de poder detectar amenazas y, además, recabar información importante sobre ellas de modo que se puedan implementar mejoras de seguridad en el sistema real. Por otra parte, son flexibles porque hay muchas y muy variadas formas de poder implementar un honeypot, de tal manera que se pueden adaptar perfectamente a las necesidades de nuestro sistema.

Desde honeypots de baja interacción (como el ejemplo que usamos en la demo) donde la información recogida no es muy amplia hasta honeypots puros con servicios completos que nos permiten monitorizar el sistema señuelo y obtener datos detallados de los ataques. Todos ellos son útiles en el ámbito de la ciberseguridad ya que ofrecen (en mayor o menor medida) un sistema de prevención de ataques rentable que permite mejorar la protección de los sistemas informáticos.

No obstante, también nos gustaría destacar que, como cualquier herramienta de ciberseguridad, los honeypots nos son perfectos: si se gestionan mal, lejos de ofrecer una ventaja, pueden llegar a ser una gran brecha de seguridad para nuestro sistema. Como hemos mencionado en el trabajo, es muy importante aislar totalmente el honeypot de la red real de nuestro sistema, ya que de lo contrario el atacante podrá establecer puertas traseras en el honeypot (por ejemplo) para luego poder acceder al sistema real sin autenticarse.

9. Anexo

```
WARNING: No preprocessors configured for policy 0.
06/11-11:41:25.400893 172.19.48.1:32813 -> 172.19.58.175:22
TCP TTL:128 TOS:0x0 ID:61525 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x35160B15  Ack: 0x4D06112B  Win: 0x401  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/11-11:41:26.365674 172.19.48.1:32813 -> 172.19.58.175:22
TCP TTL:128 TOS:0x0 ID:61526 IpLen:20 DgmLen:124 DF
***AP*** Seq: 0x35160B15  Ack: 0x4D06112B  Win: 0x401  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/11-11:41:26.409030 172.19.58.175:22 -> 172.19.48.1:32813
TCP TTL:64 TOS:0x0 ID:36590 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x4D06112B  Ack: 0x35160B69  Win: 0x1F5  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/11-11:41:29.222776 172.19.58.175:22 -> 172.19.48.1:32813
TCP TTL:64 TOS:0x0 ID:36591 IpLen:20 DgmLen:92 DF
***AP*** Seq: 0x4D06112B  Ack: 0x35160B69  Win: 0x1F5  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/11-11:41:29.281905 172.19.48.1:32813 -> 172.19.58.175:22
TCP TTL:128 TOS:0x0 ID:61527 IpLen:20 DgmLen:40 DF
***A**** Seq: 0x35160B69  Ack: 0x4D06115F  Win: 0x401  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Imagen 15. Captura de paquetes correspondientes a una solicitud SSH.

```
WARNING: No preprocessors configured for policy 0.
06/11-11:40:19.783874 172.19.48.1 -> 172.19.58.175
ICMP TTL:128 TOS:0x0 ID:61513 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:22  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/11-11:40:19.783918 172.19.58.175 -> 172.19.48.1
ICMP TTL:64 TOS:0x0 ID:41753 IpLen:20 DgmLen:60
Type:0  Code:0  ID:1  Seq:22  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/11-11:40:20.874896 172.19.48.1 -> 172.19.58.175
ICMP TTL:128 TOS:0x0 ID:61514 IpLen:20 DgmLen:60
Type:8  Code:0  ID:1   Seq:23  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/11-11:40:20.874926 172.19.58.175 -> 172.19.48.1
ICMP TTL:64 TOS:0x0 ID:41804 IpLen:20 DgmLen:60
Type:0  Code:0  ID:1  Seq:23  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Imagen 16. Captura de paquetes ICMP correspondientes a un ping.

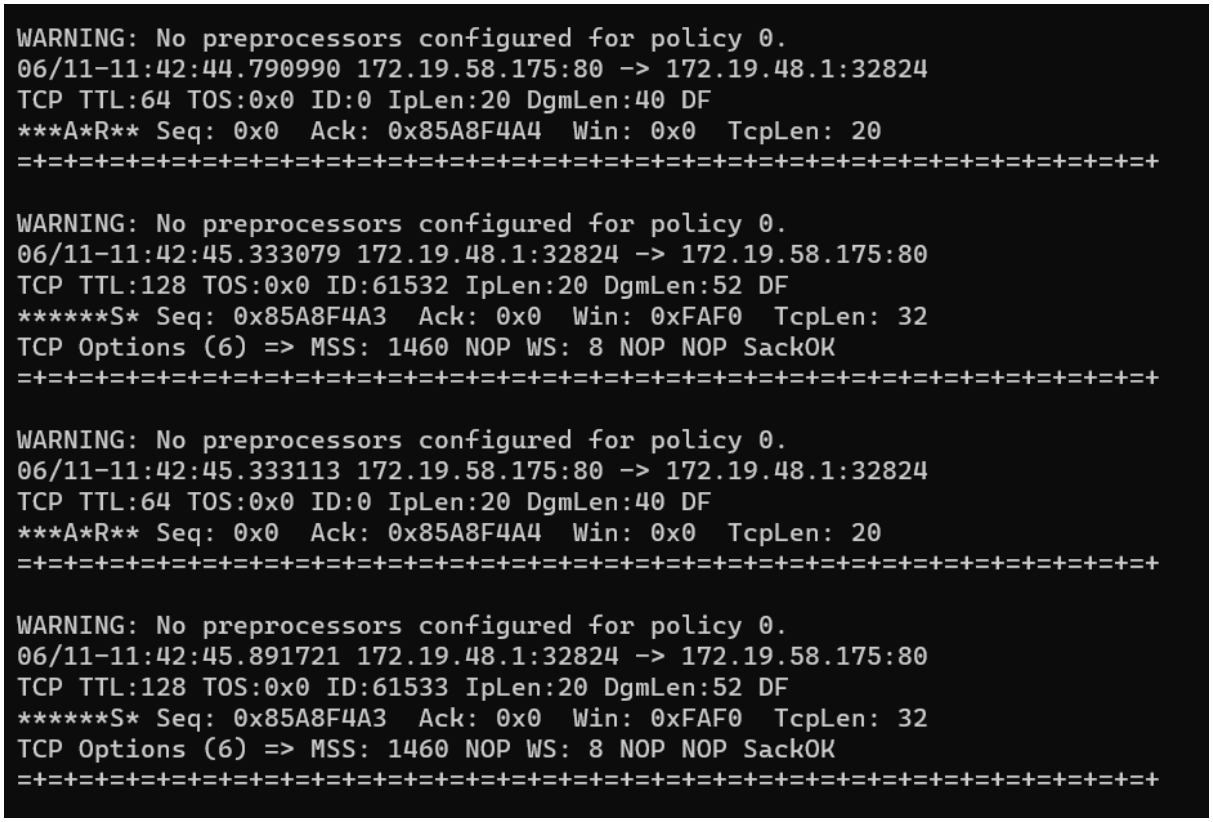


Imagen 17. Captura de tráfico correspondiente a una solicitud HTTP.

Para hacer que Snort se comporte como si fuera un sniffer de tráfico, sin generar alertas de ningún tipo, basta con ejecutarlo con el comando *sudo snort,* sin pasarle ninguna bandera que indique donde mostrar la salida, interfaz a monitorizar o fichero de configuración a emplear.

10. Bibliografía

Seguridad en la red. En Ciberseguridad.
https://ciberseguridad.com/normativa/espana/medidas/seguridad-red/

La breve historia de la ciberseguridad.
https://www.sofistic.com/blog-ciberseguridad/la-breve-historia-de-la-ciberseguridad/

Definición e historia de los Honeypots
https://1library.co/article/definici%C3%B3n-historia-honeypots-an%C3%A1lisis-conceptual.z3d93n59

Honeypot (computing)
https://en.wikipedia.org/wiki/Honeypot_(computing)

Honeypots
https://wikis.fdi.ucm.es/ELP/Honeypots

Los principales ciberataques y brechas de datos de 2022: al alza los accesos no autorizados y el ransomware.
https://bitlifemedia.com/2023/02/mayores-ciberataques-brechas-datos-de-2022/

Los responsables del ciberataque a NVIDIA exigen que los drivers GeForce sean de código abierto.
https://www.muycomputer.com/2022/03/02/ciberataque-a-nvidia/

SQL injection
https://en.wikipedia.org/wiki/SQL_injection

Qué es un honeypot y cómo implementarlo en nuestra red.
https://www.welivesecurity.com/la-es/2020/07/31/que-es-honeypot-como-implementarlo-nuestra-red/

¿Qué es un señuelo o honeypot?
https://www.kaspersky.es/resource-center/threats/what-is-a-honeypot

Qué es y para qué sirve un Honeypot.
https://www.redeszone.net/tutoriales/seguridad/que-es-honeypot/

Spam Traps (trampas contra correo no deseado).
https://support.wix.com/es/article/spam-traps-trampas-contra-correo-no-deseado

¿QUÉ SON LOS SPAMTRAPS (CORREOS TRAMPA) Y CÓMO EVITARLOS?
https://emailmarketingbootcamp.es/que-son-los-spamtraps-correos-trampa-y-como-evitarlos

Qué es una araña web y cómo afecta al posicionamiento SEO
https://www.publisuites.com/blog/arana-web/

Web Crawler.
https://en.wikipedia.org/wiki/Web_crawler

Honeypot, una herramienta para conocer al enemigo.
https://www.incibe.es/incibe-cert/blog/honeypot-herramienta-conocer-al-enemigo

How to establish a honeypot on your network.
https://www.comparitech.com/net-admin/how-to-establish-a-honeypot-on-your-network

Soenke, Justin. How to Create a Honeypot to Catch a Hacker.
https://phase3.net/blogs/tech-bytes/how-to-create-a-honeypot-to-catch-a-hacker

KFSensor Tour
https://www.kfsensor.net/kfsensor/tour/

GLASTOPF: HONEYPOT DE APLICACIONES
https://revista.seguridad.unam.mx/numero25/glastopf-honeypot-de-aplicaciones-web-i

Una colmena de Honeypots para atraparlos a todos
https://www.elladodelmal.com/2017/07/t-pot-una-colmena-de-honeypots-para.html

Honeypot: Ventajas y Desventajas como Mecanismo para la prevención de Intrusos Informáticos.
http://repository.unipiloto.edu.co/bitstream/handle/20.500.12277/2607/00000846.pdf?sequence=1#:~:text=2

Penetration test.
https://en.wikipedia.org/wiki/Penetration_test

How to install Snort on Ubuntu.
https://upcloud.com/community/tutorials/install-snort-ubuntu/

Basic snort rules syntax and usage
https://resources.infosecinstitute.com/topic/snort-rules-workshop-part-one/

Network Instrusion Detection System (Snort)
https://www.youtube.com/watch?v=iBsGSsbDMyw

Honeypots y Honeynets.
https://www.cs.upc.edu/~gabriel/files/DEA-es-4HoneypotsyHoneynets.pdf