Universidad de Valladolid

Trabajo de Fin de Grado

Desvío de intrusiones hacia honeynets dinámicas virtuales

Author:
Helena Carbajo

Tutor:
FedericoSimross

A thesis submitted in fulfillment of the requirements
for the degree of Grado en Tecnologías Específicas de Telecomunicación.
Mención en Telemática

in the

Teoría de la Señal
Teoría de la señal

11 de marzo de 2018

DeclaraciÃşn de autorÃŋa

I, Helena Carbajo, declare that this thesis titled, «Desvío de intrusiones hacia honeynets dinámicas virtuales» and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSIDAD DE VALLADOLID

Resumen

Escuela Técnica Superior de Ingenieros de Telecomunicación
Teoría de la señal

Grado en Tecnologías Específicas de Telecomunicación. Mención en Telemática

Desvío de intrusiones hacia honeynets dinámicas virtuales

por Helena Carbajo

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

Agradecimientos

The acknowledgments and the people to thank go here, don't forget to include your project advisor...

Índice general

Índice de figuras

Índice de cuadros

List of Abbreviations

LAH **L**ist **A**bbreviations **H**ere
WSF **W**hat (it) **S**tands **F**or

For/Dedicated to/To my...

Capítulo 1

Introducción

1.1. Seguridad en Internet

Es posible encontrar múltiples definiciones de lo que se entiende por seguridad en la red o Internet. Por ejemplo, Stallings se refiere a ello como la protección que se proporciona a un sistema de la información para preservar la integridad, disponibilidad y confidencialidad de sus recursos, tanto software como hardware [10]. Por otro lado, la empresa Cisco la define como la actividad destinada a proteger la usabilidad y la integridad de la red y datos. Al igual que la anterior, engloba medios software y hardware [3]. Estas son tan solo dos ejemplos de las muchas acepciones que existen, pero es posible observar que coinciden en gran medida en los aspectos que la seguridad, en términos de Internet, debería garantizar. Ambas también establecen los mismos objetivos a proteger: medios hardware y software. Es importante entender en que consisten la integridad, la disponibilidad y la confidencialidad, conjunto conocido como *CIA* (*Confidenciality, Integrity and Availability*). La confidencialidad asegura que datos de carácter privado no sean accedidos por personas no autorizadas; la disponibilidad permite que datos o cualquier otro tipo de recurso pueda ser utilizado sin ningún tipo de impedimento y, finalmente, la integridad preserva el contenido de los datos o comportamiento de un sistema, de manera que estos no sean modificados por alguien desautorizado. Todos estos términos pueden aplicarse a un sistema aislado, pero cuando este sistema pasa a estar en una red de millones de nodos, las amenazas se múltiplican.

1.1.1. Tipos de atacantes

Se distinguen distintos tipos de atacantes: hackers, criminales y empleados. Los hackers suelen realizar ataques buscando la emoción de conseguir acceder a un sistema restringido y el reconocimiento del resto de la comunidad hacker. Son frecuentes los ataques de oportunidad que aprovechan alguna vulnerabilidad para acceder a información que luego comparten en la red. Los segundos atacantes, los criminales, constituyen bandas de hackers que se asocian para llevar a cabo ataques con fines lucrativos, generalmente contra servicios de comercio electrónico. Tratan de hacerse con datos bancarios y tarjetas de crédito que después utilizan a expensas de la víctima o venden en la red. Estos grupos, que se han expandido por toda la red, suponen una amenaza común para todos los sistemas basados en Internet, buscan objetivos concretos y, en ocasiones, son contratados por gobiernos u otras organizaciones. Por úlimo, los empleados son individuos que ya se encuentran dentro del sistema y conocen su estructura. Sus ataques pueden estar motivados por venganza contra la organización en la que trabajan o sencillamente, por un sentimiento de derecho. Resultan, por lo tanto, los ataques más difíciles de detectar y prevenir, y solamente políticas de acceso y monitorización dentro de la organización ayudan a evitarlos[10].

1.1.2. Ataques de seguridad y sus motivaciones

Las normas *X.800* y *RFC 4949* clasifican los ataques en dos categorías: pasivos y activos. Los ataques pasivos serían aquellos que extraen información de un sistema, pero no alteran en modo alguno a sus recursos. Un ejemplo sería la monitorización de las transmisiones realizadas entre dos sistemas, accediendo a esta información. Por otro lado, los ataques activos sí que afectan a los recursos de un sistema e incluso a su funcionamiento. Dentro de este tipo de ataques se encuentran la suplantación, cuando un individuo u organización finge ser otra distinta; reenvio de información capturada previamente sin autorización; modificación de mensajes o la denegación de servicio, que impide el acceso normal a un servicio[10]. En lo que respecta a las motivaciones, la gran mayoría de ataques están conducidos por el espionaje o un interés financiero. Otras motivaciones serían la diversión, el resentimiento o la ideología. No obstante, hay que tener en cuanto que muchos casos de extorsión no son reportados y confirmados, por lo que las cifras recogidas en las estadísticas no reflejan la totalidad de los ataques. Aún así, es posible contar con una referencia de los fines que persiguen algunos conocidos ataques[12]:

- Financieros: uso de credenciales robadas, uso de backdoor, spyware, phising, malware para exportar data, c2.
- Espionaje: phising, c2, uso de backdoor.
- Resto: abuso de privilegios

1.2. Algunas cifras concretas

En relación a lo anterior, existen multitud de estudios e informe que tratan de recabar información acerca del estado de la seguridad en Internet partiendo de diversas fuentes, como encuestas o ataques sufridos. Pese a que gran parte de estos estudios están realizados por empresas privadas, resultan útiles para obtener una perspectiva global del problema que supone la seguridad en Internet.

1.2.1. DBIR 2017

El DBIR (*Data Breach Investigations Report*), un informe realizado por Verizon en el que participan 65 organizaciones, analiza el estado de la ciberseguridad.Según este informe hubo 1616 ataques durante el año 2016, de los que 828 supusieron la revelación de datos confidenciales. Este informe también proporciona los tipos de ataques más conocidos, así como los actores que los perpetran y sus motivaciones. Plasmando en cifras lo referido en la anterior sección, el 66 % de los ataques tenían una motivación financiera y el 33 % de espionaje. Menos del 1 % de los ataques fueron motivados por ideología o diversión. Además, el 99 % de estos ataques los llevaron a cabo individuos u organizaciones externas. Otro dato interesantes que recoge el informe es que la táctica más empleada es la de phising y que la mayoría de estos ataques van seguidos por la instación de algún tipo de malware. Finalmente, cabe mencionar que el 81 % de las brechas de seguridad se produjeron debido a credenciales inseguras o robadas. Este informe además de analizar las estadísticas de los ataques reportados, proporciona algunos consejos en base a los resultados para tratar de evitarlos. Destaca, sobretodo, la necesidad de concienciar y educar acerca de las amenzadas y riesgos que existen. También pone el foco en la importancia que supone la detección temprana de un ataque y la localización de la fuente del ataque[12].

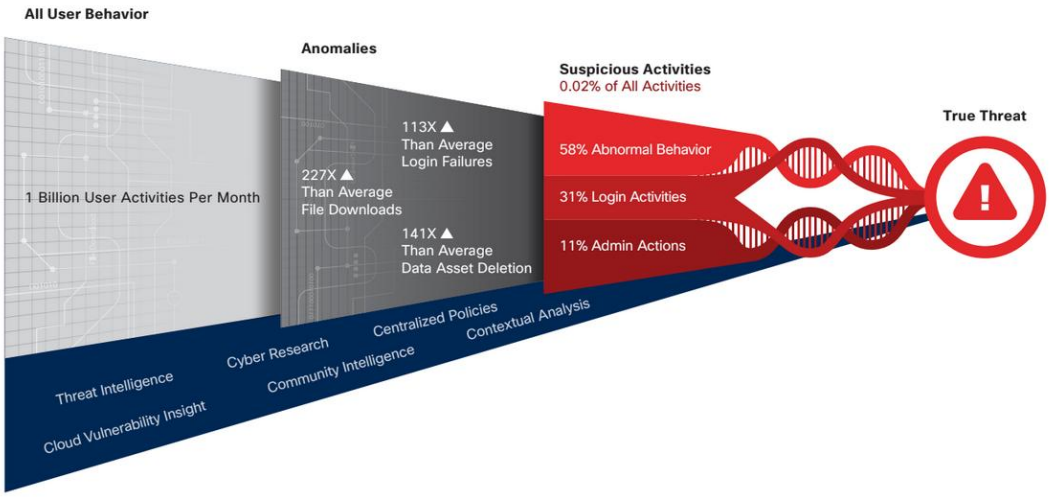

FIGURA 1.1: Identificación de patrones de comportamiento de usario normal

1.2.2. Cisco 2017 Annual Cybersecurity Report

El grupo de investigación de seguridad de Cisco publica cada año este informe, para ayudar a las organizaciones a hacer frente a las amenazas y riesgos que surgen constantemente en la red. Entre los datos recogidos por el informe cabe destacar las razones que impiden la adopción de sistemas u otras medidas de seguridad en muchas empresas. El 35 % carecía de presupuesto, para el 28 % presentaba problemas de compatibilidad, el 25 % por la certificación y el 25 % restante por falta de talento. Por este motivo, apenas la mitad de las alertas de seguridad que se reciben son investigadas. Cabe mencionar también el hecho de que aquellas organizaciones que aún no han sufrido ninguna brecha de seguridad están convencidas de que su red es segura, aunque esta seguridad parece cuestionable si se tiene en cuenta el grado de afectación que supone para cualquier empresa que su sistema se vea comprometido. Casi un cuarto de las empresas perdió alguna oportunidad de negocio al sufrir un ataque y 1 de cada 5 perdió clientes. Este estudio también muestra que muchas de las empresas recurren a las soluciones de seguridad de varias empresas especializadas, por lo general más de 5, con varios productos distintos también. Todo ello supone una complejidad extra que dificulta la automatización de tareas, algo fundamental a la hora de mejorar la seguridad de un sistema. Por ejemplo, distinguir un comportamiento anómalo y sospechoso del que, según los patrones, resulta normal requiere un proceso de varias etapas que solo puede lograrse con automatización1.1.
En lo que se refiere a los ataques, los datos revelan que en la mayoría de ellos se distinguen las siguientes fases:

- Reconomiento: los atacantes investigan, identifican y seleccionan a sus víctimas.
- Armamento: generación de paquetes con malware que permite el acceso remoto aprovechando una vulnerabilidad.
- Distribución: la carga anterior se hace llegar mediante correo, ficheros adjuntos, etc.
- Instalación: en el objetivo, el malware genera una puerta trasera que permite el acceso permanente de los atacantes.

Frente a los ataques, una de las medidas que propone cisco para conocer el progreso de las medidas de seguridad es el TTD (*Time To Detec*. Lo define como el intervalo de tiempo que transcurre desde que un sistema se ve comprometido hasta que la amenaza es detectada. Las amenazas y ataques evolucionan muy rápidamente y en ocasiones resulta difícil identificar un ataque, aunque este sea conocido en la comunidad. De la misma manera que los sistemas de seguridad trabajan en mejorar el TTD, los atacantes desarrollan nuevas técnicas y estrategias para evitar ser detectados y disponer así de más tiempo para perpetrar su ataque. Esta mejora en los ataques se puede medir con el TTE (*Time To Evolve*, el tiempo que tarda un atacante en modificar el modo en que cierto malware es distribuido o en cambiar de táctica. El hecho de que los ataques evolucionen con tanta rapidez denota, a su vez, las mejoras que experimentan los sistemas de seguridad.
Este progreso constante en ambos partes, ha supuesto un incremento en el personal dedicado exclusivamente a la seguridad en las empresas. Frente a las 25 personas que se registraron de media en cada organización durante el año 2015, el año 2016 esta cifra era de 33. El interés por combatir el progreso de las amenazas en Internet radica en el impacto que estos ataques tienen en una organización o empresa. La repercusión no se limita a cortes de servicio, con la consecuente pérdida de dinero, sino que afecta gravemente a la reputación. De hecho, el 33 % de las organizaciones encuestadas tuvieron que hacer frente a la publicación involuntaria de ataques que sufrieron.
El informe concluye que toda organización es susceptible de sufrir un ciberataque, siendo necesaria una constante mejora en los medios de seguridad teniendo en cuenta, además, las limitaciones de presupuesto y compatibilidad, entre otras[2]

1.3. Búsqueda de soluciones contra los ciberataques

Los informes comentados anteriormente reflejan tan solo una pequeña parte del total de informes elaborados por empresas y organizaciones. Pese a que ambos han sido llevados a cabo por empresas privadas con sus propios interes y es necesario estudiarlos con una actitud crítica, sí que ponen de manifiesto la magnitud del problema que supone la seguridad en Internet. Como puede extraerse de los datos, en la red toda organización es susceptible de convertirse en víctima de un ataque y la necesidad de sistemas seguros se mantiene.
Se ha discutido anteriormente cómo son y en qué consisten algunos de los ataques más frecuentes de la red, ¿pero qué soluciones existen para hacerlos frente? Se han desarrollado multitud de estándars que abarcan desde técnicas de gestión de ataques, hasta arquitecturas recomendables. Algunas de las instituciones más importantes que realizan esta tarea son el NIST(*National Institue of Standards and Technology*), una agencia federal de Estados Unidos; o la ISOC (*Internet Society*), que elabora RFCs(*Requests For Comments*)[10]. Además de estos estándares, hay diversos productos concretos que los expertos en ciberseguridad utilizan. La organización secTools proporciona una lista de las herramientas más populares[5]

- **Wireshark**: se trata de un analizador de protocolos, que permite estudiar el tráfico de red.
- **Metasploit**: herramienta para probar y desarrollar código de *exploits*.
- **Nessus**: permite realizar escáneres.
- **Aircrack**: utilidad para romper algoritmos de encriptación y recuperar contraseñas para los protocolos 802.11a/b/g WEP and WPA.

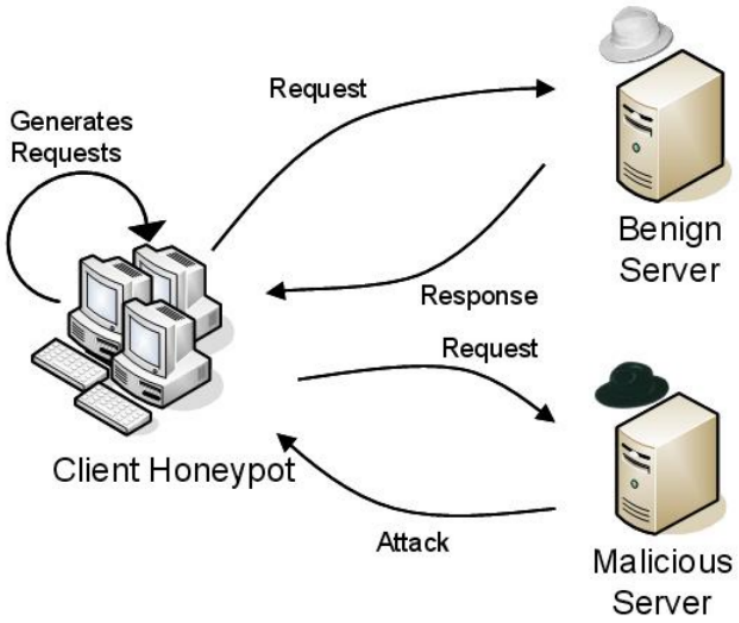

FIGURA 1.2: Funcionamiento de un cliente *honeypot*

- **Snort**: es un IDS(*Intrusion Detection Sistem*), que tal y como su nombre indica, se emplea para detectar instrusiones.
- **Cain and Abel**: herramienta para recuperar contraseñas en Windows.
- **BackTrack**: distribución de Linux que engloba herramientas forenses y de seguridad.
- **Netcat**: permite leer y escriber datos a través de conexiones TCP y UTP.
- **Tcpdump**: al igual que Wireshark analiza el tráfico de red, pero se trata de una herramienta de línea de comandos.
- **John the Ripper**: se utiliza para *crackear* contraseñas en sistemas Unix y Mac, permitiendo detectar contraseñas débiles.

Como puede comprobarse, gran parte de estas herramientas son empleadas tanto para auditorías, como para perpetrar ataques, puesto que a la hora de comprobar la eficacia de cualquier sistema de seguridad, resulta fundamental ponerlo a prueba con ataques reales. Y es ahí donde entran en juego lo que se conoce como *honeypots*.

1.3.1. Honeypots

Los *honeypots* son sistemas aislados en la red que actúan como señuelos de cara a posibles atacantes, evitando así que accedan a información crítica. Por este motivo, han de simular eficazmente un sistema productivo, pero carente de datos reales o sensibles, de manera que el atacante no sea consciente del engaño y tenga una falsa sensación de seguridad. Un *honeypot* está diseñado con el objetivo de retener al atacante el mayor tiempo posible, siendo éste uno de sus mayores atractivos, pues permite estudiar su comportamiento. Tal y como se ha mencionado anteriormente, resulta de vital importancia entender cómo piensa y actúa un *hacker* para diseñar mejores estrategias de defensa[10]. Existen diferentes tipos de *honeypots*. Si se atiende a su funcionamiento

y ámbito de actuación, tradicionalmente han actuado de forma pasiva en el lado del servidor, esperando intrusiones y monitorizando los ataques que se produjeran. Un ejemplo de este tipo sería *Kippo*, un SSH *honeypot*, que permite levantar puertos que esperen conexiones SSH de manera que los atacantes puedan acceder al sistema e interactuar con un sistema de ficheros ficticios, mientras toda esta actividad queda registrada[11]. Frente a este modelo pasivo se encuentran los clientes *honeypot*, que se encuentran en el lado del cliente e investigan posibles ataques desde servidores de manera activa. Cuando se detecta una posible amenaza o intrusión, el sistema realiza una serie de peticiones para detectar servidores malignos, tal y como se ilustra en la figura 1.2 [7].

Dentro de este tipo de *honeypots* se encuentra *Shelia*, que emula el comportamiento de un usuario que accediese a los enlaces que incluyen los correos de spam, detectando así qué servidores suponen un peligro para la seguridad del sistema[1].

1.3.2. La evolución hacia las honeynets

Partiendo del concepto de *honeypot* y teniendo presente la necesidad de desarrollar sistemas económicos que confronten las limitaciones presupuestarias que existen en el ámbito de la ciberseguridad, el presente proyecto tiene por objetivo desarrollar una *honeynet*, que emplee máquinas virtuales creadas dinámicamente para gestionar ataques e intrusiones. Para ello, aunará la capacidad de detección de un IDS con herramientas de virtualización, de manera que el resultado sea un sistema integrado.

Capítulo 2

Sistemas de Detección de Intrusiones

2.1. Qué son los IDS

Tal y como se ha visto en el capítulo anterior, la llegada de Internet y su extensión en todo tipo de ámbitos ha supuesto, además de un gran avance, la introducción de ciertos riesgos y vulnerabilidades, antes inexistentes. A día de hoy resulta inconcebible que una empresa de cierto tamaño no cuente con su propia red o utilice Internet para llevar a cabo gran parte de su actividad. Las facilidades que esta apertura al exterior puede proporcionar se contraponen con los ataques que estas organizaciones son susceptibles de sufrir. Pese a que existen soluciones que tratan de garantizar la seguridad y que únicamente los usuarios autorizados accedan a los recursos, dichas soluciones no resultan infalibles y dependen, además, de tareas de mantenimiento que están sujetas a fallos u olvidos. Es ahí donde entran en juego los IDS o sistemas de detección de intrusiones[13]. Una vez que el atacante ha traspasado las medidas de prevención resulta esencial detectarlo por las siguientes razones:

- Cuanto antes se localice la intrusión, antes se pueden tomar medidas al respecto y, por lo tanto, menor será el daño causado por el ataque.
- De la misma manera que un sistema de alarmas instalado en una casa puede disuadir a ladrones a la hora de perpetrar un robo, un IDS también puede frenar posibles ataques.
- La detección de ataques proporciona una gran cantidad de información sobre las estrategias empleadas por los atacantes, y contribuyen a solventar vulnerabilidades del sistema de prevención.

En un paso previo a la descripción de los sistemas de detección de intrusiones, a continuación se explican los riesgos y los distintos ataques que puede sufrir un sistema. Dejando al margen los virus, que junto con las intrusiones representan los ataques más comunes, podrían distinguirse distintos tipos de intrusos[10]:

- **Impostor**: en este caso el intruso entra en el sistema bajo la identidad de un usuario legitimo y hace uso de los recursos de dicha cuenta.
- **Usuario negligente**: un usuario legitimo del sistema utiliza de manera errónea o abusiva los recursos a los que tiene acceso, ya sean datos o programas.
- **Usuario clandestino**. Se trata de un atacante que intenta apropiarse de los privilegios del administrador o superusuario. Bajo la identidad del administrador el atacante puede, además, eludir controles de acceso o el registro de sus actividades.

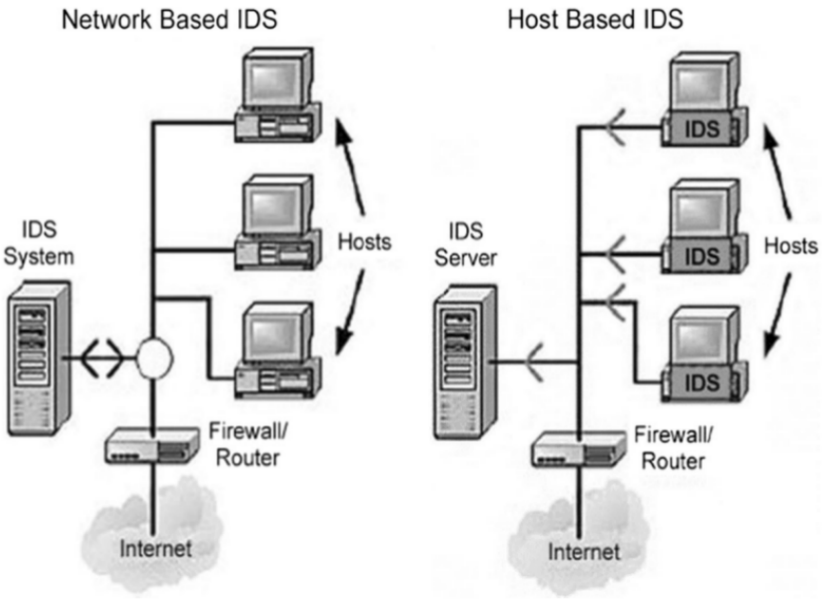

FIGURA 2.1: Comparación entre un HIDS y un NIDS

2.1.1. Tipos de IDS

Existen dos criterios para clasificar los IDS: según dónde se coloquen o según la técnica que empleen. Atendiendo al punto de la red en que se encuentre, hay:

- **IDSs basados en *host* o máquina (HIDS)**: se localizan a nivel de una única máquina, corriendo como aplicaciones, y analizan ficheros de log para detectar intrusiones.
- **IDSs basados en red (NIDS)**: este tipo de IDS se encuentra en puntos concretos de la red, ya sea a la entrada o en el área desmilitarizada *DMZ*. En este caso se analiza todo el tráfico de la red y no solamente el que llega a cierta máquina.
- **IDSs híbridos**: en este caso se combinan las dos modalidades previamente mencionadas.

Por otro, en función de la técnica que se utilice para detectar la intrusión, se encuentran los siguientes IDSs:

- **Detección por uso inadecuado o basada en conocimiento**: se buscan trazas de ataques, patrones en el tráfico de red o registros en logs que puedan denotar un comportamiento sospechoso. Este tipo de sistema reconocería como sospechoso un número elevado de intentos fallidos de acceso.
- **Detección por firma**: En este caso, se monitorizan los paquetes de la red, comparándolos con patrones predeterminados y reconocidos, que se conocen como firmas. Ante la detección de un nuevo ataque, expertos extraen características que se traducen en firmas y permitan su detección en futuras ocasiones. Este tipo de sistemas presenta como inconveniente el lapso de tiempo que transcurre entre que se detecta un ataque y la firma está disponible.
- **Detección por anomalía**: la instrusión es detectada cuando se identifica un comportamiento anómalo para determinado perfil o se superan los umbrales

fijados según análisis estadísticos. Algunos ejemplos de los ataques que se pueden detectar serían suplantación de la identidad o un ataque de denegación de servicio.

[6]

2.1.2. Características de un IDS

A la hora de utilizar un IDS es importante ser consciente de que se trata de una herramienta muy potente, pero pese a ello, presenta ciertas limitaciones. Es capaz, sin embargo de:

- Registrar la actividad de los usuarios.
- Comprobar las alteraciones y modificaciones que se producen en ciertos datos.
- Actualizarse para responder ante los últimos ataques.
- Detectar si el sistema está sufriendo un ataque.
- Localizar errores de configuración.
- Facilitar la creación de políticas de seguridad por parte del administrador.
- Realizar gestión de la seguridad.

Además, presenta un diseño dedicado a infraestructura, lo que supone una ventaja para implantarlo en un sistema. No obstante, tal y como se ha mencionado, resulta importante ser consciente de ciertos inconvenientes, como la falta de solución ante mecanismos de autenticación inseguros, la necesidad de una persona que investigue los posibles ataques detectados o los problemas que presenta a la hora de analizar el tráfico en una red, cuando este es muy elevado [6].
A continuación en las siguientes secciones se describiran dos de los IDS más utilizados a día de hoy, *Snort* y *Suricata*, discutiéndose también la elección del primero para el desarrollo del proyecto.

2.2. Snort

Snort se trata de un software gratuito y de código abierto desarrollado en 1998 por Martin Roesch. Desde su aparición, ha ido mejorando y extendiéndose, encontrándose consolidado como uno de los IDSs *open source* más potentes y eficaces. En la actualidad el proyecto se encuentra a cargo de la empresa *Sourcefire*, creada también por Martin Roesch, y que fue comprada por *Cisco* en 2013. Pese al desarrollo de una versión comercial, Sourcefire 3D System, Snort sigue adelante respaldado por una amplia comunidad.
En lo que respecta a las funcionalidades de Snort, es posible distinguir tres modos de funcionamiento distinto:

- **sniffer**: en este modo, Snort lee los paquetes que atraviesan la red y los muestra por pantalla.
- **Logger de paquetes**: loguea o registra los paquetes capturados en disco.
- **Network Intrusion Detection System**: Cuando funciona en este modo, Snort lleva a cabo tareas de análisis y detección del tráfico de la red. Este modo ofrece multitud de opciones de configuración, lo que puede suponer cierta dificultad.

[4]

2.2.1. Arquitectura de Snort

Las distintas funcionalidades descritas anteriormente se implementan a través de una serie de *plugins* que presenta la arquitectura modular del sistema. Snort consta de un núcleo o core con 4 módulos [8]:

- ***Sniffer* de paquetes**: este plugin permite escuchar el tráfico que viaja a través de la red, siendo incluso posible almacenar los paquetes capturados para su posterior lectura

- **Decodificador**: identifica los protocolos que encapsula el paquete desde el nivel de capa de enlace hasta los niveles TCP/IP. Los paquetes atraviesan por lo tanto una cascada de decodificadores hasta que su contenido queda guardado en estructuras de datos según sus correspondientes campos. De esta manera el contenido de los paquetes queda preparado para ser tratado por los preprocesadores.

- **Preprocesador**: El preprocesador de snort engloba una serie de *plugins* o módulos encargados de diversas tareas que facilitan y aceleran la detección en el siguiente módulo (motor de detección) al realizarse el *matching* con las reglas. Se puede variar el número de preprocesadores que los paquetes han de atravesar, variando con ello también el tiempo total de procesamiento, lo que resulta fundamental a la hora de determinar la eficacia de snort.
  Entre las funciones que pueden realizar los preprocesadores se encuentran [4]:

  - Detección de anomalías. Consiste en determinar si el contenido de un paquete se ajusta a lo que corresponde con los protocolos que lo encapsulan.
  - Agregación de sesiones TCP. Este preprocesador recoge los datos de una sesión TCP, agrupándolos de manera que posteriormente sean evaluados y analizados en su conjunto. Esto se debe a que gran parte de los ataques suelen llegar en distintos fragmentos, de manera que serían indetectables si fuesen estudiados por separado.
  - Ensamblado de fragmentos IP. De manera similar a lo que ocurría con las sesiones TCP, los paquetes IP pueden sufrir fragmentaciones debido a las limitaciones de la red, en concreto al MTU (*Maximum Transfer Unit*) que determina el tamaño máximo de un paquete para que pueda atravesar un enlace. De esta manera resulta posible que un ataque quede enmascarado en varios fragmentos y no genere ninguna alerta.
  - Detección de escaneo de puertos. Resulta muy difícil detectar un escaneo de puertos haciendo uso únicamente de reglas, pues hay que tener en cuenta que para realizarlo se envían paquetes a distintos hosts y puertos, en conexiones distintas. Por otro lado, existen ciertos paquetes que no cumplen las especificaciones y denotan que se esta llevando a cabo este tipo de ataque. Es el caso de un paquete *NULL*

- **Motor de detección**. Este módulo se encarga de procesar los paquetes procedentes del preprocesador. Para ello utiliza una serie de reglas contra las que analiza los paquetes. En caso de encajar, son enviados al procesador de alertas. El sistema de reglas empleado por Snort se basa en la detección mediante firmas o signatures. Este método se basa en comparar los datos de los paquetes, como cadenas, con ciertos patrones conocidos de ataques. A esta comparativa las reglas de snort añaden la posiblidad de generar expresiones de manera que

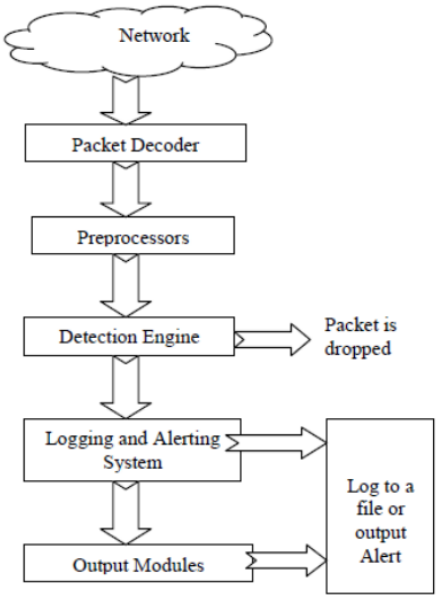

FIGURA 2.2: Arquitectura de snort

se produzca una coincidencia solo bajo determinadas circunstancias. Con todo ello, este sistema resulta extremadamente rápido (gracias a la detección mediante patrones) y fiable, pues el hecho de fijar determinados parámetros permite reducir el número de falsos positivos (alertas generadas por paquetes que no constituyen ningún ataque).

- **Módulo de alertas y logs**. Este componente se encarga de gestionar los paquetes que hayan coincidido con alguna regla. Existen multitud de posibilidades a la hora de tratar las alertas y logs generados. Por ejemplo, se pueden guardar las alertas en ficheros en máquinas remotas haciendo uso de sockets Unix o mostrar la información referente a los logs en interfaces web. Todo ello requiere el empleo de *plugins* adicionales, al igual que ocurría con los preprocesadores.

2.2.2. Reglas de Snort

Las reglas de snort constituyen la clave del sistema, encontrándose agrupadas en función del tipo de ataque que detectan. Existen ya numerosas reglas en el proyecto, que pueden añadirse a Snort con un simple *include* en el fichero de configuración. Pese a ello, la sintaxis simple y descriptiva del lenguaje que emplean estas reglas permite que cualquier usuario pueda escribir sus propias reglas, una vez se haya familiarizado con esta.
Las reglas constan de dos partes [9]:

- **Header o cabecera**. Esta primera parte contiene la acción a realizar en caso de que la regla coincida, así como, las direcciones IP y puertos, tanto de origen como destino.

- **Opciones**. Esta segunda parte, incluye el mensaje que ha de mostrarse en la alerta y otros parámetros adicionales, como las partes del paquete que han de evaluarse para determinar si se cumple la regla o no.

Bibliografía

[1] Herbert Bos. *Shelia: a client-side honeypot for attack detection*. URL: http://www.cs.vu.nl/%7Eherbertb/misc/shelia/ (visitado 10-03-2018).

[2] cisco. *Cisco 2017 Annual Cybersecurity Report*. Inf. téc. Cisco Systems, Inc., 2017.

[3] cisco. *What Is Network Security?* URL: https://www.cisco.com/c/en/us/products/security/what-is-network-security.html.

[4] Foster & Posluns. «Snort: The Inner Workings». En: *Snort intrusion detection 2.0*. 2003.

[5] Nmap Project. *SecTools.Org: Top 125 Network Security Tools*. 2018. URL: http://sectools.org/?sort=rank.

[6] Rajesh Vuppala, Mohammed Farik. «Intrusion Detection & Prevention Systems - Sourcefire Snort». En: *INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH* (2016).

[7] Jamie Riden. *Know Your Enemy: Malicious Web Servers*. 2008. URL: http://www.honeynet.org/papers/mws.

[8] Shimonski & Shinder. «Chapter 29 Introducing Snort». En: *The best damn firewall book period*. 2003.

[9] *SNORT Users Manual*.

[10] W Stallings. *Network Security Essentials*. Harlow, United Kingdom: Pearson Education Limited, 2016.

[11] TheHackerWay. *HoneyPots Parte 1 – Kippo*. 2015. URL: https://thehackerway.com/2015/03/24/honeypots-parte-1-kippo/.

[12] Verizon. *2017 Data Breach Investigations Report*. Inf. téc. VerizonEnterprise.

[13] Kemmerer & Vigna. «Intrusion detection: a brief history and overview». En: *Computer* (2002).