

Sistemas de Detección de Intrusiones

Diego González Gómez

Versión 1.0
Última revisión: Julio, 2003

Copyright (c) 2003 Diego González Gómez.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with the Invariant Sections being "Notas del Autor", and "Conclusiones", with the Front-Cover Text being "Sistemas de Detección de Intrusiones", and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Contenido general

Notas del autor	1
Introducción	3
1 Historia	5
2 Definiciones	13
3 Modelo de funcionamiento	23
4 Complementos y casos especiales	87
5 Capacidades y limitaciones	113
6 Implementación	119
7 Aspectos legales	127
8 Necesidades, líneas de trabajo	137
9 Futuro	149
10 Conclusiones	155
Apéndice A - Glosario de términos anglosajones	157
Apéndice B - Glosario	167
Apéndice C - Bibliografía	181
Apéndice D - Normativa legal	197
Apéndice E - Recursos	215
Apéndice F - Índice	223
GNU Free Documentation License	231
Licencia de Documentación Libre GNU (traducción)	239

Contenido

Notas del autor	1
Introducción	3
1 Historia	5
1.1 Auditorías: el comienzo	6
1.2 Los primeros Sistemas de Detección de Intrusiones	7
1.3 Unión de sistemas de detección basados en máquina y basados en red	9
1.4 Aparición de productos comerciales	10
1.5 Referencias	11
2 Definiciones	13
2.1 Términos de seguridad	13
2.1.1 Seguridad, dos puntos de vista	13
2.1.2 Confianza	14
2.1.3 Vulnerabilidad	14
2.1.4 Amenaza	14
2.1.5 Políticas de seguridad	14
2.1.6 Elementos de la infraestructura de seguridad	15
2.1.6.1 Control de acceso	16
2.1.6.2 Identificación y Autenticación	16
2.1.7 Cifrado	16
2.1.8 Cortafuegos	17
2.2 Motivos que originan problemas de seguridad	17
2.2.1 Diseño o desarrollo	17
2.2.2 Gestión	17
2.2.3 Confianza	18
2.3 Elementos de la detección de intrusiones	18
2.3.1 Arquitectura	18
2.3.2 Fuentes de datos, monitorización	18
2.3.3 Tipos de análisis	19
2.3.4 Respuestas	21
2.3.5 Clasificación general	21
2.3.6 Objetivos	21
2.3.7 Control	22
2.4 Referencias	22
3 Modelo de funcionamiento	23
3.1 Fuentes de información	23
3.1.1 Fuentes de información basadas en máquina	23
3.1.1.1 Registros de auditoría	23
3.1.1.2 Contenido de los registros de auditoría	24
3.1.1.3 El problema de la reducción de auditoría	28
3.1.1.4 Registro de Sistema	28
3.1.1.5 Registros de sistema comunes	29
3.1.1.6 Información de aplicaciones	30

3.1.1.7	Información recogida de objetivos	32
3.1.2	Fuentes de información basadas en red	33
3.1.2.1	Paquetes de red	34
3.1.2.2	Redes TCP/IP	35
3.1.2.3	Pila de protocolos	35
3.1.2.4	Estructura de una dirección IP	36
3.1.2.5	Estructuras de datos	37
3.1.2.6	Captura	38
3.1.2.7	Dispositivos de red	39
3.1.2.8	Fuentes de información externas	39
3.1.3	Información de productos de seguridad	39
3.1.3.1	Otros componentes como fuentes de datos	40
3.2	Análisis	41
3.2.1	Objetivos y elementos principales	41
3.2.1.1	Objetivos principales	42
3.2.1.2	Requisitos y objetivos secundarios	43
3.2.1.3	Factores de detección	43
3.2.2	Modelos	44
3.2.2.1	Construcción del analizador	45
3.2.2.2	Realización del análisis	47
3.2.2.3	Refinamiento y reestructuración	48
3.2.3	Técnicas	49
3.2.3.1	Detección de usos indebidos	49
3.2.3.2	Detección de anomalías	58
3.2.3.3	Métodos alternativos	69
3.3	Respuesta	75
3.3.1	Primeras consideraciones	75
3.3.2	Tipos de respuestas	76
3.3.2.1	Respuestas activas	76
3.3.2.2	Respuestas pasivas	78
3.3.3	Observaciones sobre las respuestas	78
3.3.3.1	Aspectos de seguridad	79
3.3.3.2	Falsas alarmas	79
3.3.3.3	Almacenamiento de registros	80
3.3.4	Adopción de políticas de respuesta	80
3.3.4.1	Intermedia o crítica	80
3.3.4.2	Oportuna	80
3.3.4.3	Largo plazo - local	81
3.3.4.4	Largo plazo - global	81
3.4	Referencias	81
4	Complementos y casos especiales	87
4.1	Escáner de vulnerabilidades	87
4.1.1	Proceso de análisis	87
4.1.2	Tipos de análisis de vulnerabilidades	88
4.1.2.1	Análisis de vulnerabilidades basado en máquina	88
4.1.2.2	Análisis de vulnerabilidades basado en red	89
4.1.2.3	"Password cracking"	91
4.1.2.4	Ventajas e inconvenientes	91
4.2	"Honeypot", "HoneyNet" y "Padded Cell"	92

4.2.1	"Honeypot"	92
4.2.1.1	Ventajas e inconvenientes	93
4.2.2	"Honeynet"	95
4.2.2.1	GenI	95
4.2.2.2	GenII	96
4.2.2.3	"Honeynet" virtual	97
4.2.2.4	Ventajas e inconvenientes	100
4.2.3	"Padded Cell"	101
4.2.3.1	Ventajas e inconvenientes	102
4.3	Verificador de integridad de ficheros	102
4.4	Cortafuegos: Prevención de Intrusiones	102
4.4.1	IDS basado en red, en modo "in-line"	103
4.4.2	Conmutador de nivel siete	105
4.4.3	Cortafuegos/IDS de aplicación	106
4.4.4	Conmutador híbrido	107
4.4.5	Aplicación engañosa	108
4.5	Referencias	109
5	Capacidades y limitaciones	113
5.1	Capacidades	113
5.2	Limitaciones	114
5.3	Referencias	117
6	Implementación	119
6.1	Sistemas de Detección de Intrusiones de red	119
6.1.1	Delante del cortafuegos externo	120
6.1.2	Detrás del cortafuegos externo	121
6.1.3	Redes principales	122
6.1.4	Subredes de valor crítico	122
6.1.5	Máquinas	123
6.2	Sistemas de Detección de Intrusiones de máquina	124
6.3	Alarmas	125
6.4	Referencias	125
7	Aspectos legales	127
7.1	Sistemas legales en el mundo	127
7.1.1	Derecho civil	128
7.1.2	"Common law"	129
7.1.3	Derecho consuetudinario	129
7.1.4	Derecho Musulmán, Derecho Talmúdico	129
7.1.5	Derecho Mixto	129
7.1.6	Territorios no independientes	129
7.2	Otros sistemas	130
7.2.1	Derecho penal	130
7.2.2	Derecho procesal	130
7.3	Situación en Europa	130
7.4	Situación en España	131
7.4.1	Legislación	131
7.4.1.1	Delitos informáticos y el Código Penal	131
7.4.1.2	Delitos informáticos y el C.N.P.	133

7.4.1.3	Legislación adicional	134
7.4.2	Intrusiones y la Legislación española	134
7.4.3	Cuerpos especiales	135
7.4.4	Necesidades y deficiencias	135
7.5	Referencias	136
8	Necesidades, líneas de trabajo	137
8.1	Normalización	137
8.1.1	CIDF	138
8.1.2	CRISIS	138
8.1.3	Formato de los datos de auditoría	138
8.1.3.1	Libro Naranja, Libro Marrón	138
8.1.3.2	IDES de Denning	139
8.1.3.3	SVR 4++ de Smaha	139
8.1.3.4	Bishop	139
8.1.3.5	IETF/IDWG	139
8.1.3.6	Mecanismos de auditoría	139
8.2	Integración	140
8.3	Escalabilidad	140
8.3.1	Tiempo	140
8.3.2	Espacio	140
8.3.2.1	GrIDS	141
8.4	Administración y gestión	141
8.5	Análisis	142
8.5.1	Detectores basados en inteligencia artificial	142
8.5.2	Falsas alarmas	143
8.5.3	Políticas de sistema	143
8.6	Fiabilidad	144
8.6.1	Fuentes de información	144
8.6.2	Análisis	144
8.6.3	Respuesta	145
8.6.4	Comunicaciones	146
8.7	Interfaz de usuario	146
8.8	Referencias	147
9	Futuro	149
9.1	Trayectoria recorrida	149
9.2	Perspectivas de futuro	149
9.2.1	Deficiencias y necesidades	150
9.2.1.1	Falsos positivos	150
9.2.1.2	Estandarización	150
9.2.1.3	Encriptación	151
9.2.1.4	Nuevos protocolos	151
9.2.1.5	Escalabilidad	151
9.2.1.6	Detección de anomalías	152
9.2.2	Correlación, composición	152
9.3	Referencias	154
10	Conclusiones	155

Apéndice A - Glosario de términos anglosajones	157
Apéndice B - Glosario	167
Apéndice C - Bibliografía	181
Apéndice D - Normativa legal	197
Marco general	197
Intrusiones, ataques	198
Ley Orgánica 10/1995, de 23 de noviembre, del Código Penal	199
Ley 1/2000, de 7 de enero, de Enjuiciamiento Civil	204
Ley de 14 de septiembre de 1882, de Enjuiciamiento Criminal.	211
Apéndice E - Recursos	215
Libros	215
Recursos WWW	217
Apéndice F - Índice	223
GNU Free Documentation License	231
Licencia de Documentación Libre GNU (traducción)	239

Índice de figuras

Figura 1-1 - Sistema de Auditorías Básico	5
Figura 1-2 - Sistema de Detección de Intrusiones Distribuido (DIDS)	10
Figura 2-1 - Esquema general de un Sistema de Detección de Intrusiones	20
Figura 2-2 -Tipos principales de IDS	21
Figura 3-1 - Estructura de los registros de auditoría BSM	25
Figura 3-2 - Estructura de un informe ("record") de auditoría BSM	26
Figura 3-3 - Dispositivo de escucha de red y cable de sólo recepción	34
Figura 3-4 - Escenarios de conexión de un rastreador	35
Figura 3-5 - Modelo OSI y Arquitectura TCP/IP	36
Figura 3-6 - Cabecera de un Datagrama Internet	37
Figura 3-7 - Formato de la cabecera de TCP	37
Figura 3-8 - Diagrama general de un modelo de gestión de seguridad	42
Figura 3-9 - Actividades de un sistema: usos indebidos y anomalías	45
Figura 3-10 - Modelo general de un detector de usos indebidos	49
Figura 3-11 - Ejemplo de diagrama if-then-else	50
Figura 3-12 - Estructura de Bro	53
Figura 3-13 - Diagrama de transiciones de estados	54
Figura 3-14 - Modelo Petri-net	56
Figura 3-15 - Ejemplo de patrón de ataque mediante el lenguaje Petri-net	57
Figura 3-16 - Arquitectura del sistema STALKER	58
Figura 3-17 - Modelo general de un detector de anomalías	59
Figura 3-18 - Ataques que detecta un filtro de anomalías de protocolo	61
Figura 3-19 - Generación de patrones probables	66
Figura 3-20 - Redes neuronales para la detección de anomalías	68
Figura 3-21 - Arquitectura del sistema AAFID	73
Figura 4-1 - Clasificación de analizadores de vulnerabilidades	88
Figura 4-2- Arquitectura de Nessus	90
Figura 4-3 - Ejemplo de un "Honeypot" (Sistema trampa)	93
Figura 4-4 - Arquitectura "Honeynet" GenI	96
Figura 4-5 - Arquitectura "Honeynet" GenII	97
Figura 4-6 - "Honeynet" virtual auto-contenida	99
Figura 4-7 - "Honeynet" virtual híbrida	100

Figura 4-8 - Procedimiento general de "Bait and Switch"	101
Figura 4-9 - NIDS en modo de escucha ("Tap mode")	104
Figura 4-10 - NIDS en modo en línea ("In-line mode")	104
Figura 4-11 - Procedimiento general de "Hogwash"	105
Figura 4-12 - Procedimiento general de un conmutador de nivel siete	106
Figura 4-13 - Cortafuegos/IDS de aplicación	107
Figura 4-14 - Procedimiento general de un conmutador híbrido	108
Figura 4-15 - IPS basado en aplicación engañosa ("deceptive application")	109
Figura 6-1 - Situaciones de implementación de un IDS	120
Figura 6-2 - Implementación progresiva de HIDS	124
Figura 7-1 - Sistemas legales en el mundo	128
Figura 9-1 - Fuentes de información comunes	153

Índice de tablas

Tabla 2-1 - Ejemplo de política de seguridad procesal	15
Tabla 3-1 - Permisos en Windows NT y Windows 2000	27
Tabla 3-2 - Registros relativos a seguridad en Solaris	29
Tabla 3-3 - Estructura de registros CLF	31
Tabla 3-4 - Contenido de un inodo del sistema de ficheros UNIX (System V)	33
Tabla 3-5 - Redes y máquinas estándar en Internet	36
Tabla 3-6 - Formato de registro FW-1	40
Tabla 3-7 - Clasificación de medidas, con ejemplos, de IDES	47
Tabla 3-8 - Plataformas soportadas por Snort	51

Notas del autor

El presente documento, fruto de mi proyecto de fin de carrera, pretende describir los aspectos más relevantes relacionados con los Sistemas de Detección de Intrusiones (IDS), tales como su historia, funcionamiento, ventajas e inconvenientes, implementación, aspectos legales, o su futuro. Por esta razón, la labor de documentación es fundamental. Para realizar esta tarea con el máximo rigor, he optado por utilizar las normas ISO existentes en materia de referencias bibliográficas, que en concreto son la ISO 690:1987 [1] (equivalente a la norma UNE 50-104-94) e ISO 690-2 [2]. En lo referente a términos en inglés, he procurado utilizar en la medida de lo posible sus equivalentes en castellano. He dejado los vocablos más comunes en su idioma original, seguidos de una breve descripción en castellano. Con este fin, he acudido y consultado a diversos profesionales, libros como el *Diccionario Comentado de Terminología Informática* [3], el *Diccionario de ideas afines* [4] y glosarios como ORCA (Glosario de Informática Inglés-Español) [5], y el *Glosario básico inglés-español para usuarios de Internet* [6].

Por otra parte, me gustaría agradecer la colaboración desinteresada de diversos profesionales y compañeros durante el transcurso del proyecto. En especial a Inmaculada Álvarez, Javier de la Cueva y Lance Spitzner, por sus valiosas sugerencias y el tiempo que han dedicado a resolver pacientemente mis dudas. También deseo mostrar mi agradecimiento a mis familiares y amigos por su constante interés y apoyo, sin los cuales, este documento no habría sido posible.

Diego González Gómez
dggomez@users.sourceforge.net
<http://www.dgonzalez.net>

- [1] ISO 690:1987, *Information and documentation -- Bibliographic references -- Content, form and structure*.
- [2] ISO 690-2, *Information and documentation -- Bibliographic references -- Part 2: Electronic documents of parts thereof*.
- [3] Aguado de Cea, Guadalupe. *Diccionario Comentado de Terminología Informática*. Madrid: Paraninfo, 1983.
- [4] Corripio Pérez, Fernando. *Diccionario de ideas afines*. Barcelona: Herder, S.A. , agosto de 2000.
- [5] Villate, Jaime E., *ORCA - Glosario de Informática Inglés-Español*. [en línea]. 3 de Diciembre, 2002 [consultado en febrero, 2003]. Versión número 2.1.160. Última versión disponible en diferentes formatos en <<http://quark.fe.up.pt/orca/>>.
- [6] Fernández Calvo, Rafael, *Glosario básico inglés-español para usuarios de Internet*. [en línea]. Julio, 2001. Cuarta edición [consultado en febrero, 2003]. Disponible en formato ASCII en <http://www.ati.es/novatica/glosario/glosario_internet.txt>.

Introducción

Desde su invención hasta nuestros días, el número de ordenadores ha crecido hasta consolidarse como un instrumento casi imprescindible en la vida cotidiana del hombre. Su versatilidad, potencia de cálculo y cada vez más fácil manejo hacen de ellos una herramienta muy importante en gran variedad de actividades, desde la científica a la lúdica.

Con la posibilidad de interconectar múltiples ordenadores formando redes, surgieron nuevos retos y aplicaciones. Es difícil imaginarse hoy algún banco, hospital, o gran superficie comercial en un país desarrollado, que no mantenga los datos de sus clientes o hagan sus transacciones de forma electrónica. Hoy en día los bancos hacen uso de redes para efectuar sus operaciones financieras, los hospitales tienen los historiales de sus pacientes en bases de datos, y muchos comercios están presentes en Internet, de forma que cualquier usuario del planeta puede tanto escoger el producto que desea como pagarlo a través de la red. Los datos que manejan este tipo de empresas deben mantenerse a salvo de cualquier intruso a toda costa. La seguridad en este tipo de empresas tiene una importancia crítica.

La red ARPAnet, creada por el gobierno estadounidense en 1969 para actividades de desarrollo y defensa, sería la precursora de la que hoy conocemos como Internet. En aquel entorno, la seguridad era mínima. Se trataba de una red compuesta por una pequeña comunidad cuyos miembros eran de confianza. La mayoría de los datos que se intercambiaban no eran confidenciales, y muchos usuarios se conocían.

Por el contrario, las redes globales sí requieren un mayor nivel de seguridad. Manejan notables volúmenes de información, atendiendo de forma independiente operaciones de distintos países que en muchas ocasiones intercambian datos privados. Es común la existencia de servidores que diariamente reciben algún tipo de ataque con diversos fines, desde detener sus servicios hasta obtener algún tipo de dato confidencial.

En este marco, las necesidades de seguridad son importantes. Se debe mantener un no siempre fácil equilibrio entre recursos utilizados y privacidad requerida. También debería ser lo suficientemente flexible para cumplir con los requisitos necesarios para permitir el seguimiento de los culpables a través de distintas jurisdicciones. El abanico de medidas y recursos técnicos necesarios para establecer el escenario adecuado de seguridad es muy amplio, y los expertos lo saben.

El área de las auditorías de seguridad y detección de intrusiones viene siendo más indispensable cada día. Estas tecnologías no sólo identifican y rastrean intrusiones, sino que mejoran la estabilidad y confianza de otros mecanismos de seguridad del sistema que monitorizan.

La detección de intrusiones es el proceso de monitorizar los eventos que ocurren en un sistema o red, para analizarlos en busca de problemas de seguridad. Este término es aplicable a distintas actividades. Existen alarmas de ladrones, o cámaras de vigilancia usadas por bancos o comercios. También se podría afirmar que la defensa militar también entra en esta categoría. Aunque todas ellas pertenezcan a distintas áreas, tienen características comunes: todas tienen funciones de vigilancia, alarma y emiten alarmas cuando un determinado suceso tiene lugar.

Esta tecnología es relativamente joven, surgiendo en los años 80. Desde entonces, han aparecido una enorme variedad de propuestas que intentan dar solución a este enfoque, tan complicado como rico en posibilidades.

Capítulo 1

Historia

*"Todo tiempo pasado, fue anterior."
Les Luthiers*

Cuando la gente oye hablar de Sistemas de Detección de Intrusiones (IDS), generalmente los asocia a "alarmas de ladrones para ordenadores o redes". Es fácil entender un concepto como este usando comparaciones sencillas. En realidad, la explicación es bastante aproximada, y los usuarios que no se dedican a la seguridad no necesitan saber más. Sin embargo, los expertos en seguridad no pueden cometer el error de conformarse con algo tan trivial, sin tener conocimiento alguno sobre la historia de estos sistemas.

La detección de intrusiones es el fruto de la aplicación del Procesamiento Electrónico de Datos (EDP) a las auditorías de seguridad, utilizando mecanismos de identificación de patrones y métodos estadísticos. Es una parte imprescindible en las modernas tecnologías de seguridad de redes.

Antes de la detección de intrusiones existían las auditorías de seguridad. La auditoría es el proceso de generar, almacenar y revisar eventos de un sistema cronológicamente.

La Figura 1-1 muestra un esquema simple del funcionamiento de un sistema de auditorías. Los eventos de sistema son capturados por los generadores de auditorías, que llevan los datos al elemento encargado de guardarlos en un fichero de "logs". El analizador, en base a unas políticas de seguridad, emite los resultados a través de un terminal.

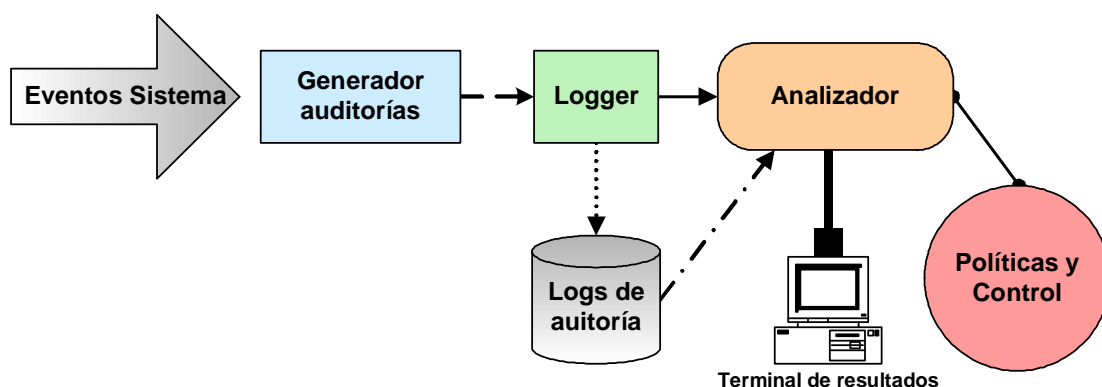


Figura 1-1 - Sistema de Auditorías Básico

Durante los comienzos de la historia de los ordenadores, estas máquinas eran relativamente escasas y muy caras. Su uso estaba restringido a técnicos e ingenieros especializados. Los primeros sistemas de auditorías tenían como propósito medir el tiempo que dedicaban los

operadores a usar los sistemas que monitorizaban, con una precisión de milésimas de segundo, y servían entre otras cosas para poder facturarles el mismo.

1.1 Auditorías: el comienzo

A mediados de los años 50 la empresa "Bell Telephone System", de Estados Unidos, creó un grupo de desarrollo con el objetivo de analizar el uso de los ordenadores en el futuro en el negocio de las empresas de telefonía. Este equipo estableció la necesidad de utilizar auditorías mediante el Procesamiento Electrónico de Datos (EDP), rompiendo con el anterior sistema basado en los tradicionales informes de papel. Esto hizo que a finales de los 50 la "Bell Telephone System" se embarcara en el primer sistema a gran escala de facturación telefónica controlada por ordenadores. [1]

El Departamento de Defensa de EEUU empleó numerosos recursos en los años 70 para la investigación de políticas de seguridad, directrices y pautas de control de lo que denominaban "sistemas de confianza". Estos esfuerzos culminaron con la Iniciativa de Seguridad de 1977.

Los Sistemas de Confianza son aquellos "sistemas que emplean los suficientes recursos hardware y software para permitir el procesamiento simultáneo de una variedad de información confidencial o clasificada". [2]

En estos sistemas se albergaban distintos tipos de información repartida en niveles, que correspondían a su grado de confidencialidad.

En un principio, los desarrolladores no tenían claro si las auditorías jugaban un papel importante en la seguridad de un sistema de confianza. Más tarde, se terminó incluyendo un apartado sobre los mecanismos de las auditorías en el "Trusted Computer System Evaluation Criteria" o TSCSEC [3] (Libro Naranja), como un requisito para cualquier sistema de confianza de clase C2 o superior. La serie de documentos del Departamento de Defensa de EEUU sobre Sistemas de Confianza se conoce como la "serie Arco Iris" ("Rainbow series") debido a los colores de las tapas de los libros que publicaban.

El documento que trata el tema de las auditorías está incluido en el "Libro Marrón" titulado "A Guide to Understanding Audit in Trusted Systems" [4]. En este libro se enumeran los cinco objetivos de un mecanismo de auditoría:

- Permitir la revisión de patrones de acceso (por parte de un objeto o por parte de un usuario) y el uso de mecanismos de protección del sistema.
- Permitir el descubrimiento tanto de intentos internos como externos de burlar los mecanismos de protección.
- Permitir el descubrimiento de la transición de usuario cuando pasa de un menor nivel de privilegios a otro mayor (elevación de privilegios).
- Permitir el bloqueo de los intentos de los usuarios por saltarse los mecanismos de protección del sistema.

- Servir además como una garantía frente a los usuarios de que toda la información que se recoja sobre ataques e intrusiones será suficiente para controlar los posibles daños ocasionados en el sistema.

1.2 Los primeros Sistemas de Detección de Intrusiones

A medida que el número de ordenadores crecía, el número de eventos de sistema a analizar era tal que esta tarea se volvía humanamente imposible. Las autoridades militares de norteamericanas se dieron cuenta de que el uso cada vez más masivo de ordenadores en sus instalaciones requería algún mecanismo que facilitara la labor de los auditores.

James P. Anderson fue la primera persona capaz de documentar la necesidad de un mecanismo que automatizara la revisión de los eventos de seguridad. Describió el concepto de "Monitor de Referencias" en un estudio encargado por las Fuerzas Aéreas de EEUU, y redactó un informe en 1980 que sería el primero de los futuros trabajos sobre detección de intrusiones. Uno de los objetivos de este informe era la eliminación de información redundante o irrelevante en los registros de sucesos¹. [5]

Anderson propuso un sistema de clasificación que distinguía entre ataques internos y externos, basado en si los usuarios tenían permiso de acceso o no al ordenador. Estos eran los principales objetivos de los mecanismos de auditoría de seguridad:

- Debían proporcionar suficiente información para que los encargados de seguridad localizaran el problema, pero no para efectuar un ataque.
- Debía ser capaz de obtener datos de distintos recursos de sistema.
- Para evitar ataques internos, debía detectar usos indebidos ("misuse") o fuera de lo normal por parte de los usuarios (o recursos).
- El diseño del mecanismo de auditoría debía ser capaz de obtener la estrategia usada por el atacante para entrar en las cuentas.

Ideó un sistema para dar solución al problema de los intrusos que se habían apoderado de cuentas de usuario legítimas. El cual debía distinguir entre el comportamiento normal o inusual de las cuentas basándose en patrones de uso, creados a partir del análisis de estadísticas de comportamiento de usuario. Los sistemas posteriores trabajarían esta idea. [6]

El "Intrusion Detection Expert System" (IDES), desarrollado entre 1984 y 1986 por Dorothy Denning y Peter Neumann, fue un modelo que definía un sistema de detección de intrusiones en tiempo real [7]. Este proyecto, fundado entre otros por la Marina estadounidense proponía una correspondencia entre actividad anómala y abuso, o uso indebido. Entendiendo por anómala, aquella actividad rara o inusual en un contexto estadístico. Usaba perfiles para describir a los sujetos del sistema (principalmente usuarios), y reglas de actividad para definir las acciones que

¹ Este método es conocido en inglés como "audit reduction".

tenían lugar (eventos de sistema o tiempos de CPU). Estos elementos permitían establecer mediante métodos estadísticos las pautas de comportamiento necesarias para detectar posibles anomalías. IDES era un sistema híbrido porque añadía un nivel de seguridad adicional mediante el uso de un sistema experto, basado en reglas de seguridad, que minimizaba los efectos de un intruso que intentara eludir el detector de anomalías.

Este modelo fue usado para el prototipo del sistema IDES por "SRI International", en el que trabajaron Teresa Lunt, R Jagganathan, Peter Neumann, Harold Javitz, y Fred Gilham. [8]

En los años ochenta aparecieron numerosos sistemas de detección de intrusiones. Desde 1984 hasta 1985 un grupo de desarrollo en Sytek dirigió un proyecto denominado "Automated Audit Analysis". Utilizaba información recogida a nivel de interfaz de comandos ("shell") de un sistema UNIX, para posteriormente compararlos con una base de datos. Estos datos se analizaban estadísticamente para demostrar que se podían detectar comportamientos fuera de lo normal. Algunos investigadores del proyecto trabajaron más tarde en "SRI International".

Discovery fue un sistema creado para detectar e impedir problemas en la base de datos de TRW. La novedad de Discovery radicaba en que monitorizaba una aplicación, no un sistema operativo. Utilizaba métodos estadísticos escritos en COBOL para detectar los posibles abusos. Su creador fue William Tener. [9]

El proyecto Haystack, del Centro de Soporte Criptológico las Fuerzas Aéreas de EEUU fue usado para ayudar a los oficiales a encontrar signos de ataques internos en los ordenadores principales de sus bases. Estas máquinas eran principalmente "mainframes" (servidores corporativos) que manejaban información no clasificada pero confidencial. El sistema estaba escrito en C ANSI y SQL. Examinaba los datos de forma periódica, recogiendo colas de eventos de forma periódica. Utilizaba dos fases de análisis para detectar las posibles anomalías. El principal responsable del proyecto fue Steve Smaha.[10]

Otro proyecto importante fue el "Multics Intrusion Detection and Alerting System" (MIDAS), creado por el National Computer Security Center (NCSC). Se utilizó para monitorizar el sistema NCSC's Dockmaster, un Honeywell DPS 8/70 en el que corría uno de los sistemas operativos más seguros de entonces, un Multics. Al igual que IDES, MIDAS utilizaba un sistema híbrido en el que combinaba tanto la estadística de anomalías como reglas de seguridad de un sistema experto.

MIDAS usaba un proceso de análisis progresivo compuesto por cuatro niveles de reglas. Además de estas reglas, también contaba con una base de datos que usaba para determinar signos de comportamiento atípico. Fue uno de los primeros sistemas de detección de intrusiones conectados a Internet. Fue publicado en la red en 1989 y monitorizó el Mainframe Dockmaster en 1990. Contribuyó a fortalecer los mecanismos de autenticación de usuarios. Además, no sólo había contribuido a mejorar la seguridad contra ataques externos, sino que también seguía bloqueando intrusiones internas. [11]

"Network Audit Director and Intrusion Reporter" (NADIR) fue desarrollado en Laboratorio Nacional de Los Alamos, para monitorizar el "Integrated Computing Network" (ICN). Esta red estaba inicialmente compuesta por unos 9.000 usuarios. NADIR usaba técnicas de detección similares a los sistemas de su tiempo como el IDES o MIDAS. Fue uno de los sistemas con más éxito de los años ochenta. La principal responsable de NADIR fue Kathleen Jackson.

El "Network System Monitor" (NSM) fue desarrollado en la Universidad de California para trabajar en una estación UNIX de Sun. Fue el primer sistema de detección de intrusiones que monitorizaba el tráfico de red, utilizando los datos del propio tráfico como principal fuente de datos. Los anteriores sistemas utilizaban los eventos de sistema o registraban las pulsaciones de teclado. El funcionamiento del NSM, que muchos sistemas de detección de intrusiones de red utilizan hoy en día, se puede describir en estos pasos:

- Ponía el dispositivo de red en modo promiscuo ("promiscuous mode"), de forma que monitorizara todo el tráfico que recibiera, incluido el que no iba dirigido al sistema.
- Capturaba los paquetes de red.
- Identificaba el protocolo utilizado para poder extraer los datos necesarios (IP, ICMP, etc.).
- Utilizaba un enfoque basado en matrices para archivar y analizar las características de los datos, en busca tanto de variaciones estadísticas que revelaran un comportamiento anómalo como de violaciones de reglas ya preestablecidas.

Una de las pruebas que se hicieron con el NSM duró dos meses. Monitorizó más de 111.000 conexiones, y detectó correctamente más de 300 posibles intrusiones. Como dato significativo, y para enfatizar la necesidad del uso de este tipo de sistemas, hay que señalar que los administradores no llegaron a percibir ni el 1% de dichas intrusiones.

Los principales responsables del NSM fueron Karl Levitt, Todd Heberlein, y Biswanath Mukherjee de la Universidad de California. [12]

El sistema "Wisdom and Sense" fue un detector de anomalías creado en el Laboratorio Nacional de Los Alamos en cooperación con el Laboratorio Nacional de Oak Ridge. Utilizaba técnicas no paramétricas ("nonparametric techniques"), que eran técnicas estadísticas que no hacían suposiciones sobre la distribución de los datos. Usaban este método para crear su propio conjunto de reglas. Luego analizaba los "logs" de las auditorías en busca de excepciones de esas reglas, las cuales estaban organizadas en "arrays" con forma de árbol. Definían lo que era el comportamiento normal desde un punto de vista cronológico de los datos de auditoría. [13]

1.3 Unión de sistemas de detección basados en máquina y basados en red

Como hemos visto, durante los comienzos de la detección de intrusiones, la mayoría de los sistemas estaban pensados para monitorizar "hosts" (máquinas). Salvo algunas excepciones como el mencionado NSM, que empezó a utilizar el tráfico de red como objetivo de vigilancia.

A partir de los años 90, el rápido crecimiento de las redes de ordenadores hizo que surgieran nuevos modelos de detección de intrusiones. Los daños provocados por el famoso gusano de Internet en 1988, ayudaron a que unieran esfuerzos las actividades comerciales y académicas en busca de soluciones de seguridad en este campo [14]. El primer paso para la fusión de sistemas de detección basados en máquina y red fue el "Distributed Intrusion Detection System" (DIDS).

El DIDS fue fruto del esfuerzo y la colaboración de grandes entidades como el Centro de Soporte Criptológico de las Fuerzas Aéreas de EEUU, El Laboratorio Nacional de Lawrence

Livermore, la Universidad de California y los Laboratorios Haystack. Fue el primer sistema capaz de hacer que un grupo de seguridad pudiera monitorizar las violaciones e intrusiones de seguridad a través de las redes. El principal responsable de este proyecto fue Steve Smaha. [15]

El objetivo inicial del DIDS era proporcionar medios que permitieran centralizar el control y publicación de resultados en un controlador central.

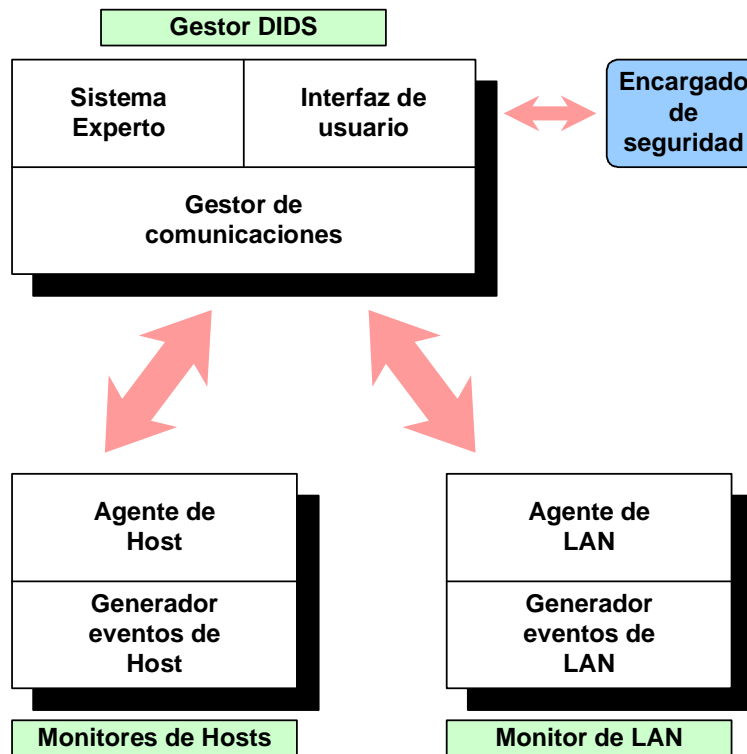


Figura 1-2 - Sistema de Detección de Intrusiones Distribuido (DIDS)

El DIDS afrontó diversos problemas. Los más importantes estaban relacionados con el hecho de tener que registrar eventos asociados a distintas máquinas a lo largo de la red. Un atacante suele aprovecharse de las redes para distribuir sus ataques, realizando éstos desde distintas máquinas. El DIDS fue el primer sistema capaz de relacionar los eventos que recibía para poder detectar una posible intrusión. Además, reunía la información de forma que era posible hacer un seguimiento del posible intruso. Esto permitía su uso para poder ser identificado y perseguido por la ley.

Para solventar el problema de relacionar los eventos que tenían lugar en los distintos niveles de abstracción de la red, el DIDS utilizaba un modelo de detección de intrusiones estratificado en seis niveles que distinguían los distintos tipos de datos.

1.4 Aparición de productos comerciales

Alrededor de 1990, tuvo lugar la aparición de los primeros programas de detección de intrusiones para uso comercial. Algunas empresas los desarrollaban para ocupar una posición destacada en el ámbito de la seguridad, y otras para mejorar los niveles exigidos por la NCSC.

Entre los productos más famosos de aquella época cabe mencionar el "Computer Watch" desarrollado por la empresa AT&T, el "Information Security Officer's Assistant" (ISOA) de PRC y el "Clyde VAX Audit" por Clyde Digital (luego RAXCO, y más tarde Axent).

1.5 Referencias

- [1] Wasserman, Joseph J. *The Vanishing Trail*. Bell Telephone Magazine 47, no. 4, July - August 1968: 12 - 15.
- [2] National Computer Security Center. *Glossary of Computer Security Terms*. Versión 1, Rainbow Series, octubre 1988.
- [3] National Computer Security Center. *Department of Defense Trusted Computer System Evaluation Criteria*. Orange Book, DOD 5200.28-std, December 1985.
- [4] National Computer Security Center. *A Guide to Understanding audit in Trusted Systems*. Versión 2, June 1988.
- [5] Anderson, James, P. *Computer Security Technology Planning Study*. ESD-TR-73-51, v II. Electronic Systems Division, Air Force Systems Command, Hanscom Filed, Bedford, MA, octubre 1972.
- [6] Anderson, James P. *Computer Security Threat Monitoring and Surveillance*. Fort Washington, PA: James P. Anderson Co., 1980.
- [7] Denning, Dorothy E. *An Intrusion Detection Model*. Proceedings of the 1986 IEEE Symposium on Security and Privacy, Oakland, CA, April 1986.
- [8] SRI International. *System Design Laboratory Laboratory - Intrusion Detection*. [en línea]. Fecha no disponible [consultado en enero, 2003]. Next-Generation IDES (NIDES). Disponible desde Internet <<http://www.sdl.sri.com/programs/intrusion/history.html>>.
- [9] Tener, William T. *Discovery: An Expert System in the Commercial Data Security Environment*. Proceedings of the IFIP Security Conference, Monte Carlo, 1986.
- [10] Smaha Steve E. *An Intrusion Detection System for the Air Force*. Proceedings of the Fourth Aurospace Computer Security Applications Conference, Orlando, FL, December 1988.
- [11] Sebring, Michael M., E. Shellhouse, M. E. Hanna, and R. A. Whitehurst. *Expert Systems in Intrusion Detection: A Case Study*. Proceedings of the Eleventh National Computer Security Conference, Washington, DC, October 1988.
- [12] Heberlein, Todd. *Network Security Monitor (NSM) - Final Report*. Lawrence Livermore National Laboratory, Davis, CA, February 1995.
- [13] Vaccaro, Henry S. and G. E. Liepins. *Detection of Anomalous Computer Session Activity*. Proceedings of the 1989 IEEE Symposium on Security and Privacy, Oakland, CA, May 1989.
- [14] Spafford, Eugene H. *The Internet Worm: Crisis and Aftermath*; Communications of the ACM; 32(6): 678 - 687, June 1989.
- [15] Snapp, S.R. et al. *DIDS (Distributed Intrusion Detection System) – Motivation, Architecture, and An Early Prototype*. Proceedings of the Fifteenth National Computer Security Conference, Baltimore, MD, October 1992.

Capítulo 2

Definiciones

La detección de intrusiones es el proceso de monitorizar redes de ordenadores y sistemas en busca de violaciones de políticas de seguridad [1]. Los sistemas de detección de intrusiones están compuestos por tres elementos funcionales básicos:

- Una fuente de información que proporciona eventos de sistema.
- Un motor de análisis que busca evidencias de intrusiones.
- Un mecanismo de respuesta que actúa según los resultados del motor de análisis.

La detección de intrusiones es la evolución de las auditorías tradicionales. El término auditar, en inglés "audit", y este del latín "audire" (oír), se define como "Examinar la gestión económica de una entidad a fin de comprobar si se ajusta a lo establecido por ley o costumbre". [2]

En términos de seguridad informática, llevar a cabo la auditoría de un sistema significa examinar y analizar el rastro de auditoría ("audit trail") que genera el sistema operativo y otros elementos del sistema. La revisión de los eventos se llevaba a cabo entre otros motivos para asegurarse de que no se violaban una serie de políticas de seguridad. Cuando se encontraba alguna irregularidad, surgían nuevos elementos básicos que cubrir:

- Responsabilidades: Encontrar el responsable de provocar la violación.
- Evaluación de daños: Verificar los problemas provocados en el sistema y de qué forma fueron realizados.
- Recuperación: Qué acciones son necesarias para recuperar el estado normal.

A medida que las máquinas se fueron haciendo más rápidas y complejas, el número de sucesos a analizar era tal que no podía llevarse a cabo de la manera tradicional. Por esta razón se desarrollaron mecanismos cada vez más eficaces para simplificar la labor de los auditores de sistemas. Los primeros sistemas que se encargaban de esta labor utilizaban soluciones basadas en técnicas de reducción de eventos y patrones estadísticos. Este aspecto está explicado más a fondo en el capítulo 1.

2.1 Términos de seguridad

Antes de hacer que un sistema o red sea seguro, primero es necesario definir lo que se entiende por términos como seguridad, confianza, vulnerabilidad, etc.

2.1.1 Seguridad, dos puntos de vista

La seguridad se puede entender desde dos puntos de vista; el práctico y el formal.

Desde una perspectiva práctica, un sistema seguro es "aquel con que se cuenta que actúe de la manera esperada". Este punto de vista tiene unas explícitas implicaciones de confianza. Pero la confianza no se puede medir. No podemos confiar en que un sistema se comporte como debe. Nadie nos puede asegurar que un sistema se está comportando como realmente tiene que hacerlo.

Según el enfoque formal, más preciso, la seguridad se define a través de una "tríada de conceptos": *confidencialidad, integridad y disponibilidad*.

La **confidencialidad** implica que la información sea accedida exclusivamente por el personal autorizado a la misma.

La **integridad** consiste en la necesidad de mantener la información inalterada.

La **disponibilidad** se refiere a la necesidad de ofrecer un servicio ininterrumpidamente, de forma que pueda ser accedido en cualquier momento y desde cualquier lugar, evitando en lo posible que algún tipo de incidencia detenga el mismo.

2.1.2 Confianza

Otro aspecto muy importante en la seguridad de sistemas es la confianza. La confianza es la esperanza que se tiene de que un sistema se comporte como realmente debería. Establecer relaciones de confianza sin garantías conlleva la aparición de vulnerabilidades, que se convierten en potenciales amenazas.

2.1.3 Vulnerabilidad

Las vulnerabilidades son deficiencias o agujeros de seguridad del sistema que pueden ser utilizadas para violar las políticas de seguridad. Existen muchos tipos de vulnerabilidades. Pueden ser debidas a problemas en el diseño de una aplicación, bien de software o de hardware. O también pueden ser debidas a un plan poco exhaustivo o insuficiente de políticas de sistema.

2.1.4 Amenaza

Las amenazas son el resultado de explotar las vulnerabilidades. Una amenaza es una situación que tiene la capacidad de perjudicar o dañar al sistema. Aunque tanto las amenazas como las vulnerabilidades estén muy relacionadas, no son lo mismo. La detección de intrusiones se debe encargar de identificar y responder a ambas.

2.1.5 Políticas de seguridad

Las políticas de seguridad son el resultado de documentar las expectativas de seguridad. El concepto de seguridad, como se explicó antes, está relacionado con el comportamiento esperado de un sistema. Se puede afirmar que las políticas de seguridad intentan plasmar de alguna manera en el mundo real, los conceptos abstractos de seguridad.

Hay dos formas de definir las políticas de seguridad: procesal (o directiva) y formal.

La política de seguridad procesal consiste en plasmar de forma práctica las ideas o filosofías de la empresa en cuanto a seguridad. Aquí abajo se puede observar el funcionamiento básico de esta forma de entender las políticas de seguridad.

Política	Procedimiento	Práctica
Necesitamos proteger nuestro servidor Web contra accesos no autorizados.	Se mantendrá actualizado el servidor Web en cuanto a seguridad.	Se comprobará diariamente si existen parches de seguridad del servidor Web, en cuyo caso se aplicarán.
	Se instalará un IDS configurado para comprobar que la actividad en el servidor Web es normal.	Se instalará la última versión del "Snort", y se aplicarán los cambios de configuración pertinentes para concentrar la vigilancia especialmente en el servidor Web.

Tabla 2-1 - Ejemplo de política de seguridad procesal

Hay que señalar que los objetivos de la política de seguridad de un sistema son similares a los de los códigos legales. Ambos pretenden proteger a los usuarios legítimos del sistema de los delincuentes. Las políticas de seguridad se escriben en lenguaje informal, no de forma matemática.

Una política de seguridad formal es un modelo matemático del sistema que abarca todos los posibles estados y operaciones así como un esquema de cómo cada estado y operación pueden tener lugar. Definir este tipo de política de seguridad es una ardua labor. Es más apropiada para los diseñadores de sistemas de detección de intrusiones porque, al estar definida de una forma precisa, es más fácil de traducir en patrones de detección. Además, esta forma de describir el sistema ayuda al diseñador a escoger el tipo de información que se debe recopilar para el análisis.

2.1.6 Elementos de la infraestructura de seguridad

Aunque la detección de intrusiones pueda ser uno de los sistemas más importantes en el ámbito de la seguridad, no es la solución definitiva. Existen otros elementos que ayudan en la labor de mantener un sistema seguro, sin los cuales no se podría obtener un nivel apropiado de fiabilidad. En una instalación física segura, como un edificio, se utilizan materiales robustos para su construcción. Se sitúan ventanas de forma que no sean fácilmente accesibles por ladrones. Se colocan barreras y controles de acceso alrededor de la instalación. En el interior, además, se dispone de sistemas de vigilancia y alarmas, así como de personal debidamente equipado que patrulla continuamente la instalación. Este tipo de protección se encuentra a diario en bancos o instalaciones militares.

Aunque se contara con los mejores equipos de alarma de ladrones del mundo, a nadie se le ocurriría pensar que podrían sustituir al resto de elementos de la infraestructura de seguridad del complejo.

Pues bien, esto mismo ocurre con los ordenadores y las redes de datos. Hay numerosos componentes y funciones que forman parte del intrincado plan de estrategias de protección de un sistema. Algunos de los cuales se comentan a continuación.

2.1.6.1 Control de acceso

El control de acceso restringe el acceso a objetos según los permisos de acceso del sujeto. Se divide en Control de Acceso Obligatorio (MAC), en el que los permisos de acceso los proporciona el sistema; y el Control de Acceso Discrecional (DAC), en el que los permisos de acceso los controla y configura el propietario del objeto.

2.1.6.2 Identificación y Autenticación

Los mecanismos de identificación y autenticación (I&A) posibilitan la identificación adecuada de sujetos y objetos al sistema.

Estos elementos se pueden dividir en tres categorías, dependiendo de los datos que necesiten: lo que sabes, lo que tienes, lo que eres. Todas y cada una de las categorías implica un secreto que sólo conocen el sistema y el usuario. Si el secreto del usuario coincide con el que guarda el sistema, se valida la identidad del usuario y se obtiene permiso de acceso al sistema.

"Lo que sabes" se corresponde con el mecanismo básico de I&A, en el que cada sujeto se identifica y autentifica con un nombre de usuario y una contraseña. Desgraciadamente, este mecanismo ha demostrado ser ineficaz ante varios ataques, como "password-crackers" (rompedores de contraseñas) o troyanos que capturan actividad de teclado. Esta técnica de autenticación está siendo lentamente reemplazada por otras más robustas, de conocimiento cero que evitan el acto de pasar el secreto en sí mismo. [3]

La siguiente categoría, "lo que tienes" se puede ejemplificar claramente en sistemas "token-based" (basados en testigos), tales como los que necesitan el uso de una tarjeta inteligente, una clave ("key"), un disco especial. Muchos de estos testigos se han diseñado para utilizar medios criptográficos y soportes físicos resistentes para protegerse de ataques o suplantaciones de identidad (enmascaramiento).

Por último "lo que eres" representa a los mecanismos de I&A que utilizan elementos biométricos tales como la voz, huellas dactilares, o retina.

Los procesos de autenticación también se utilizan para proporcionar seguridad, y no sólo para dar acceso al sistema a los usuarios. Además, también sirven a los sistemas de detección de intrusiones para detectar si los comportamientos sospechosos son iniciados por usuarios legítimos o intrusos.

2.1.7 Cifrado

El cifrado es probablemente el método más antiguo utilizado para proteger información. No sólo permite ocultar información a sujetos no autorizados, sino que permite detectar posibles alteraciones, intencionadas o accidentales, en la misma.

El cifrado es el proceso por el cual un documento en claro, sometido a un algoritmo de cifrado con una clave, da lugar a un documento cifrado. Aunque el cifrado protege en gran medida la información, no puede evitar que sea eliminada de forma malintencionada. No puede proteger el documento antes de ser cifrado ni después de ser descifrado. Además, es completamente inútil si la clave es descubierta.

2.1.8 Cortafuegos

Los cortafuegos proporcionan una barrera de seguridad entre redes de distintos niveles de confianza o seguridad, utilizando políticas de control de acceso de nivel de red. Los elementos que entran en este grupo son por ejemplo los servidores "proxy", filtros de paquetes de red, túneles de datos cifrados (también conocidos como Redes Privadas Virtuales (VPN)). Los cortafuegos filtran paquetes de red, permitiendo o denegando su paso según las políticas establecidas. También hacen traducciones de direcciones, permitiendo mantener oculta la configuración interna de una red local.

2.2 Motivos que originan problemas de seguridad

Los problemas de seguridad pueden deberse a una enorme variedad de razones. No obstante, la etiología de la gran mayoría de los problemas de seguridad se divide en tres categorías: diseño/desarrollo, gestión, y confianza.

2.2.1 Diseño o desarrollo

Los problemas originados por un diseño o desarrollo ineficaces afectan tanto al software como al hardware. Un ejemplo de esto lo tenemos en las tarjetas inteligentes que albergan claves de cifrado. Se ha conseguido extraer muchas de estas claves de las tarjetas reproduciendo el reloj adecuado y variando el voltaje de algunas señales eléctricas de entrada. Otro ejemplo clásico es el que ocurre cuando un usuario malicioso substituye un valor en un programa en el intervalo de tiempo en que este no lo está usando. Este fallo se denomina condición de carrera ("race condition"), y tiene lugar cuando aparece un intervalo entre el momento de creación de un valor y el de su chequeo. Otro ejemplo famoso es el que consiste en desbordar el "buffer" de entrada de una determinada aplicación, pasándole como argumentos unos parámetros intencionadamente largos, provocando la caída del programa y consiguiendo entrar al sistema, casi siempre con privilegios de administrador.

Muchos de estos problemas se pueden prevenir mediante una sólida formación en mecanismos de diseño y desarrollo seguro, y sometiendo a los productos a duros controles de calidad.

2.2.2 Gestión

En este grupo entran los problemas debidos a una incorrecta configuración del sistema o de cualquier mecanismo encargado de protegerlo. Son de índole diversa, como por ejemplo una inadecuada aplicación de los permisos de los archivos de sistema o una plantilla de seguridad demasiado permisiva. También entrarían aquí situaciones en la que bien los administradores o los usuarios sortean los mecanismos de seguridad de alguna manera. Por ejemplo, cuando en una red local, protegida mediante un cortafuegos, alguien decide utilizar un módem para establecer una conexión con el exterior. Esta conexión está, naturalmente, burlando los controles establecidos por el administrador.

2.2.3 Confianza

Los problemas más agudos son los relacionados con la confianza. Y muchas veces ocurren por no diferenciar entre el entorno de desarrollo y el de producción. Como ejemplo está el sistema operativo UNIX. Durante sus comienzos, fue desarrollado en un entorno universitario. Con el paso del tiempo, sin embargo, surgieron expectativas comerciales para este sistema. Ya no iba a ser utilizado exclusivamente por programadores o ingenieros de sistemas. Al principio los diseñadores confiaban en que los usuarios utilizaran el sistema según las especificaciones, en un determinado entorno bajo unas características especiales. Los usuarios confiaban en que hubiera un administrador controlando el sistema de forma fiable y competente. Pero ¿qué pasaba cuando se rompía esta confianza? Este escenario sigue teniendo lugar, provocando fallos de seguridad.

2.3 Elementos de la detección de intrusiones

2.3.1 Arquitectura

A la hora de proteger un sistema basándose en los registros que se analizan mediante una auditoría requiere que estos registros sean almacenados de una forma segura en un entorno distinto al del sistema protegido. Este requisito lo cumple cualquier sistema de detección de intrusiones con un mínimo de calidad. Esto se hace por varias razones. Para evitar que el intruso pueda eliminar los registros, para evitar que el intruso pueda alterar la información contenida en los registros y para no perjudicar con el mecanismo de detección de intrusiones el rendimiento del sistema a proteger.

En este tipo de arquitectura el sistema que ejecuta el sistema de detección de intrusiones se denomina "host" y el sistema monitorizado "target" (objetivo).

2.3.2 Fuentes de datos, monitorización

La fuente de datos es una de las primeras cosas a tener en cuenta a la hora de diseñar un sistema de detección de intrusiones. Estas fuentes se pueden clasificar de muchas maneras. En lo que respecta a la detección de intrusiones las clasificaremos por localización. De esta forma, la monitorización de sistemas, y por tanto la detección de intrusiones, se puede dividir en cuatro categorías: "host", red, aplicación y objetivo. Usaremos el término "monitorizar" como el acto de recoger datos de una determinada fuente y enviarlos a un motor de análisis.

- **Monitores basados en máquina ("host based"):** Recogen los datos generados por un ordenador, normalmente a nivel del sistema operativo. Los registros de sucesos y las colas de auditoría pertenecen a este grupo.
- **Monitores basados en múltiples máquinas ("multi-host based"):** Como su propio nombre indica, utiliza la información recogida en dos o más máquinas. Su enfoque es muy similar al basado en máquina, con la dificultad añadida de tener que coordinar los datos de varias fuentes.

- **Monitores basados en redes ("network based"):** Capturan paquetes de red. Para ello, normalmente se utilizan dispositivos de red en modo promiscuo, convirtiendo al sistema en un "sniffer" o rastreador.
- **Monitores basados en aplicación ("application based"):** Registran la actividad de una determinada aplicación. Por ejemplo, los registros de un servidor ftp.
- **Monitores basados en objetivos ("target based"):** Estos monitores difieren ligeramente del resto porque generan sus propios registros. Utilizan funciones de cifrado para detectar posibles alteraciones de sus objetivos, y contrastan los resultados con las políticas. Este método es especialmente útil cuando se usa contra elementos que, por sus características, no permiten ser monitorizados de otra forma.
- **Monitores híbridos ("hybrid"):** Combinan dos o más fuentes de distinto tipo. Cada vez es más frecuente encontrarse con productos de detección de intrusiones basados en este punto de vista. Así, amplían sus posibilidades de detección.

Existen productos que combinan varias estrategias de monitorización. Estas soluciones se denominan *soluciones integradas*.

Hay que añadir que existen sistemas de detección de intrusiones que reciben el nombre de NNID ("Network Node Intrusion Detector"), es decir, **Detector de Intrusiones de Nodo de Red**. En realidad, son un caso especial de la detección basada en red. Este nombre se aplica cuando el monitor basado en red se sitúa en un "host", monitorizando los paquetes destinados u originados por la máquina anfitriona. Una de las razones más importantes para hacer esto es para sortear el problema de las encriptaciones durante la comunicación, ya que el tráfico es cifrado o descifrado por el propio "host". Un detector basado en red convencional no podría analizar el tráfico en un punto intermedio entre dos nodos que cifraran sus comunicaciones. Los NNIDS son también útiles en entornos de red con conmutadores ("switches") en los que un "host", aunque esté en modo promiscuo, sólo percibe el tráfico destinado a él. En el siguiente capítulo, en el apartado 3.1.2.1 "Paquetes de red", se comenta en detalle este último aspecto.

2.3.3 Tipos de análisis

Después del proceso de recopilación de información, se lleva a cabo el proceso de análisis. La detección de intrusiones también se puede clasificar según los objetivos del motor de análisis. Los dos tipos principales de análisis son:

- **Detección de usos indebidos ("misuse"):** Para encontrar usos indebidos se comparan *firmas*¹ con la información recogida en busca de coincidencias.
- **Detección de anomalías:** Para la detección de anomalías se manejan técnicas estadísticas que definen de forma aproximada lo que es el comportamiento usual o normal.

¹ Patrones de ataques conocidos.

La siguiente figura muestra un esquema general de detector de intrusiones de usos indebidos (mediante comparación de patrones) y de anomalías.

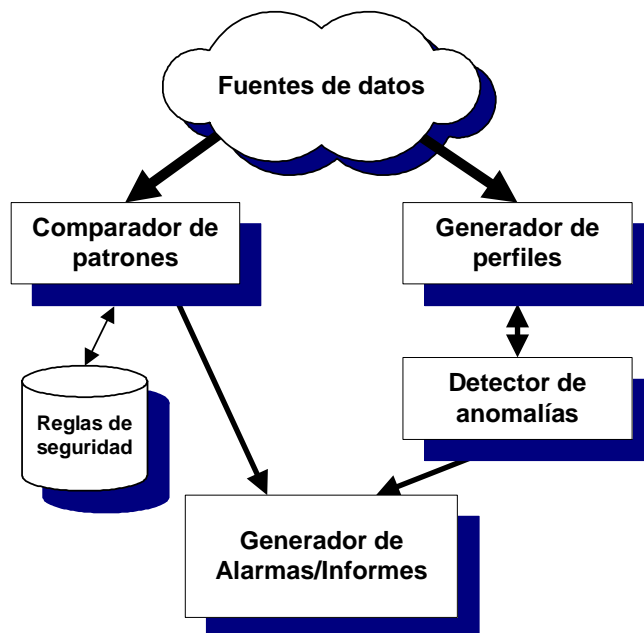


Figura 2-1 - Esquema general de un Sistema de Detección de Intrusiones

La mayoría de los detectores son de usos indebidos, anomalías o una mezcla de ambos. Algunas empresas están empezando a utilizar también técnicas específicas para la detección de ataques de denegación de servicio (DoS), dadas sus características especiales.

Aparte del análisis basado en firmas y estadísticas, también existe el análisis *de integridad*. Este es el método utilizado por las herramientas de chequeo de integridad de ficheros, que complementan a los Sistemas de Detección de Intrusiones. Estas herramientas detectan cambios en ficheros u objetos, utilizando mecanismos robustos de encriptación tales como funciones resumen ("hash functions").

Otro enfoque a la hora de distinguir formas de detección de intrusiones es teniendo en cuenta el uso que hacen los análisis del *tiempo*:

- **Por lotes ("batch mode"):** Cada intervalo de tiempo se procesa una porción de los datos recibidos, enviando las posibles alarmas de intrusiones después de que hayan ocurrido.
- **Tiempo real:** Los datos son examinados en el tiempo en que son recibidos (o con un retardo mínimo). La aparición de los análisis en tiempo real hizo posible las respuestas automáticas.

2.3.4 Respuestas

El mecanismo de respuesta, explicado en detalle en el capítulo 3, es otro de los factores que ayudan a definir el tipo de sistema de detección de intrusiones:

- **Respuestas pasivas:** En este caso, el detector no toma acciones que puedan cambiar el curso de un ataque. En vez de esto, se limita a enviar o registrar la alarma correspondiente al responsable cualificado.
- **Respuestas activas:** Pertenecen esta categoría aquellos sistemas que, además de generar la alarma correspondiente, reaccionan modificando el entorno. Un ejemplo de este tipo de respuesta activa consiste en el bloqueo de las acciones del intruso, o el cierre de la sesión del usuario sospechoso.

2.3.5 Clasificación general

Lo visto hasta ahora permite realizar una clasificación de los IDSs según diversos criterios. Aunque hay más formas de clasificar estos sistemas, se han sintetizado las más comunes en la figura a continuación:

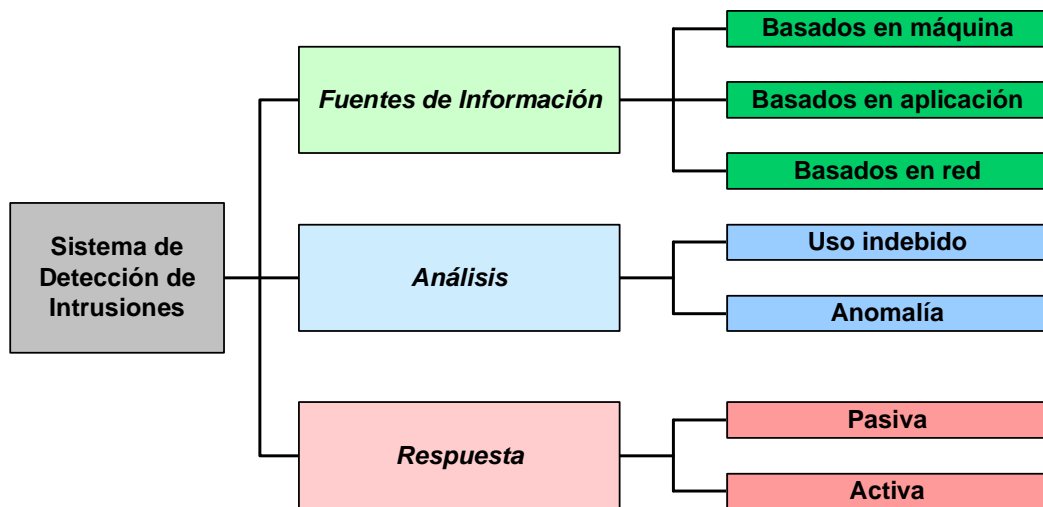


Figura 2-2 -Tipos principales de IDS

2.3.6 Objetivos

Los objetivos de la detección son otro factor a tener en cuenta en el análisis de la detección de intrusiones.

Si además de detectar posibles errores de seguridad, se pretende perseguir al atacante mediante acciones legales, es importante dedicar tiempo a la apropiada conservación y formato de los registros generados por el sistema. Si, por el contrario, sólo se desea mantener seguro el sistema, utilizando las posibles intrusiones para corregir los posibles errores que puedan ir surgiendo, los datos de las auditorías se pueden eliminar.

Por otra parte, como ya se comentó, la mejora de las capacidades de proceso ha hecho posible el análisis en tiempo real. Esto ha permitido desarrollar mecanismos automáticos de respuesta ante posibles ataques, como por ejemplo denegando el acceso a un posible intruso, o reflejando contra el atacante los ataques realizados. Estos métodos se describen con más detalle en capítulos posteriores.

2.3.7 Control

La forma de administrar un sistema de detección de intrusiones es otro elemento a tener en cuenta. Existen dos acercamientos, según el sistema monitorice múltiples "hosts" o redes: la centralización y la integración con herramientas de gestión de redes.

La centralización consiste en concentrar las funciones de control en un nodo, que dirige a los demás elementos de detección de intrusiones. Para este punto de vista es necesario establecer comunicaciones seguras entre los elementos del sistema. También es necesario poder mostrar los resultados recogidos por todos los elementos de forma coherente y clara.

Otra forma de solucionar el control de la detección de intrusiones centralizada es hacer que forme parte de las funciones de gestión de redes. Muchos productos comerciales de detección de intrusiones ofrecen la posibilidad de generar mensajes SNMP (Protocolo de Gestión de Redes Simple) para la herramienta de captura de gestión de redes.

2.4 Referencias

- [1] Bace, R. *Intrusion Detection*. Macmillan Technical Publishing, 2000.
- [2] Real Academia Española. *Diccionario de la lengua Española*. Espasa Calpe, S.A. 1994.
- [3] H. Danisch. *The Exponential Security System TESS: An Identity-Based Cryptographic Protocol for Authenticated Key-Exchange* (E.I.S.S.-Report 1995/4). [en línea]. Agosto, 1995 [consultado en febrero, 2003]. Capítulo 3.1, Zero Knowledge Authentication. Categoría informativa.
<<http://www.ietf.org/rfc/rfc791.txt>>.

Modelo de funcionamiento

En este capítulo se describe en detalle el modelo más aceptado para la detección de intrusiones. Tiene tres funciones principales, la fase de recogida de datos (fuentes de información), la fase de análisis, y la fase de respuesta. En líneas generales, para que la detección de intrusiones pueda obtener buenos resultados, debe llevar a cabo un proceso de recopilación de información, que posteriormente deberá someter a diversas técnicas de análisis, y en función de los datos obtenidos tendrá que dar algún tipo de respuesta.

3.1 Fuentes de información

Como se comentó en el capítulo anterior, la detección de intrusiones se puede clasificar según las fuentes de información que utiliza. Aquí se estudiarán los casos pertenecientes a la recopilación de datos basados en máquina, en red, y en fuentes externas como sistemas de seguridad físicos. Explicados en ese orden, se tratan las capas de menor a mayor nivel de abstracción.

Ninguno de los sistemas de detección de intrusiones en particular es mejor que los otros. La elección de una detección de intrusiones basada en máquina, red o cualquier otra, depende de las necesidades particulares. A veces basta con instalar un detector en algunos tramos de red, mientras que en otras ocasiones es necesario incrementar la seguridad recogiendo datos de las máquinas.

3.1.1 Fuentes de información basadas en máquina

Este tipo de fuentes de información consisten principalmente en registros de auditoría de sistemas operativos (registros generados por mecanismos del sistema operativo), y los registros de sistema (ficheros generados por el sistema y aplicaciones, generalmente ficheros de texto generados línea a línea). Los monitores basados en múltiples "hosts" utilizan las mismas fuentes, por lo que los apartados a continuación también son aplicables a estos.

3.1.1.1 Registros de auditoría

El primer elemento de importancia en sistemas de detección basados en máquina son los registros de auditoría. Estos registros son una colección de información sobre las actividades del sistema, creados cronológicamente y dispuestos en un conjunto de ficheros de auditoría. Estos registros son originados por los usuarios y los procesos y comandos que estos invocan. Muchos de los registros de auditoría fueron creados originalmente para cumplir los requisitos del Programa de Evaluación de Productos Fiables (TCSEC), una iniciativa del gobierno estadounidense. El TCSEC, también conocido como Libro Naranja, define las características requeridas por los sistemas operativos de uso comercial y las aplicaciones de software que contuvieran o procesaran información clasificada. [1]

Por aquel entonces, el Libro Naranja presentaba un problema, ya que presentaba extensos requerimientos de auditoría, pero no las directrices para su utilización. Listaba una gran cantidad sucesos que debían ser registrados, pero posteriormente no explicaba la forma de seleccionarlos. Tampoco explicaba de qué modo o con qué estructura debían ser almacenados los registros de auditoría. Por lo tanto, los fabricantes crearon numerosas y variadas soluciones para cumplir los requerimientos de auditoría de la clase C2¹. Los desarrolladores de detección de intrusiones que se familiarizaban con los registros de auditoría de un sistema operativo no comprendían los registros generados por otro sistema, que podía estar cumpliendo los mismos requerimientos.

Los fabricantes utilizaron al menos dos soluciones en sus sistemas de auditoría. Una creaba registros "autónomos", que no necesitaban de otros registros para su interpretación. La eliminaba información redundante en los registros almacenando la información de un evento en registros múltiples.

Los sistemas de detección de intrusiones hacen un uso importante de la información contenida en los registros de auditoría. Desgraciadamente, gran parte de los sistemas operativos comerciales existentes no cumplen con los requisitos de auditoría necesarios, a pesar de la documentación que hay sobre el tema. Esto dificulta la labor a los desarrolladores de detección de intrusiones. Algunos han sugerido que para una detección de intrusiones basada en máquina sea efectiva es necesario añadir funcionalidades al núcleo del sistema para que genere la información de auditoría necesaria. Esto provoca costes en el rendimiento del sistema, y costes asociados al mantenimiento de las alteraciones de los sistemas operativos. [2]

No obstante, muchos desarrolladores prefieren utilizar los registros de auditoría frente otras fuentes de información por varias razones. Una de ellas es que la propia estructura del sistema operativo está diseñada para otorgar suficiente seguridad al sistema de auditoría, y los registros que este genera. Otro de los motivos que les llevan a esta decisión es que el sistema de auditoría trabaja a bajo nivel, por lo que ofrece mayor nivel de detalle que el que puede ofrecer otro mecanismo.

Sin embargo, es importante resaltar que obtener datos excesivamente detallados dificulta la diferenciación entre las actividades originadas directamente por usuarios y aquellas originadas por programas que han tomado la identidad de un usuario. Hay numerosos ataques que explotan la capacidad de los programas para asumir la identidad de un usuario que tiene mayores privilegios que el actual. Para detectar esta clase de ataques, la fuente de datos debe proporcionar suficiente información para permitir la diferenciación entre usuario y proceso.

3.1.1.2 Contenido de los registros de auditoría

Los eventos de sistema contienen información sobre la actividad del sistema como del objeto que la ha originado. Los sistemas operativos comerciales guardan eventos a nivel de núcleo (llamadas de sistema) y a nivel de usuario (eventos de aplicación). Para identificar a los procesos y a los usuarios, se proporciona información inequívoca sobre los procesos e identificadores de usuario (userID). A veces incluso se registra el userID original, y el userID adoptado por el proceso, si este ha cambiado.

¹ La clase C2, descrita en el Libro Naranja, pertenece a las clases de "Protección Discrecional" (tipo C). Define una serie de capacidades de auditoría referentes al control de acceso y hace responsables a los usuarios de sus acciones.

A continuación se explicará la estructura del sistema de auditoría de dos sistemas operativos: Sun, el Basic Security Module (BSM) y Windows NT.

3.1.1.2.1 Solaris BSM

El Módulo de Seguridad Básico de Sun se ha diseñado para cumplir los requisitos del TCSEC C2.

La estructura del subsistema de auditoría de BSM consiste en un registro ("log") de auditoría, varios ficheros de auditoría, informes ("records") de auditoría, y testigos ("tokens") de auditoría.

Un registro de auditoría consiste en un conjunto de ficheros de auditoría, que a su vez están compuestos por varios informes de auditoría. Estos informes, como se verá más adelante, están formados por varios testigos de auditoría.

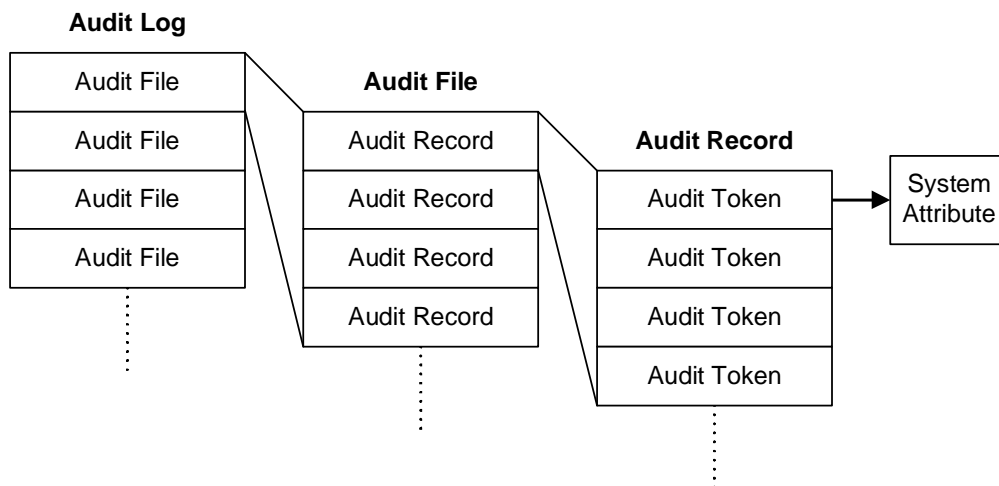


Figura 3-1 - Estructura de los registros de auditoría BSM

Cada informe de auditoría se almacena en formato binario y describe cada evento ocurrido, e incluye información tal como quién hizo la acción, qué ficheros están involucrados, qué acción tuvo lugar y dónde y cómo sucedió. [3]

La Figura 3-2 muestra el conjunto de testigos que constituyen un informe de auditoría.

Audit Record	
Header Token	
Argument Token	
Data Token	
Subject Token	
Return Token	

Figura 3-2 - Estructura de un informe ("record") de auditoría BSM

Hay eventos generados a nivel de núcleo ("kernel-level audit events") y a nivel de usuario ("user-level audit events"). La estructura de ambos es similar.

Para facilitar la labor de la auditoría, los eventos están organizados por clases ("audit classes"). Existen 32 clases de auditoría.

Para configurar la auditoría se utilizan indicadores de auditoría ("audit flags"), que permiten especificar qué clases eventos auditar.

El sistema operativo ofrece algunas herramientas para la gestión de los eventos de auditoría. De esta manera, se puede utilizar el comando `praudit` para traducir los historiales de auditoría almacenados en binario en un formato legible para el usuario. Por otra parte, `auditreduce` permite realizar filtrados de los eventos generados, a partir de parámetros como intervalos de tiempo, determinados identificadores de usuarios, o determinados eventos de sistema.

3.1.1.2.2 Windows NT/2000

Este sistema operativo genera tres tipos de eventos de sistema:

- Eventos de sistema operativo.
- Eventos de seguridad.
- Eventos de aplicación.

Los **eventos de sistema** son los que generan los componentes de Windows. Son sucesos relacionados con fallos de controladores u otros objetos de sistema, pérdida de datos, problemas con el registro, etc. Estos tipos de eventos son predeterminados por el sistema operativo.

Los **eventos de aplicación** son los generados por las diferentes aplicaciones del sistema. Por ejemplo, información asociada a una base de datos, un antivirus, o una operación de copia de seguridad. Estos eventos los definen los desarrolladores de software, y los "software toolkits" (conjuntos de herramientas "software") les ayudan en esta labor.

Los **eventos de seguridad** son los que tienen especial relevancia en materia de seguridad. Están diseñados a partir de las definiciones del TCSEC (clase C2). Estos eventos tienen

que ver con accesos a objetos, inicios y cierres de sesión, cambio de políticas de sistema, etc. A diferencia de los otros tipos de sucesos, estos sólo pueden ser accedidos por administradores, y son los más importantes para los detectores de intrusiones.

Los registros de eventos de Windows están formados por historiales de eventos. Cada historial tiene una cabecera, seguida de una descripción y, a veces, datos adicionales. La cabecera tiene los siguientes campos. La mayoría son auto explicativos: Fecha, Hora, Nombre de Usuario, nombre de Máquina, ID de evento, Fuente (una aplicación, un servicio de sistema, un controlador), Tipo (indica la gravedad del evento: un error, un aviso, una información, éxito, fracaso), Categoría (utilizado principalmente en eventos de seguridad para indicar qué evento ha tenido éxito o fracaso).

Al igual que pasa con el sistema operativo Solaris de Sun, Windows NT proporciona varios elementos para hacer más llevadera la auditoría. Existen mecanismos de filtrado de sucesos. También es posible ordenarlos, ascendente o descendientemente, según los distintos elementos que componen cada evento. Se pueden definir parámetros tales como el tamaño del archivo del registro de eventos, y cómo se debe comportar el sistema en caso de llenarse, pudiéndose detenerse inmediatamente o sobrescribir los más antiguos.

Permission Components (Windows NT and Windows 2000)	Permission Types (Windows NT)					
	Read (R)	Write (W)	Execute (X)	Delete (D)	Change Permissions (P)	Take Ownership (O)
Traverse Folder / Execute File			•			
List Folder / Read Data	•					
Read Attributes	•		•			
Read Extended Attributes	•					
Create Files / Write Data		•				
Create Folders / Append Data		•				
Write Attributes		•				
Write Extended Attributes		•				
Delete Subfolders and Files						
Delete				•		
Read Permissions	•	•	•			
Change Permissions					•	
Take Ownership						•

Tabla 3-1 - Permisos en Windows NT y Windows 2000

La Tabla 3-1 muestra la relación entre los permisos básicos de Windows NT (lectura, escritura, ejecución, etc.) y los distintos componentes de permisos, indicados a la izquierda, formados a partir de combinaciones de aquellos.

Establecer una política de auditoría eficaz no es trivial. Aunque se podrían auditar todos los elementos del sistema, rara vez es la solución adecuada. Por una parte, tal medida tendría un importante impacto en el rendimiento del sistema. Además, de esta forma se generan muchos más eventos, que en muchos casos son irrelevantes o inútiles, enturbiando la claridad de los datos y sobrecargando la fase de análisis.

3.1.1.3 El problema de la reducción de auditoría

La reducción de auditoría, en anglosajón "audit reduction", tiene lugar cuando se pretende eliminar información redundante o no necesaria de los registros de auditoría.

Un dato clave para llevar a cabo la reducción de auditoría es el de introducir determinismo en procesos relativamente no deterministas. Es decir, si sabemos que un suceso A siempre suele venir seguido de otros elementos dados (U, V, W) bajo ciertas condiciones (R, S), podemos determinar el suceso: A ocurre bajo condiciones R, S seguido de U, V, W, al suceso A.

Desgraciadamente, el determinismo no siempre es aplicable. Especialmente en entornos multitarea. Por ejemplo, en UNIX de Sun, un simple comando de alto nivel, como `ls`, en una estación de trabajo puede generar más de mil historiales de auditoría. Si este comando se repite en unos segundos puede generar un número distinto de eventos, y en diferente orden.

Existen todo tipo de mecanismos para llevar a cabo reducciones de auditoría. Desde simples filtrados de eventos según la información de alguno de sus campos. Hasta elaborados modelos matemáticos, como el "Filtro Basado en Concordancia de Patrones" [4] o la "Detección de Intrusiones utilizando patrones de rastro de auditoría de longitud variable" [5].

También hay importantes reducciones de eventos eliminando los eventos generados por procesos fiables. Aquí, los procesos fiables son aquellos certificados como soporte para obtener seguridad.

3.1.1.4 Registro de Sistema

El registro de sistema ("system log") es otro elemento importante a la hora de recopilar información del sistema. Es un fichero en el que se guardan los eventos generados por el sistema. El sistema operativo UNIX cuenta con una variedad importante de registros de sistema, relacionados por un servicio común, denominado "syslog". Este servicio genera y actualiza los registros de eventos mediante el proceso `syslogd`.

La seguridad de los registros de sistema es uno de los puntos débiles frente a la de los de auditoría. En este sentido, los registros de sistema son menos fiables que los de auditoría. Hay varias razones por las que esto es así. Los registros de sistema son escritos por aplicaciones, más vulnerables que el subsistema de auditoría. Por otra parte, suelen almacenarse en directorios no protegidos del sistema, relativamente fáciles de localizar y alterar. Además, están escritos en texto en claro, y no en una forma más críptica como los registros de auditoría, que siempre ayuda más a detectar cambios.

No obstante, los registros de sistema aportan información muy útil a los programas de detección de intrusiones, y complementan a los datos provenientes de los registros de auditoría. Además, son más fáciles de revisar que los registros de auditoría de sistema.

Por otra parte, siempre es más conveniente utilizar varias fuentes de información que una sola. Así, se pueden detectar signos de intrusiones a través de las discrepancias encontradas.

Para solventar los problemas de seguridad inherentes a los registros de sistema se han desarrollado diferentes métodos. Spafford y Garfinkel propusieron uno que consistía en enviar los registros de un sistema a una máquina dedicada utilizando una conexión serie [6].

3.1.1.5 Registros de sistema comunes

Como ya se explicó, existen numerosos registros de sistema. Pero no todos sirven para detectar intrusos. Los programas de recopilación de información de seguridad seleccionan los más relevantes. Los sistemas operativos basados en UNIX los almacenan en los directorios:

/usr/adm	Utilizado en las primeras versiones de UNIX.
/var/adm	Utilizado en versiones más modernas de UNIX.
/var/log	Utilizado en algunas versiones de Linux, BSD, FreeBSD.

A continuación se muestra una tabla con los registros más utilizados por los sistemas de detección de intrusiones. Además de estos registros, los desarrolladores pueden utilizar `syslogd` para escribir eventos, ampliando las posibilidades de detección de intrusiones a procesos de sistema no predeterminados:

Nombre de registro	Descripción
acct or pacct	Comandos ejecutados por todos los usuarios.
aculog	Llamadas de los módems.
lastlog	Última entrada con éxito y sin éxito en el sistema de cada usuario.
loginlog	Todos los intentos de acceso sin éxito.
messages	Mensajes de salida de la consola del sistema y otros generados desde el servicio syslog.
sulog	Uso del comando "su"
utmp[x]	Dentro de este o estos directorios está la información del usuario que está en el sistema.
utmpx	utmp extendido.
wtmp[x]	Contiene un listado de tiempos de cada entrada y salida de cada usuario.
wtmpx	wtmp extendido.
xferlog	Accesos mediante FTP.

Tabla 3-2 - Registros relativos a seguridad en Solaris

3.1.1.6 Información de aplicaciones

Hasta ahora nos hemos centrado en el nivel de sistema como fuente de información para la detección de intrusiones. El nivel de sistema se supone el más robusto e inaccesible para todos salvo los más expertos. Sin embargo, los modelos de seguridad y de protección están en constante evolución, al mismo tiempo que los sistemas operativos.

Los profesionales del campo de la detección de intrusiones piensan que en el futuro, la mayoría de los datos de importancia procederán del nivel de aplicación. Uno de los ejemplos de esta realidad es el progresivo avance de los sistemas distribuidos y los sistemas orientados a objetos. En el sistema operativo Windows NT, muchos de los eventos generados por el nivel de registro de sistema operativo han migrado a almacenes de datos de aplicación. Además, casi todos los sistemas operativos comerciales soportan la entrada de registros de auditoría generados en el nivel de aplicación.

3.1.1.6.1 Bases de datos

En las grandes organizaciones, cada vez con más frecuencia, la información se almacena y manipula mediante un sistema de gestión de bases de datos.

Uno de los aspectos más importantes de las bases de datos está relacionado con el volumen de información de auditoría que pueden generar. En algunos casos, se pueden producir "gigabytes" de datos de auditoría en cuestión de horas. Esto obliga, durante el diseño de sistemas de detección de intrusiones, a crear mecanismos de compresión para los datos, técnicas de reducción de auditoría, selección de grupos de eventos útiles descartando los no necesarios.

Los registros de las transacciones generadas por las bases de datos, al igual que los registros de sistema, no están tan protegidos como los eventos de auditoría. Pero ocurre que en las bases de datos se almacena información que los usuarios y las empresas desean proteger. Se están haciendo esfuerzos de desarrollo en este aspecto. Los sistemas de auditoría relacionados con la gestión de bases de datos son una fuente de información importante para los sistemas de detección de intrusiones. [7]

3.1.1.6.2 Servidores Web

Debido al enorme crecimiento experimentado en los últimos años del uso de estos servicios, los servidores Web son cada vez más utilizados. Muchos de estos servidores permiten la generación de valiosa información a nivel de aplicación, en forma de registros.

Aquí comentaré los formatos de los registros más estandarizados entre este tipo de aplicaciones.

Common Log Format (CLF)

Este formato fue utilizado originalmente por el servidor Web del NCSA¹, llegando a convertirse en el estándar a utilizar por los servidores Web en sus registros. La mayoría de las herramientas de análisis de registros lo soportan. Los registros CLF contienen casi toda la información necesaria para realizar estudios exhaustivos sobre la actividad de un servidor Web, cuya estructura se describe a continuación [8]:

¹ El "National Center for Supercomputing Alliances" (NCSA) fue el responsable de la creación del primer navegador extensamente conocido.

Campo	Descripción	Ejemplos
máquina remota	Dirección IP o nombre DNS del cliente.	192.168.0.1, www.upm.es
rfc931	Identificación remota del cliente. No se aplica. "identd" es un servicio exclusivo de UNIX. La búsqueda de identidad consume mucho ancho de banda, y sobrecarga innecesariamente a los servidores. Raramente se utiliza. Si no existe se escribe un guión ("-").	
Autenticación de usuario	Nombre de usuario del cliente.	diego, dggomez
Fecha y hora	Fecha y hora: formato [DD/MMM/AAAA:HH:MM:SS +-TZO].	[03/07/2003:11:30:00 +0100]
Petición	Línea de petición hecha por el cliente, entre comillas.	"GET /menu.html HTTP/1.0" "POST /form.cgi HTTP/1.0"
Estado [9]	NNN. Estado HTTP devuelto al cliente, "-" si no está disponible.	200, 406, 303
longitud	NNNN. Número de bytes enviados al cliente, "-" si no está disponible.	1995, 573, 907, 2003

Tabla 3-3 - Estructura de registros CLF

Un ejemplo válido de una entrada de registro CLF podría ser:

```
www.microsoft.com - - dggomez [03/07/2003:1:40:00 +0100] "GET
/passw.txt HTTP/1.1" 200 1976
```

W3C Extended Log Format (ELF)

Consiste básicamente en el NSCA CLF más los campos agente de usuario ("user-agent") y URL de procedencia ("referrer URL information") sin comillas [10]. Quedando una estructura como la siguiente:

CLF user_agent referrer_URL

El siguiente registro, aunque en más de una línea por falta de espacio, podría ser un ejemplo válido de este formato:

```
www.euitt.upm.es - - dggomez [19/Mar/2003:21:14:55 -0200] "GET /
HTTP/1.1" 200 1234 Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
http://www.euitt.upm.es/index.html
```

Definable Lof Format (DLF)

Este formato muy similar al ELF, pero al final de cada entrada de registro invierte los campos URL de procedencia y agente de usuario, dejándolos con o sin comillas, de la siguiente forma:

CLF "referrer_URL" "user_agent"

CLF referrer_URL user_agent

Un ejemplo de este formato, sería similar a:

```
138.100.52.100 - - dggomez [08/Apr/2003:16:02:20 +0100] "GET /
HTTP/1.1" 200 5678 "http://www.euitt.upm.es/index.html" "Mozilla/4.05
(X11; I; IRIX64 6.4 IP30)"
```

Existen más formatos, como el "Binary Log Format" (BLF), muy similar al NCSA CLF pero más sintetizado, con menos campos.

La obtención de datos de aplicaciones presenta dos de los grandes retos de la detección de intrusiones a la hora de recoger información. Por un lado está el problema del tiempo. Es fundamental establecer un orden cronológico o incluir algún parámetro de tiempo en las entradas de los registros para que la información pueda ser analizada correctamente. Por otra parte tenemos el problema de la combinación de los diferentes registros, para que los usuarios puedan comprenderlos adecuadamente. El proceso de combinar varios flujos de datos para obtener otra cosa se denomina *composición* (posible mediante *correlación* previa). Cuando estos flujos sirven para crear otro de nivel de abstracción más alto, se utiliza el término de inteligencia artificial *fusión*. Estos dos términos son dos aspectos técnicos muy relevantes en la detección de intrusiones.

Un ejemplo de producto de detección de intrusiones basado en aplicación es *Appshield*, de la empresa Sanctum Inc. [11]

3.1.1.7 Información recogida de objetivos

Un monitor basado en objetivo ("target based") es muy similar a un monitor basado en máquina ("host based"). La monitorización basada en objetivo establece mecanismos para vigilar el estado de una serie de recursos valiosos del sistema, grabando periódicamente el estado de los objetos monitorizados. Estos estados se comparan con unas políticas de seguridad, registrando las posibles discrepancias.

Uno de los recursos más utilizados para la monitorización basada en objetivo es el de los verificadores de integridad. Estas herramientas se utilizan para complementar la labor de los detectores de intrusiones, monitorizando cambios en el estado de objetos de sistema, como ficheros importantes. Este enfoque es *estático*, no como los mecanismos de registro de auditoría o sistema, que son *dinámicos*. Para aclarar el concepto, se podría decir que un ejemplo estático, como la monitorización basada en objetivo, es una imagen, mientras que uno dinámico es como un vídeo. Una cámara fotográfica puede hacer imágenes a intervalos periódicos de tiempo, mientras que una cámara de vídeo, que es más cara, permite ver lo que pasa en tiempo real. Cada solución depende de las necesidades particulares. Algunas empresas no necesitan una alta velocidad de detección.

Un verificador de integridad genera una *suma de comprobación* ("checksum") de cada objeto de sistema y las almacena en un lugar protegido. Para fabricar la suma de comprobación se utiliza un *algoritmo de resumen de mensaje* ("message digest algorithm"), o *función de resumen* ("hash function"). Estos algoritmos se diseñan con dos propósitos. Primero, para que sean tan seguros que el hecho de obtener el mismo resultado utilizando dos entradas distintas sea prácticamente nulo. Y segundo, para que cualquier modificación en la entrada, por pequeña que sea, produzca una enorme diferencia en la salida.

Los monitores basados en objetivos son especialmente útiles en sistemas UNIX. La razón es que en estos sistemas operativos, cualquier objeto de interés (como conexiones de red, procesos

o dispositivos) puede ser representado como un fichero. Estos objetos se representan por estructuras denominadas *inodos* ("inodes").

Campo	Bytes	Descripción
Mode	2	Tipo de fichero, bits de protección, "setuid", bits "setgid"
NLinks	2	Número de entradas de directorio apuntando a este inodo
Uid	2	UID del propietario del fichero
Gid	2	GID del propietario del fichero
Tamaño	4	Tamaño del fichero en "bytes"
Dirección	39	Dirección de los primeros 10 bloques de disco, luego 3 bloques indirectos
Gen	1	Número de generación (Se incrementa con cada reutilización del inodo)
Atime	4	Hora en la que el inodo fue accedido por última vez
Mtime	4	Hora en la que el inodo fue modificado por última vez
Ctime	4	Hora en la que el inodo fue cambiado por última vez

Tabla 3-4 - Contenido de un inodo del sistema de ficheros UNIX (System V)

Los primeros verificadores existentes en UNIX comprobaban el estado de los ficheros utilizando códigos de redundancia cíclica (CRC). Estos códigos fueron diseñados originalmente para detectar errores en comunicaciones sobre canales con ruido. Por lo tanto, detectaban cambios aleatorios en los objetivos y no cambios intencionados en sus contenidos. Algunos ataques demostraron que, con la ayuda de las herramientas adecuadas, era posible modificar el contenido de los ficheros sin alterar su CRC.

A principios de los años noventa, Gene Spafford y Gene Kim desarrollaron una herramienta denominada Tripwire® que optimizaba el proceso de crear códigos de resumen criptográfico para proteger ficheros críticos de sistema. Actualmente este programa es comercial, y está disponible para plataformas Solaris, Windows NT y Linux. [12]

3.1.2 Fuentes de información basadas en red

Los monitores basados en red son quizás los más famosos en el ámbito de la detección de intrusiones. El tráfico de red; flujo de información tal como viaja por un segmento de red, es la fuente de información que se tratará en esta sección.

La recopilación de datos de red tiene varias ventajas. Para empezar, utilizar como fuente de información el tráfico de red, no afecta al rendimiento del resto de las máquinas de la red.

Por otra parte, el monitor puede ser transparente al resto de los miembros de la red. Esto significa que puede ser indetectable, lo que es una ventaja ya que no puede convertirse en objetivo directo de posibles intrusos. Con este propósito, existe la posibilidad de utilizar un cable de sólo recepción ("sniffing cable") para el monitor, de forma que sólo pueda recibir datos, impidiendo físicamente cualquier envío de señales [13]. Una opción equivalente al cable de sólo recepción es el

uso de un "network tap" (dispositivo de escucha de red); un dispositivo de aspecto similar a un concentrador de red, que permite a un rastreador *pinchar* las comunicaciones sin ser detectado.

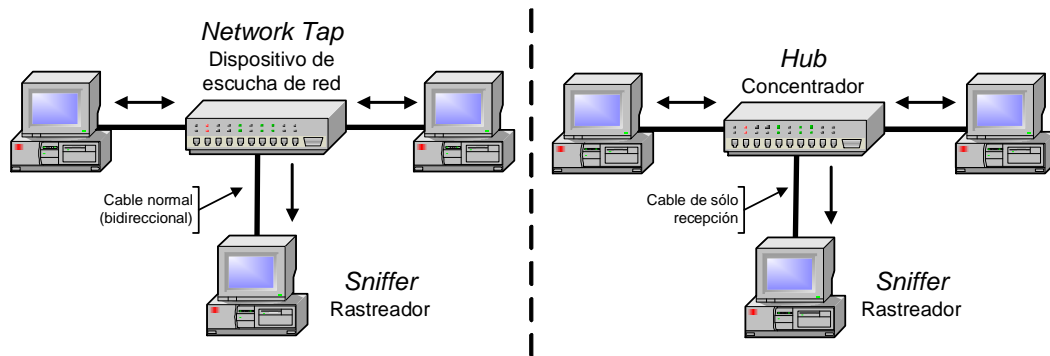


Figura 3-3 - Dispositivo de escucha de red y cable de sólo recepción

Por último, el tráfico de red puede revelar información sobre ataques que no podrían ser detectables por un monitor basado en máquina. Estos ataques podrían ser basados en paquetes malformados y algunos de denegación de servicio.

3.1.2.1 Paquetes de red

Para extraer los paquetes de un segmento de red, un detector de intrusiones basado en red, suele utilizar un dispositivo de red en *modo promiscuo*. Esto hace que el dispositivo de red genere una interrupción cada vez que detecta algún paquete en la red. Una máquina dedicada a monitorizar tráfico de red de esta manera, se suele denominar "sniffer" (rastreador). Este método es eficaz, pero tiene inconvenientes. Es útil en los casos en los que el dispositivo está situado en algún punto de la red en el que puede haber tráfico no destinado a sí mismo. Por ejemplo, en redes con "switches" (conmutadores), el modo promiscuo no es efectivo, ya que el dispositivo de red sólo recibe el tráfico destinado a él. Por otra parte, un rastreador tampoco puede monitorizar conexiones hechas con un módem, puesto que utilizan distintas interfaces.

En la Figura 3-4 se observa a la izquierda un escenario con concentrador ("hub"), en el que el rastreador puede recibir todo el tráfico relacionado con las máquinas que comparten el medio. En la siguiente situación (con conmutador), el rastreador sólo detecta el tráfico enviado o destinado a él mismo. Por último, se ilustra un tipo de conexión que el rastreador no es capaz de interceptar.

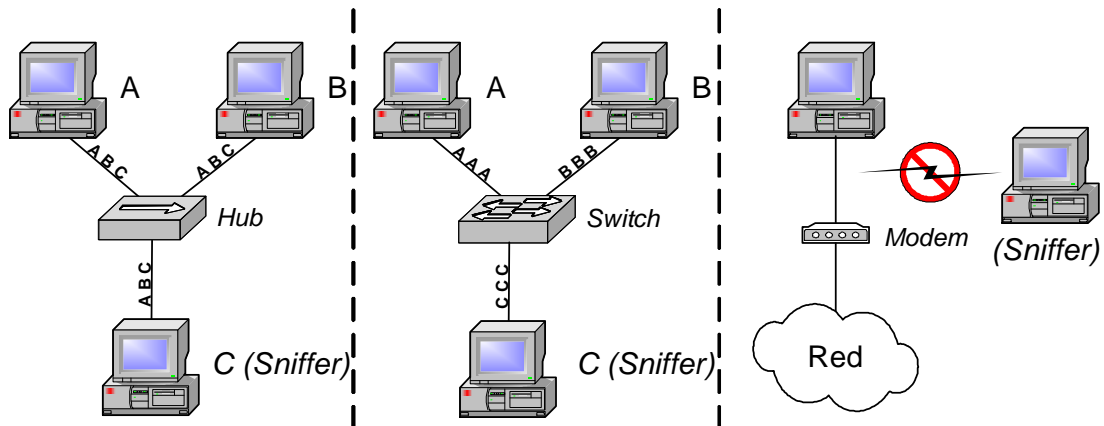


Figura 3-4 - Escenarios de conexión de un rastreador

3.1.2.2 Redes TCP/IP

Los protocolos TCP ("Transmission Control Protocol") e IP ("Internet Protocol") son los más ampliamente utilizados en Internet. Estudiar cómo funcionan y cómo se estructuran sus datos es imprescindible para comprender y evitar los problemas de seguridad relacionados con el tráfico de red.

En 1977 se empezó a utilizar TCP, desarrollado en 1974 por Kahn y Cerf, para sustituir al NCP ("Network Control Protocol") en ARPAnet. TCP era más rápido, fácil de usar y de implementar que su predecesor. En 1978, IP se añadió al TCP, encargándose del encaminamiento de los mensajes. Varios años más tarde, en 1983, cualquier elemento conectado a ARPAnet debía soportar los protocolos TCP/IP. Fue entonces cuando se empezó a referirse a ARPAnet y sus redes como "Internet". [14]

Las redes TCP/IP son de conmutación de paquetes. Cuando se establece una comunicación entre dos elementos de red, se produce un intercambio de un número determinado de paquetes entre ellos. Estos paquetes pueden viajar a través de una serie de segmentos de red, interconectados por dispositivos como puertas de enlace ("gateways") y "routers", que los encaminan hacia su destino.

3.1.2.3 Pila de protocolos

La suite de protocolos TCP/IP está compuesta por cuatro niveles o capas, de forma que cada uno utiliza los servicios del nivel inferior. Para comprender su mejor su estructura, la expondremos frente al modelo propuesto por OSI ("Open Systems Interconnection").

OSI es la estructura de protocolos en siete niveles propuesta por ISO e ITU-T ("International Telecommunication Union Telecommunication Standardization Sector"). Este modelo es teóricamente más elaborado y didáctico que TCP/IP, por el contrario, más sencillo y práctico. Desgraciadamente, aunque OSI tiene muchos puntos a su favor, no ha logrado sustituir a la arquitectura TCP/IP como estándar en Internet. La figura siguiente muestra una comparación entre ambas pilas de protocolos. [15]

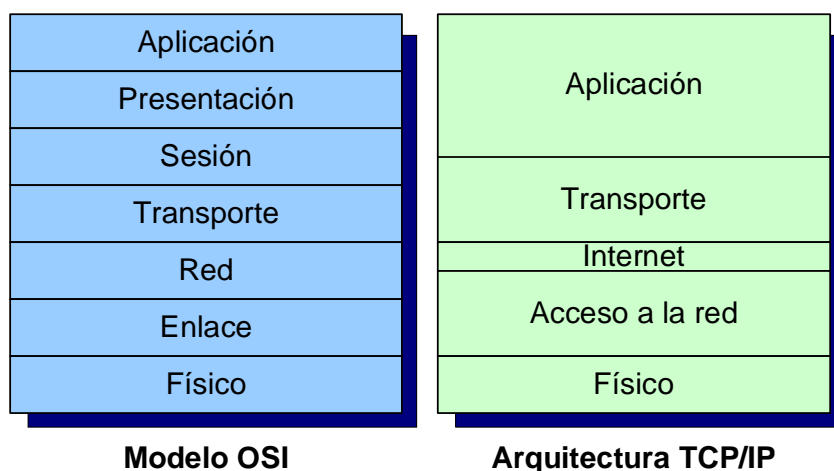


Figura 3-5 - Modelo OSI y Arquitectura TCP/IP

3.1.2.4 Estructura de una dirección IP

Las direcciones del Protocolo Internet en su versión 4 están compuestas por un número de 32 bits. Cada dirección está dividida en dos partes, el identificador de red y el identificador de máquina. El identificador de máquina se suele utilizar para designar diferentes subredes ("subnetting"), aprovechando mejor el espacio de direcciones y optimizando el encaminamiento.

Una de las formas habituales de representar una dirección IP es utilizando cuatro octetos, por ejemplo: 138.100.52.100. En muchos casos, una dirección IP se acompaña de una máscara de subred. Se han definido redes y máscaras estandarizadas, tal y como en la tabla siguiente. [16]

Clase	Formato (r=red, m=máquina)	Número de redes	Número de máquinas por red	Rango de direcciones de redes	Máscara de subred
A	r.m.m.m	128	16.777.214	0.0.0.0 - 127.0.0.0	255.0.0.0
B	r.r.m.m	16.384	65.534	128.0.0.0 - 191.255.0.0	255.255.0.0
C	r.r.r.m	2.097.152	254	192.0.0.0 - 223.255.255.0	255.255.255.0
D	grupo	-	-	224.0.0.0 - 239.255.255.255	-
E	no válidas	-	-	240.0.0.0 - 255.255.255.255	-

Tabla 3-5 - Redes y máquinas estándar en Internet

Según la notación CIDR ("Classless Inter-Domain Routing"), una dirección IP se indica mediante: dirección IP/número de bits de máscara de red. Por ejemplo: La dirección IP 138.100.52.100/25, indica que la dirección de máquina es 138.100.52.100 y la máscara de subred es 255.255.255.128 (primeros 25 bits a "1"; 11111111.11111111.11111111.10000000). Por lo tanto, la dirección de red es 138.100.52.0, que corresponde a la primera de las dos subredes.

Otra manera de representar una dirección IP es mediante la base de datos distribuida DNS ("Domain Name System"). DNS se encarga de hacer traducciones entre direcciones IP y nombres de dominio (como por ejemplo mailer.upm.es).

3.1.2.5 Estructuras de datos

El Protocolo Internet (IP) añade una cabecera a los datos que recibe del nivel superior, creando un *datagrama*, que es el mensaje que se envía en una red de comunicaciones de ordenadores por intercambio de paquetes. La estructura de la cabecera de un Datagrama Internet es la siguiente. [17]



Figura 3-6 - Cabecera de un Datagrama Internet

TCP también agrega su propia cabecera a los datos que recibe, con el siguiente formato. [18]

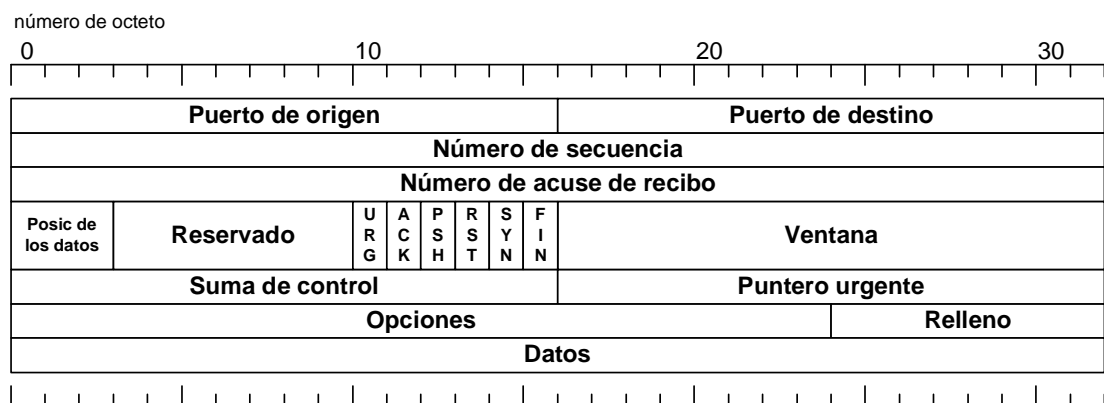


Figura 3-7 - Formato de la cabecera de TCP

La unidad lógica de datos TCP se denomina *segmento*. Por otra parte, denominaremos *paquete* al conjunto de datos con una cabecera que puede estar o no lógicamente completa (este término se suele referir más a un empaquetamiento físico de datos que lógico).

Cuando los paquetes llegan al dispositivo de red, son encaminados hacia el destino. Una vez son recibidos, a medida que pasan por cada nivel, se retiran las cabeceras correspondientes. En el nivel de transporte, se reconstruye el datagrama a partir de los segmentos recibidos.

3.1.2.6 Captura

Una vez explicadas las estructuras de datos más utilizadas en el tráfico de red, es de rigor tratar los métodos utilizados para extraer esa información de los segmentos de red.

Captura de paquetes en sistemas Windows

Existen numerosas opciones para extraer y analizar los paquetes que pasan por un dispositivo de red utilizando estos sistemas operativos. Spynet, Iris, Windump Ethdump, Ethload, son sólo algunos de los productos que se pueden utilizar para llevar a cabo esta tarea.

Con el desarrollo del "Systems Management Server" (SMS), apareció el "Microsoft Network Monitor". Este es el rastreador de paquetes ("packet sniffer") propuesto por Microsoft. Ofrece la capacidad de capturar paquetes de red con soporte para varios tipos de protocolos, un conjunto de filtros, y la interfaz de usuario común de Windows. También ofrece la posibilidad a una máquina remota conectarse y capturar los datos locales.

Captura de paquetes en sistemas UNIX

Los sistemas UNIX cuentan con una infraestructura en materia de redes bastante más amplia y robusta que Windows. Esto no es así de forma arbitraria. No en vano, fueron los entornos más utilizados para desarrollar la mayoría de las tecnologías de red.

Los primeros trabajos en materia de captura de paquetes se atribuyen al ordenador personal Xerox Alto, que ya contaba con un filtro para monitorizar tráfico Ethernet. Este filtro fue adaptado a UNIX en 1980 por un equipo de CMU y Stanford. El "CMU/Stanford Packet Filter" (CSPF) se utilizó con éxito en un ordenador DEC PDP-11. Estaba optimizado para los ordenadores de la época.

El "Lawrence Berkeley Nacional Laboratory", notando que había problemas de ajuste entre el CSPF y las modernas CPU basadas en RISC, desarrolló el "Berkeley Packet Filter" (BPF). Este filtro aportó importantes mejoras de rendimiento frente al CSPF.

El BPF utiliza dos componentes, un "network tap" (dispositivo de escucha de red) y un filtro de paquetes. Estos datos son enviados a las aplicaciones que están en modo de escucha. Entonces, el filtro procesa la información según los parámetros enviados, y muestra los datos resultantes.

Hay dos aplicaciones de red destacables que utilizan BPF, y son tcpdump y arpwatc. Tcpdump es una herramienta de monitorización de red y adquisición de datos que permite realizar filtrados, recopilación de paquetes y visualización de los mismos. Arpwatc monitoriza la actividad que involucra los mapeos de direcciones IP y Ethernet, y avisa a los administradores cuando detecta nuevos registros o actividades anormales. [19]

Por otra parte, muchos monitores de red y programas de detección de intrusiones utilizan libpcap. Se trata de una librería de captura de paquetes, usada también por tcpdump. Libpcap es una interfaz independiente del sistema que permite hacer monitorizaciones de red a bajo nivel. Su portabilidad a distintas plataformas, como a Linux, que cada vez es más utilizado en entornos de monitorización de red y detección de intrusiones, es una de sus características más relevantes. [20]

Captura de paquetes basada en STREAMS (flujos)

STREAMS es un modelo de programación de sistemas para el desarrollo de controladores de dispositivos. La mayoría de los sistemas UNIX lo soportan (entre ellos Sun Solaris, HPX, SCO UNIX y AIX de IBM). Un "stream" (flujo) tiene una estructura de tubería ("pipe") que permite intercambiar datos con controladores de dispositivo, mediante mensajes.

La posibilidad de captura de paquetes está incluida en las librerías. Esta solución no es tan óptima como BPF. Se suele utilizar en conjunción con una interfaz proveedora de enlace de datos (DLPI), que se utiliza para funciones de rastreo. [21]

3.1.2.7 Dispositivos de red

Aparte de los rastreadores de paquetes, los propios dispositivos de red pueden aportar información de valor en materia de detección de intrusiones. Por poner un ejemplo, las estadísticas de uso pueden revelar información sobre algún posible problema de seguridad. Hay que tener presente que siempre es preferible aprovechar las fuentes de datos existentes que crear nuevos mecanismos de recopilación de datos.

3.1.2.8 Fuentes de información externas

En esta categoría entran las fuentes de información que tienen un origen manual, externo al propio sistema. La mayoría de las veces se trata de intervenciones humanas. El valor informativo aportado por el factor humano es irremplazable. Ejemplos de este caso podrían ser entradas de registro manuales, describiendo fallos de hardware o del entorno de sistema, caídas de alimentación, análisis de eventos, o incluso comportamientos anómalos.

Un sistema de detección de intrusiones no se puede concebir en ningún caso como un sistema completamente autónomo, sino como una herramienta de diagnóstico. Permite a los administradores supervisar a alto nivel la actividad del sistema desde una perspectiva de seguridad, ayudándoles a prevenir y evitar posibles problemas.

3.1.3 Información de productos de seguridad

En la detección de intrusiones, mientras más fuentes de información existan, más posibilidades habrán de tener éxito. Muchos cortafuegos, sistemas de Inteligencia Artificial, dispositivos de seguridad y sistemas de control de acceso generan sus propios registros. Estos datos tienen evidentemente un valor importante en materia de seguridad, y junto con las otras fuentes de información ayudan a mejorar la eficacia del proceso de detección de intrusiones.

Además, los sistemas de seguridad, debido a su naturaleza, son los principales objetivos de los ataques bien planificados. Por esta razón, su monitorización y comprobación de su buen funcionamiento contribuyen a determinar la correcta estabilidad del sistema.

A continuación se describe a modo de ejemplo el formato de una entrada de registro de un producto de seguridad, concretamente el Firewall-1 de "Checkpoint Technologies". [22]

Número de campo	Nombre de campo	Contenido
1	Número	Identificador de transacción
2	Fecha	Fecha de evento
3	Hora	Hora de evento
4	Acción	Acceptar o denegar
5	Tipo	
6	Origen	
7	Alarma	
8	Nombre de interfaz	Dirección MAC o tarjeta Ethernet
9	Dirección de interfaz	Entrante / Saliente
10	Tipo de protocolo	TCP o UDP
11	"Host" origen	Dirección IP de origen
12	"Host" destino	Dirección IP de destino
13	Tipo de servicio	Tipo de servicio de red
14	Numero de puerto de origen	Puerto utilizado por el paquete
15	"Regla"	Regla de cortafuegos enviada
16	Tiempo transcurrido	Tiempo desde el comienzo de la sesión
17	Paquetes de esta sesión	Número de paquetes asociados a la sesión
18	Número de bytes	
19	Nombre de usuario autenticado	
20	Mensajes	

Tabla 3-6 - Formato de registro FW-1

La herramienta "fwlogsum" permite el análisis estadístico de registros generados por productos como firewall-1. Este tipo de información vuelve a poner de relieve la necesidad de algún sistema que permita la coordinación de fuentes de información de distintos orígenes. El uso del NTP ("Network Time Protocol") o alguna otra fuente fiable de información del tiempo, puede ser de gran ayuda.

3.1.3.1 Otros componentes como fuentes de datos

Entran en esta categoría otras fuentes de información, que no se consideran partes integrantes del sistema.

Uno de los ataques practicados contra los sistemas es el del *enmascaramiento* o suplantación de identidad. Consiste en robar la identidad de un usuario legítimo, normalmente el administrador, para entrar en el sistema. Este tipo de ataques no es fácil de detectar. Uno de los métodos para evitar este ataque, como se contempló en el capítulo 2, es el uso de técnicas estadísticas para detectar comportamientos anómalos. Es relativamente común recibir un número alto de falsos positivos mediante este método. No obstante, si se utiliza algún dispositivo externo (por ejemplo, un sistema de reconocimiento por vídeo), capaz de identificar al usuario que accede físicamente al sistema y enviar esta información al mismo, sería posible reducir de forma significativa las probabilidades de error ante este ataque.

Otro ejemplo válido es el de los ataques realizados a un sistema a través de una conexión telefónica. Una vez identificado el módem utilizado, se puede investigar el registro de la línea de teléfono implicada para averiguar la identidad del atacante. En este caso, también se ha utilizado una fuente de datos externa al sistema.

Como se ha observado, para la detección de posibles intrusiones, cualquier fuente de información puede aportar datos valiosos, y aunque no provenga de una parte legítima del sistema monitorizado, no debe ser subestimada.

3.2 Análisis

Una vez obtenidos los datos de las distintas fuentes de información en la fase de recopilación, se los somete a diversas técnicas de estudio en la fase de análisis. Para ello, la información se ordena cronológicamente, se clasifica, se evalúa de forma estadística y se identifica con patrones de actividad relativos a aspectos de seguridad. A continuación se detallarán los objetivos principales de la fase de análisis, los métodos utilizados y los requisitos necesarios para poder llevarla a cabo de forma eficaz.

3.2.1 Objetivos y elementos principales

Antes de continuar, para evitar confusiones, es conveniente definir el significado del elemento principal de esta etapa. *Análisis*, en el contexto de la detección de intrusiones, consiste en organizar y clasificar los datos sobre la actividad de usuario y de sistema para identificar actividades de interés [23]. En ocasiones esta actividad puede ser examinada mientras ocurre, o después de haber tenido lugar. También puede ser necesario recopilar cierta cantidad de información para poder realizar un estudio estadístico adecuado.

La Figura 3-8 describe un modelo general de gestión de seguridad.

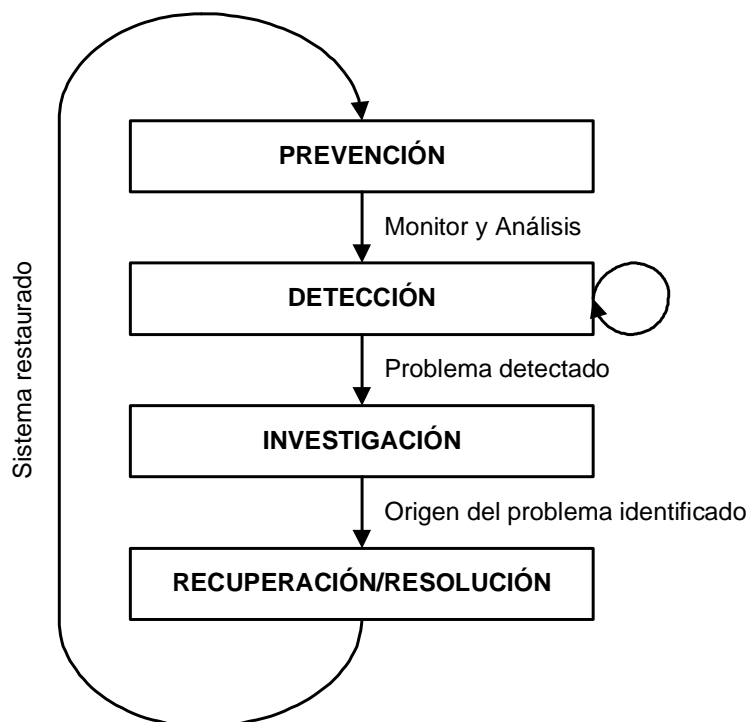


Figura 3-8 - Diagrama general de un modelo de gestión de seguridad

Como se puede observar, la detección de intrusiones corresponde al segundo paso. Este diagrama es útil porque algunos mecanismos de análisis proporcionan información necesaria para los estados de investigación y recuperación del modelo.

3.2.1.1 Objetivos principales

En líneas generales, la detección de intrusiones se utiliza para mejorar la seguridad de un sistema. El análisis se lleva a cabo por una serie de motivos, entre los cuales se podrían destacar los siguientes.

- **Determinar comportamientos:** Una de las funciones de análisis es el estudio del comportamiento o conducta mediante funciones estadísticas. Gracias a estas funciones, se puede determinar con un porcentaje aceptable de aciertos, cuál es comportamiento normal, para así, detectar posibles intrusiones a través de variaciones en el mismo.
- **Control de calidad para el diseño de seguridad y administración:** Mediante el análisis se pueden detectar posibles problemas en la gestión y el diseño de la seguridad del sistema. Esto se puede aprovechar para corregir estos fallos.
- **Información necesaria en intrusiones reales:** En determinadas ocasiones es necesario que la información sea suficientemente detallada y fiable para poder efectuar acciones legales contra los culpables.

3.2.1.2 Requisitos y objetivos secundarios

El análisis necesita que se cumplan al menos dos características para su funcionamiento. Una de ellas es la "accountability" (responsabilidad), esto es, la capacidad de relacionar una actividad con la persona u objeto responsable de ella. Naturalmente, esto requiere un sistema fiable de identificación y autenticación del sistema. Aunque este aspecto pueda resultar trivial a simple vista, no lo es. Resulta relativamente fácil identificar las actividades de un usuario a nivel de "host", pero realizar la misma tarea a nivel de red, cuando las actividades de un mismo usuario pueden tener origen en múltiples "hosts", requiere más tiempo de proceso y mecanismos más complejos.

La otra característica es la de la *detección en tiempo real y respuesta*. Esto implica reconocer con rapidez los eventos generados por un intruso para identificar un posible ataque, y reaccionar ante este, bloqueándolo (impidiendo las conexiones de red) o restaurando el sistema (ejecutando los comandos necesarios para llevar el sistema al estado de "pre-ataque").

El análisis también puede tener otros objetivos distintos a los mencionados arriba. La información puede ser utilizada para análisis forenses de red o de sistema. O puede ser necesaria para monitorizar y mejorar el rendimiento del sistema.

El siguiente paso fundamental después de determinar los objetivos y requisitos del análisis es establecer las prioridades. Las prioridades pueden ser ordenadas mediante diferentes métodos, como el horario, o el sistema ("los requerimientos de este sistema tienen preferencia sobre estos otros"). En ocasiones, algunas prioridades entran en conflicto con otras. Puede ser que almacenar los datos según unos determinados requisitos de nivel de seguridad, provoquen una caída de rendimiento que no permita efectuar análisis suficientemente rápidos.

3.2.1.3 Factores de detección

Los elementos que intervienen en el análisis de la detección de intrusiones son de diversa naturaleza. Es necesario comprenderlos mejor antes de pasar a los siguientes apartados.

- **Factor humano:** Los seres humanos son la fuente de información de intrusiones más común y tradicional. En lo que al análisis se refiere, existen estudios demuestran que en un escenario desarrollado durante un período de tiempo, en el que un sistema de detección de intrusiones es capaz de detectar miles de intrusiones, el ser humano sólo es capaz de detectar alrededor del 2 por ciento de las mismas.
- **Eventos externos:** Cualquier evento externo al sistema puede aportar información sustancial al análisis. La contratación o el despido de empleados clave en la gestión de seguridad, un inusual crecimiento de informes de anomalías de un determinado sistema o los resultados de baterías de pruebas de vulnerabilidades contra sistemas, son sólo algunos de estos ejemplos.
- **Preámbulos de intrusiones:** Se puede averiguar si un sistema va a ser atacado si presenta ciertas evidencias de intrusiones. Algunas de estos síntomas pueden ser la instalación de algún troyano, la adición de nuevas cuentas de usuario no autorizados al sistema o la aparición de nuevos "hosts" en el fichero de equipos de confianza (en el fichero `/etc/.rhosts` en sistemas UNIX, o el fichero `%WINDIR%\system32\drivers\etc\hosts` en sistemas Windows). Uno de los primeros proyectos de seguridad en contemplar este tipo de problemas fue COPS.

- **Artefactos de intrusiones:** Los ficheros de registro de un rastreador de contraseñas, fallos inexplicables del sistema, un inusual aumento del consumo de recursos o la aparición de ficheros dañados son ejemplos válidos de estos artefactos, es decir, posibles evidencias de intrusiones. Pueden utilizarse en análisis en tiempo real o en proceso por lotes. Se puede observar que algunos artefactos no han sido pensados para detectar intrusos. No obstante, ayudan en este propósito y a descubrir posibles vulnerabilidades.
- **Tiempo:** La posibilidad de realizar monitorizaciones en tiempo real, frente al tradicional modelo basado en proceso por lotes ("batch mode based"), hizo que la detección de intrusiones diera un paso adelante, creando nuevas vías de trabajo como mecanismos capaces de reaccionar de forma activa a ataques.

3.2.2 Modelos

Tanto la variedad como las posibilidades de los modelos utilizados en el análisis de la detección de intrusiones son enormes. Pueden ser tan simples como un sencillo "script" que procese los registros de sistema, descartando los más comunes, o tan complejos como un elaborado sistema no paramétrico entrenado con millones de transacciones.

El análisis de detección de intrusiones se puede efectuar de múltiples formas. A continuación, se mostrará el modelo propuesto por Bace [23], que intenta cubrir todas las formas de búsqueda de evidencias de intrusiones en los registros de eventos del sistema. Esto estructura el modelo en tres secciones principales: construcción del analizador, realización del análisis, y refinamiento o reestructuración del proceso. Las dos primeras secciones se dividen a su vez en dos partes: preproceso de datos y posproceso.

No toda la actividad que hay en un sistema es *normal*. Existe lo que se denominan *anomalías* ("anomalies"), que son actividades poco comunes, y *usos indebidos* ("misuse"), es decir, comportamientos no permitidos. La distinción entre unas y otras es absolutamente imprescindible durante el análisis. La Figura 3-9 representa la relación que hay entre los distintos tipos de actividades de un sistema. No existe consenso en la comunidad de profesionales de seguridad en cuanto al tamaño de área de intersección entre actividades normales y usos indebidos del sistema. Algunos expertos piensan que área de intersección es mínima, mientras que otros afirman que existe una importante relación entre las dos. El debate sigue abierto, por lo que existen distintas vías de trabajo al respecto.

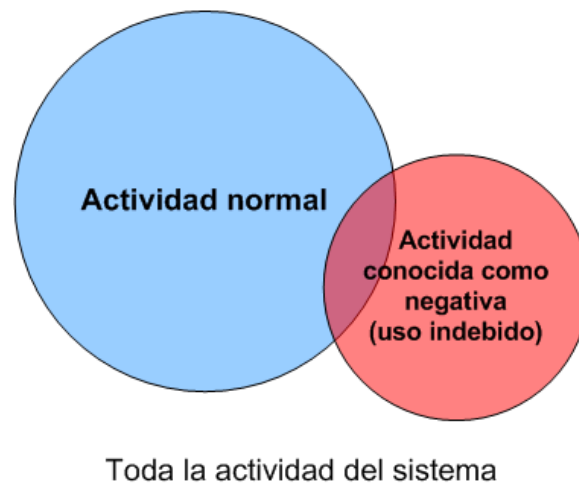


Figura 3-9 - Actividades de un sistema: usos indebidos y anomalías

3.2.2.1 Construcción del analizador

El primer paso en el modelo es la construcción del motor de análisis. Este elemento se encarga del preproceso, clasificación y posproceso de la información. Dada su naturaleza, es necesario que esté bien adaptado al entorno en el que va a trabajar.

3.2.2.1.1 Recopilación y generación de eventos

Lo primero que hace el analizador es recopilar eventos. Esta información puede haber sido generada en un entorno de producción, en un laboratorio, o incluso hecha a mano por algún desarrollador que comprueba especificaciones.

- **Detección de usos indebidos:** En este caso, la recopilación se centra a información relacionada con la intrusión, incluyendo datos de vulnerabilidades, ataques, amenazas, herramientas para escanear puertos, y otras áreas de interés. También se recoge cualquier elemento sobre políticas de sistema o procedimientos que ayuden a la siguiente fase del modelo, el análisis, a discernir sobre problemas de seguridad relacionados con la organización o ataques externos.
- **Detección de anomalías:** Para este tipo de detección, se recogen los eventos generados por el sistema, o por algún objetivo similar. El objetivo en este caso es poder crear un perfil, a partir de esta información, que permita diseñar un patrón de comportamiento "normal".

3.2.2.1.2 Preproceso

Después de la recopilación de los datos, se someten a una transformación, *preparando* los mismos. En algunos casos es necesario convertirlos a un formato *canónico*. Este formato suele estar integrado en el propio diseño del analizador. El término "canónico" se aplica en este caso a un formato único de estructura de datos. Hecho originado por la necesidad de facilitar la labor al motor de análisis en un entorno en que coexisten distintos sistemas operativos, cada uno con su formato de eventos nativo. En redes en las que trabajan sistemas operativos homogéneos este concepto no es necesario.

- **Detección de usos indebidos:** Aquí, el preproceso de datos suele consistir en la transformación de los eventos de forma que se corresponda con los datos a filtrar cuando se ponga en marcha el sistema de detección. Así, los indicios de ataques o las violaciones de las políticas de seguridad pueden ser convertidas en patrones o firmas reconocibles; y en los sistemas de detección basados en red, los paquetes pueden ser almacenados para reconstruir sesiones TCP.
- **Detección de anomalías:** En esta detección, los eventos se pasan a "arrays" o tablas, con algún tipo de información distintiva, como el nombre del proceso. Después de esto, se traducen a formas numéricas. El hecho de trabajar con datos numéricos reduce las necesidades de espacio, a la vez que mejora las capacidades de búsqueda e identificación de patrones en busca de problemas de seguridad.

3.2.2.1.3 Construcción de un motor o modelo de clasificación de comportamiento

Una vez se tienen los datos formateados, se hace la clasificación. Aquí se distinguen los datos que indican posibles intrusiones de los que no presentan riesgo alguno.

Detección de usos indebidos

La información se clasifica en comportamientos traducibles en patrones o reglas. Estas reglas pueden estar compuestas de una sola firma (llamadas *atómicas*), o de múltiples firmas (denominadas *compuestas*). Un ejemplo de regla atómica es la que detecta paquetes IP malformados (como aquellos con valores de campos no reconocidos por los RFC). Una regla compuesta podría consistir en la detección de un ataque a un servidor web utilizando un "exploit" consistente en una secuencia de órdenes HTTP, o un escáner que abra varios puertos que atiende el servidor.

Una de las posibles estructuras que podría tener un detector de usos indebidos es la de un sistema experto ("expert system"). Por una parte, contiene una base de conocimientos ("knowledge base") con los comportamientos sospechosos recogidos de intrusiones pasadas, y por otra, cuenta con reglas que permiten su identificación. Reglas que normalmente consisten en sentencias "if-then-else".

Otra de las estructuras más comunes es la del motor de comprobación de patrones, que representa las intrusiones como firmas de ataques (patrones) contra los que se comprueban los datos de auditoría.

Detección de anomalías

Consiste en la elaboración de perfiles estadísticos de comportamiento a lo largo del tiempo. Estos perfiles se construyen mediante determinados algoritmos, capaces de detectar cambios graduales en los patrones de conducta de los usuarios.

El "Intrusion Detection Expert System" (IDES), originado por el modelo de Denning, es un ejemplo de detector de anomalías. IDES define el comportamiento en términos de *medidas* ("measures"): aspectos de la conducta de usuario en los sistemas monitorizados. La Tabla 3-7 muestra la clasificación de estas medidas. Las *ordinales* o *continuas* se expresan en forma de cantidad numérica o cuantificada. Las *categorías* o *discretas* se expresan en forma de identidad y frecuencia de suceso.

Las categóricas están divididas en dos tipos, *binarias* y *lineales*. Las binarias indican si una medida ha ocurrido o no (positivo/falso). Las lineales se expresan en términos de cantidades, indicando el número de veces que un determinado comportamiento ha tenido lugar. [24]

	Ordinal (continua)	Categórica (discreta)
Binaria	Tiempo utilizado de CPU. Número de registros de auditoría producidos.	Si un directorio fue utilizado. Si un fichero ha sido accedido. Si los registros de auditoría indicaron el uso por día/semana/mes.
Lineal		Nº de veces que cada comando fue utilizado. Nº de errores de sistema. Nº de fallos de entrada en la última hora. Nº de eventos de auditoría registrados. Nº de ficheros modificados.

Tabla 3-7 - Clasificación de medidas, con ejemplos, de IDES

3.2.2.1.4 Suministrar datos al modelo

Después de construir el modelo, se le pasan los datos preprocesados. Es entonces cuando se crea el motor de análisis.

- **Detección de usos indebidos:** A los detectores de usos indebidos se le suministran los eventos provenientes de una base de conocimiento de ataques; una recopilación de ataques en un formato que el analizador puede entender.
- **Detección de anomalías:** A los detectores de anomalías se le suministran los datos de referencia recopilados, permitiendo la creación de perfiles de usuario. Se suele asumir el hecho de que estos datos están libres de intrusiones, lo que implica una previa labor de comprobación de los datos recogidos.

3.2.2.1.5 Almacenar el modelo abastecido en una base de conocimientos

Este modelo constituye la base del motor de análisis. Una vez suministrados los datos pertinentes, se almacena en una localización determinada, listo para ser utilizado.

3.2.2.2 Realización del análisis

Esta es la segunda fase del analizador. Aquí es donde el analizador se aplica al flujo de datos en tiempo real para la detección de intrusiones.

- **Nueva entrada de registro de evento:** Lo primero que ocurre en el análisis es la llegada de un nuevo evento generado por alguna fuente de información, la cual se asume que no ha sido comprometida. El origen de este evento puede ser muy variado, como un paquete de red, un registro de auditoría de sistema,...
- **Preproceso:** Tal y como sucedía en la construcción del analizador, aquí también es necesario tratar los datos según las necesidades del motor de análisis. Este tratamiento puede consistir en la extracción de las cabeceras TCP de varios mensajes para definir

un nivel de abstracción superior, reconstruyendo una *sesión*. También se puede utilizar la información de los identificadores de proceso para elaborar un esquema de proceso jerárquico superior.

- **Detección de usos indebidos:** Para la detección de usos indebidos, los datos se pasan a algún tipo de formato canónico, de forma que se puedan aplicar a los patrones de ataques.
- **Detección de anomalías:** En este caso, los datos son convertidos a perfiles numéricos con comportamientos expresados en forma de puntuaciones e indicadores.
- **Contrastar los registros de eventos con la base de conocimiento:** Ahora, el evento ya formateado, es comparado con la base de conocimiento. A partir de este paso, se podrán dar dos circunstancias. Una vez hecha la comparación del suceso con la base de conocimiento, si hay alguna coincidencia, el evento indica una intrusión y será registrado. Si, por el contrario, no existe, se desechará. Hecho esto, se repetirá la operación con el siguiente.
- **Detección de usos indebidos:** Para encontrar signos de usos indebidos, los eventos son comparados con un motor de comprobación de patrones. Si el motor encuentra una igualdad, genera una alarma. No obstante, algunos motores almacenan una equivalencia parcial (que puede indicar un posible ataque compuesto por múltiples eventos) y esperan a los siguientes datos en busca de una decisión más completa.
- **Detección de anomalías:** El contenido de los perfiles de comportamiento es comparado con el perfil histórico del usuario correspondiente. Según el algoritmo de análisis utilizado, se comprobará con mayor o menor exactitud el grado de similitud entre el comportamiento y el historial del usuario, generando en caso necesario la pertinente alarma.
- **Generar una respuesta:** Se obtiene una respuesta en caso de detectar una intrusión. La respuesta viene determinada por el tipo de analizador utilizado; puede ser una entrada de registro de sistema, una alarma, una acción de respuesta automática programada por el administrador, ...

3.2.2.3 Refinamiento y reestructuración

Algunos analizadores cuentan con una tercera fase o estado, que se ejecuta de forma simultánea con la fase principal. Consta de elementos de mantenimiento del motor de análisis y de modificación de ciertas funciones del analizador, como los patrones.

3.2.2.3.1 Detección de usos indebidos

En este tipo de detección, las funciones se centran básicamente en la modificación de los patrones para identificar nuevos tipos de ataque. Esta actualización puede ser automática o manual. Independientemente del método utilizado, se suele poder hacer "al vuelo"; no es necesario interrumpir el proceso de análisis mientras se hace la actualización.

En algunos casos, este estado se usa en conjunción con otro cuyo objetivo es el de optimizar situaciones de retención de estados, eliminando eventos que no han sido resueltos. Uno de los elementos a tener en cuenta en el análisis de usos indebidos es el tiempo. El término "event horizon" (horizonte de evento) se aplica al tiempo límite aplicable a una característica de sistema determinada, como por ejemplo la diferencia de tiempo entre dos entradas a un sistema (para detectar intentos de acceso por fuerza bruta).

3.2.2.3.2 Detección de anomalías

En este tipo de análisis, los perfiles históricos de comportamiento son actualizados de forma regular, según el tipo de motor. En el caso de IDES, los perfiles son actualizados diariamente. Después de esto, el resumen de las estadísticas de comportamiento de cada usuario es añadido a la base de conocimiento, y se eliminan las estadísticas más antiguas, como las de más de un mes. Además, a cada día se le aplica un factor de multiplicación inversamente proporcional a su antigüedad, para dar mayor valor a los comportamientos más recientes que a los ocurridos hace más tiempo. [24] Esto se hace para adaptar el sistema a los cambios de conducta graduales de cada usuario, con la intención de reducir los falsos positivos, sin que con ello se disminuya la capacidad de detección del analizador.

3.2.3 Técnicas

Una vez descrito un modelo general que abarca los tipos de análisis en la detección de intrusiones, se describirán algunas de las soluciones empleadas en este campo. Como ya se comentó en apartados anteriores, la fase de análisis se divide en dos categorías principales: la *detección de usos indebidos*, que compara los datos con patrones en busca de violaciones de seguridad, y la *detección de anomalías*, que se vale de la rama estadística de las matemáticas para identificar comportamientos sospechosos.

3.2.3.1 Detección de usos indebidos

Un detector de usos indebidos es, a grandes rasgos, un comparador de patrones. Para que funcione correctamente necesita: una base de conocimiento con patrones fiables, una serie de eventos para poder ser analizados, y un eficaz motor de análisis.

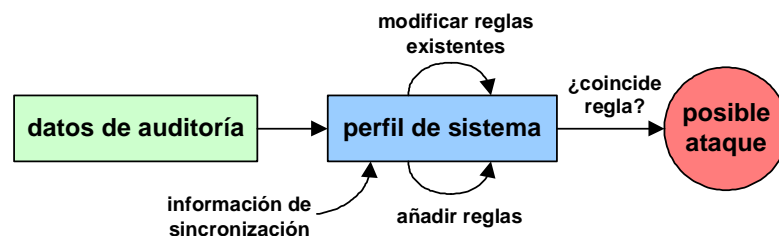


Figura 3-10 - Modelo general de un detector de usos indebidos

La detección de usos indebidos, por su propia naturaleza, tiene una limitación importante. Y es que sólo puede identificar problemas de seguridad cuando ya se han definido en su base de conocimiento. Esto significa que se han de conocer de antemano los métodos y ataques utilizados.

O bien, ser lo suficientemente inteligente como para poder anticiparse a nuevos ataques, intentando describirlos antes de que sucedan.

La detección de usos indebidos se puede implementar de las siguientes formas:

- **Sistemas expertos:** Aquellos que reconocen ataques o intrusiones mediante reglas "if-then-else".
- **Sistemas de razonamiento basados en modelos:** Usan modelos de usos indebidos junto con mecanismos de razonamiento para identificar la ocurrencia de una intrusión.
- **Análisis de transiciones de estados:** Hacen uso de grafos o autómatas finitos para representar y utilizar patrones de ataque.
- **Monitorización de pulsación de teclas:** Registran la actividad de periféricos manuales, como el teclado, para obtener información que pueda ayudar a la detección de usos indebidos.

3.2.3.1.1 Sistemas expertos o de producción

Un sistema experto, según P. Jackson, es un programa de ordenador que representa y razona con la información de un determinado tema especializado con el objeto de resolver problemas o de dar consejos [25]. Estos sistemas fueron utilizados por los primeros productos de detección de usos indebidos. Algunos ejemplos son MIDAS, IDES, "Next Generation IDES" (NIDES), DIDS, o CMDS. Sistemas de detección de intrusos basados en red como Snort y Bro, comentados más adelante, son detectores de patrones que también utilizan esta técnica.

Los sistemas de producción utilizan reglas "if-then-else" para examinar los datos. Realizan análisis mediante funciones internas al sistema, de forma completamente transparente al usuario. Antes de la existencia de los sistemas expertos, los programadores tenían que diseñar sus propios motores de decisión, lo que implicaba un trabajo adicional. Además, el hecho de que cada uno tuviera que implementar sus propios motores y reglas, impedía la llegada de una solución estándar.

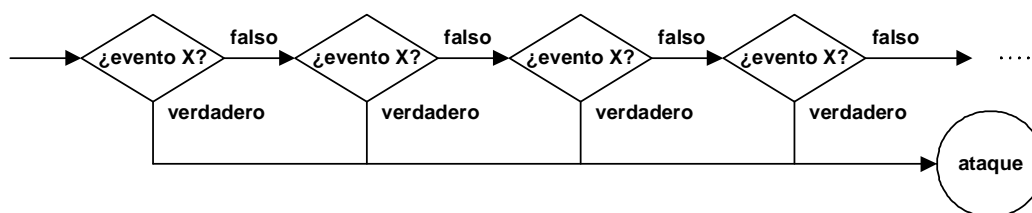


Figura 3-11 - Ejemplo de diagrama if-then-else

Una de las ventajas más importantes de utilizar reglas "if-then" es que mantiene separados el control de razonamiento y la formulación de la solución del problema. En lo que respecta a los usos indebidos, permite deducir una intrusión a partir de la información disponible.

La principal desventaja que se plantea es que los patrones no definen un orden secuencial de acciones. Detectar mediante este método ataques compuestos por una sucesión eventos encierra grandes dificultades. Por otra parte, la labor de mantenimiento y actualización de la base de datos es otros de los puntos críticos de estos sistemas.

Snort

Snort es uno de los más populares sistemas de detección de intrusiones. Aunque soporta algunas funciones de detección de anomalías, es principalmente un detector de intrusiones de red basado en reglas. Esta herramienta es Software Libre y es capaz de realizar análisis de tráfico en tiempo real, así como registrar paquetes en redes IP. Esta herramienta fue creada por Marty Roesch, que actualmente dirige el equipo de desarrollo, compuesto por expertos provenientes de diversas entidades como Sourcefire, CERT, Nitro Data Systems o CodeCraft Consultants.

Este sistema tiene la ventaja de funcionar bajo gran variedad de plataformas. Está basado en las librerías `libpcap`¹, de modo que admite cualquier plataforma que acepte las mismas, como las enumeradas en la siguiente tabla.

i386	Sparc	M68k/P PC	Alpha	Other	
X	X	X	X	X	Linux
X	X	X			OpenBSD
X			X		FreeBSD
X		X			NetBSD
X	X				Solaris
	X				SunOS 4.1.X
				X	HP-UX
				X	AIX
				X	IRIX
			X		Tru64
		X			MacOS X Server
X					Win32 (Win9x/NT/2000/XP)

Tabla 3-8 - Plataformas soportadas por Snort

Snort puede realizar análisis de protocolos, búsqueda y comparación de contenidos, y puede detectar gran variedad de ataques y sondeos, tales como desbordamientos de "buffer", escaneo sigiloso de puertos ("stealth port scans"), ataques CGI, sondeos SMB, intentos de identificación de Sistema Operativo ("OS fingerprinting"), etc.

Para su labor, utiliza un lenguaje flexible de reglas para describir el tráfico de red que debe recoger o dejar pasar, además de un motor de detección que utiliza una arquitectura de "plugins" modular. Este sistema tiene también capacidades de alarma en tiempo real, y soporta diversos mecanismos de alarma, a través del "syslog", un fichero específico, sockets UNIX, mensajes WinPopup a clientes Windows, etc.

¹ `libpcap` es un interfaz independiente del sistema, para la captura de paquetes de nivel de usuario, escrito en el Lawrence Berkeley National Laboratory.

Snort tiene tres usos principales. Puede utilizarse como un rastreador de paquetes ("sniffer") de forma similar a tcpdump¹, como registrador de paquetes (útil para depuración de tráfico de red), y como detector de intrusiones de red.

Snort está escrito en código fuente abierto, y se puede obtener en su sitio oficial. [26]

Bro

Bro es un detector de intrusos en tiempo real basado en red, creado por Vern Paxson, del Laboratorio Nacional de Lawrence Berkeley y "AT&T Center for Internet Research at ICSI, Berkeley" en California. Trabaja monitorizando de forma pasiva el tráfico de red. Fue diseñado teniendo en mente estos objetivos principales:

- Monitorización de redes de alta velocidad; tipo FDDI (Fiber Distributed Data Interface), a 100 Mbps.
- Notificación en tiempo real.
- Separación entre mecanismo y políticas.
- Extensibilidad.

Para ello, el diseño de Bro está dividido en un *motor de eventos* que reduce el flujo de tráfico de red filtrado, a una serie de eventos de alto nivel. Por otro lado, utiliza un *intérprete de guiones de políticas* que analiza los eventos, escritos en el lenguaje especializado de Bro, para expresar las políticas de seguridad del sistema. En la siguiente figura se muestra la estructura general del sistema.

¹ tcpdump es una herramienta, basada en libpcap, que permite la captura y filtrado del tráfico (mediante expresiones booleanas) que pasa por un dispositivo de red, mostrando las cabeceras de los paquetes, o escribiendo el contenido de los campos de datos en un fichero de registro indicado.

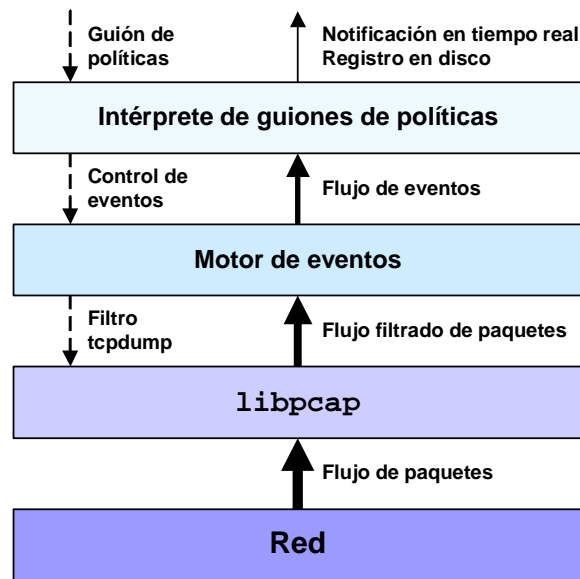


Figura 3-12 - Estructura de Bro

Bro no sólo detecta ataques realizados a través del tramo de red que monitoriza, sino que también contempla la posibilidad de ser en sí mismo un potencial objetivo de ataques. Para ello, cuenta con mecanismos específicos para su detección y defensa. Algunos de los ataques que reconoce son:

- Sobrecarga: cuyo objetivo es sobrepasar la capacidad de proceso del detector.
- Caída: provocan fallos en el monitor, o lo dejan sin recursos de sistema.
- Subterfugio: intentan engañar al monitor mediante el envío de paquetes TCP con sumas de control inválidas, o paquetes IP cuyo TTL (Tiempo de vida) es suficiente para llegar al monitor, pero no para llegar a su destino.

Aparte de esto, y para ampliar su grado de efectividad, Bro también cuenta con capacidades de proceso específico de algunas aplicaciones de red, tales como: Finger, FTP, Portmapper, Ident, Telnet y Rlogin.

Bro se distribuye bajo una licencia tipo BSD; es Software Libre, pero no copyleft. [27]

3.2.3.1.2 Sistema basado en modelos

Este sistema, que consiste en una variación de la detección de usos indebidos, fue propuesto por T.D. Garvey y T. Lunt. Utiliza una base de datos de escenarios de ataque, en la que cada escenario consiste en una secuencia de comportamientos que forman el ataque. En cada momento, el sistema analiza subconjuntos de comportamientos de su base de datos que coincidan con los que está experimentando en ese instante. Utiliza los resultados para identificar y anticipar posibles ataques (*anticipador*). El anticipador determina las posibles hostiles y las envía al *planificador*. Este planificador determina el grado de acierto entre el comportamiento recibido y el que figura en los registros de auditoría, y traduce los resultados en registros de auditoría del sistema.

La ventaja de este modelo radica en que está basado en una teoría matemática que utiliza el principio de incertidumbre. El diseño del *planificador* es independiente de la sintaxis del registro de auditoría. Además de esto, esta solución consume poco tiempo de proceso por cada registro de auditoría generado.

Un inconveniente que presenta este modelo es que depende de la buena pericia del encargado de seguridad a la hora de diseñar patrones creíbles y precisos. Por otra parte, el modelo no especifica claramente la forma en que se deben compilar los comportamientos en el planificador para que sea más eficiente, lo que afecta al rendimiento del detector. Esto no es una debilidad inherente al modelo, pero es algo a tener en cuenta en la implementación.

3.2.3.1.3 Transiciones de estados

El uso de transiciones de estados ("state transitions") para la detección de usos indebidos permite trabajar con técnicas avanzadas de comprobación de patrones. Se utilizan estados y transiciones del sistema para definir y encontrar intrusiones.

Entre los métodos que adoptan esta metodología destacan tres: el análisis de transiciones de estados, las "Colored Petri Nets" (CP-Nets), y el "Application Programming Interface" (API).

Análisis de transiciones de estados

Este análisis utiliza diagramas de transiciones de estados que representan ataques conocidos para la detección de usos indebidos. Este método fue utilizado por primera vez por el sistema STAT [28], portado a redes UNIX bajo el nombre de USTAT [29]. Estos sistemas fueron creados en la Universidad de California.

Los diagramas de transición, como ilustra la Figura 3-13, son representaciones gráficas de escenarios de intrusiones. Utilizan autómatas finitos (grafos) para los ataques. El paso por cada estado depende de que se cumplan o no una serie de características.

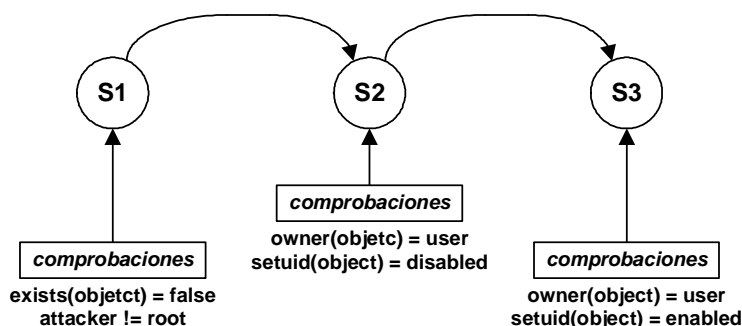


Figura 3-13 - Diagrama de transiciones de estados

Los nodos representan los estados, y las flechas (arcos) las transiciones. Utilizar diagramas de transición facilita la asociación entre los estados y los distintos pasos que realiza un intruso desde que entra en un sistema, con privilegios limitados, hasta que se hace con el control del mismo.

Los estados del diagrama indican situaciones particulares de un sistema, a las que se puede llegar de distintas maneras. De esta forma, varios atacantes pueden confluir en el mismo estado habiendo realizado diferentes pasos.

El estado inicial representa el estado del sistema antes de ser comprometido. La intrusión se produce cuando se llega al último estado del diagrama. Las transiciones ocurren por acciones del usuario. El sistema comprueba su esquema de transiciones para determinar a qué estado se llega. Si una acción determinada no lleva a ningún estado en concreto, el sistema devuelve al usuario al estado más cercano en que estaba. Si las acciones llevan al usuario al estado final de un diagrama, el sistema envía una alarma al responsable de seguridad con las acciones tomadas en la última transición.

Algunas de las ventajas del STAT son las siguientes:

- Los diagramas de transición permiten hacer una representación a alto nivel de escenarios de penetración.
- Las transiciones ofrecen una forma de identificar una serie de patrones que conforman un ataque.
- El diagrama de estados define la forma más sencilla posible de definir un ataque. Así, el motor de análisis puede utilizar variantes del mismo para identificar ataques similares.
- El sistema puede detectar ataques coordinados y lentos.

Estas son algunos de los inconvenientes del STAT:

- Los diagramas de transición y las firmas o patrones son creados a mano.
- El lenguaje utilizado para describir los ataques es demasiado limitado, y en ocasiones puede resultar insuficiente para recrear ataque más complejos.
- El análisis de algunos estados puede requerir más datos del objetivo, por parte del motor. Esto reduce el rendimiento del sistema.
- Las limitaciones de este sistema hace que no pueda detectar algunos ataques comunes, siendo necesario el uso de motores de análisis adicionales.

IDIOT y "Colored Petri Net"

"Coloured Petri Nets" (CP-nets o CPNs) es un lenguaje orientado a objetos para el diseño, especificación y verificación de sistemas. Es especialmente apropiado para sistemas formados por gran variedad de procesos que necesitan estar comunicados y sincronizados. Algunas áreas típicas en las que se aplica este lenguaje son sistemas distribuidos, protocolos de comunicación, o sistemas de producción automática.

La siguiente figura muestra algunas de los modelos de operaciones contemplados por el lenguaje Petri-net.

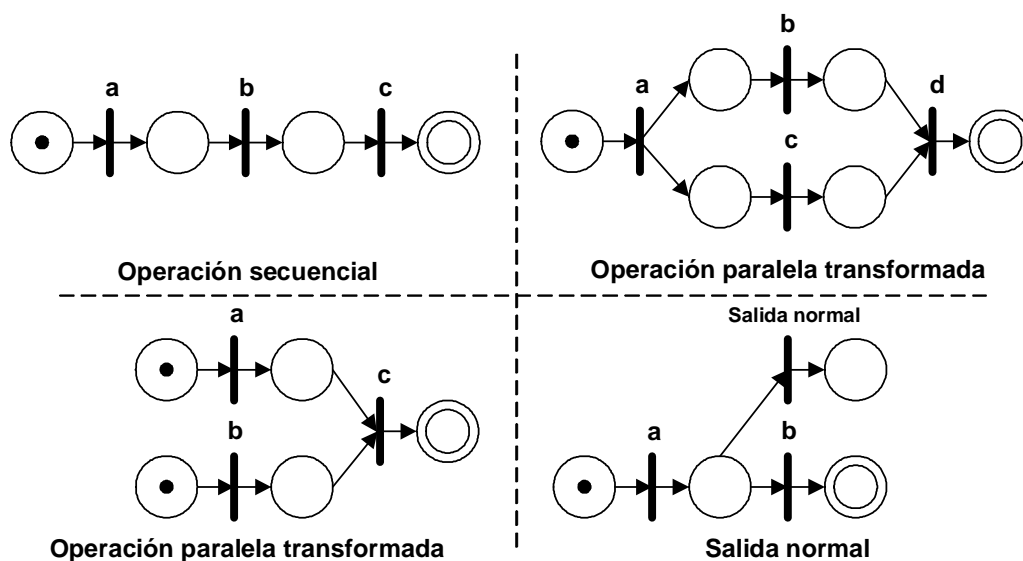


Figura 3-14 - Modelo Petri-net

IDIOT¹ fue desarrollado en la Universidad de Purdue, y representa sus patrones de intrusiones mediante el lenguaje de estados y transiciones definidos por CPNs. [31]

Aunque el sistema STAT y el IDIOT puedan parecer iguales, no es así. Existen importantes diferencias entre ambos puntos de vista. En STAT las intrusiones son detectadas por el impacto que tienen sobre el estado del sistema, en IDIOT las intrusiones se detectan mediante la comparación de los patrones que forman el ataque. En STAT, los detectores ("guards") están ubicados en los estados, mientras que en IDIOT están en las transiciones.

En el sistema IDIOT las intrusiones se expresan en forma de patrones que representan relaciones entre los eventos de sistema y su contexto. Este método es independiente de la arquitectura del sistema, y permite representar cualquier categoría de intrusiones. El siguiente ejemplo consiste en un patrón creado mediante el lenguaje Petri-net para la detección de repetidos intentos de acceso fallido.

¹ El nombre IDIOT fue utilizado como un chiste por su autor. Significa "Intrusion Detection In Our Time" (Detección de Intrusiones en nuestro tiempo); refiriéndose a los retrasos que muchos desarrolladores de detección de intrusiones deben soportar para poder implementar sus desarrollos en sistemas de producción..

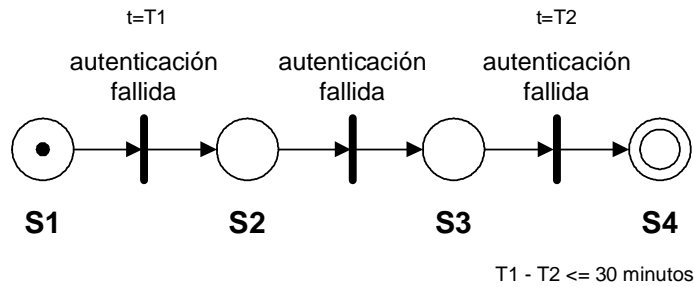


Figura 3-15 - Ejemplo de patrón de ataque mediante el lenguaje Petri-net

Una de las ventajas del sistema IDIOT es que es muy rápido; consume menos del 5% del tiempo de CPU en una estación Sun SPARC 5 que genera alrededor de 6 MB por hora. El motor encargado de la comparación de patrones es independiente de la plataforma en que trabaja. Esto hace que las firmas que utiliza se puedan intercambiar entre distintos sistemas, independientemente de la sintaxis de registros de auditoría usada. Además, soporta respuestas automáticas; permite especificar las acciones que se ejecutarán en caso de detectar intrusiones.

Las desventajas son las mismas que tienen los sistemas de detección de usos indebidos. No puede detectar lo que no está definido en sus bases de patrones.

3.2.3.1.4 Soluciones basadas en Lenguaje/API

Otra de las formas de abordar el problema de los usos indebidos es definir lenguajes específicos para los motores de detección de intrusiones. Esta opción es mejor que utilizar otros lenguajes ya existentes como P-BEST o CLIPS, diseñados para otros fines.

Algunas de las soluciones basadas en este método son el lenguaje RUSSEL, desarrollado en la Facultad Universitaria de Notre-Dame de la Paix [32]; el sistema STALKER, ilustrado en la figura de abajo, patentado por Smaha y Snapp de los Laboratorios Haystack [33]; el lenguaje de filtrado de paquetes N usado como parte del NFR (Network Flight Recorder) [34]; el lenguaje ASL, escrito en la Universidad de Iowa; o el IMDL (Intrusion Detection Markup Language), creado en la Universidad Nacional de Chiao Tung [35].

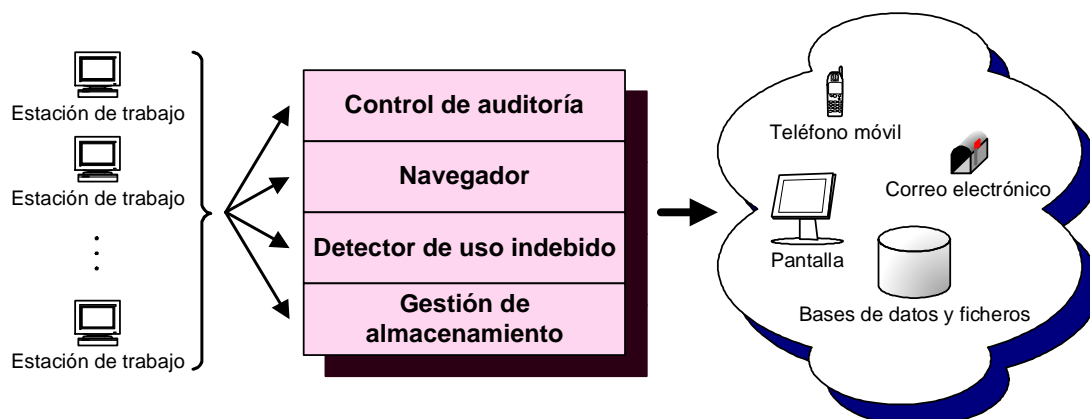


Figura 3-16 - Arquitectura del sistema STALKER

3.2.3.1.5 Monitorización de teclado

Esta técnica consiste en recoger las pulsaciones de teclado para obtener datos que puedan revelar indicios de alguna intrusión.

La principal ventaja de esta solución es que registra toda la actividad tecleada en una sesión interactiva, cosa que está fuera de las posibilidades de los registros de sistema o de auditoría.

Desgraciadamente, existen ataques que no hacen uso del teclado. Además, la misma intrusión se puede expresar de múltiples formas a nivel de pulsaciones de teclas. Si no se hace un análisis semántico de la información obtenida es casi imposible detectar un ataque mediante este método.

3.2.3.1.6 Recuperación de información en modo "batch"

Antes de pasar al siguiente apartado, hay que explicar que los modelos vistos anteriormente trabajan en tiempo real. No obstante, hay soluciones que buscan información adicional en los ficheros de sistema, una vez que el ataque se ha producido. Esto es especialmente útil para expertos de seguridad especializados en el área forense.

Un modelo basado en técnicas IR ("Information Retrieval"), utilizado en algunos motores de búsqueda en Internet como Altavista, es el propuesto por Ross Anderson y Abida Khattak de la Universidad de Cambridge.

Su solución utiliza el registro de sistema UNIX `lastcomm` en conjunción con GLIMPSE, un motor de búsqueda de la Universidad de Arizona. Un guión basado en Perl divide el registro en ficheros únicos para cada usuario. Sobre estos ficheros se ejecutan búsquedas de GLIMPSE relacionadas con los ataques recibidos. Este método sencillo permite encontrar rápidamente información extra sobre las incidencias detectadas. Otra de las ventajas de este enfoque, es que no tiene coste alguno. GLIMPSE es gratuito, y se incluye por defecto en algunos sistemas. [36]

3.2.3.2 Detección de anomalías

En la detección de anomalías se utilizan diversos algoritmos para crear perfiles de comportamiento normal, que sirvan de modelos a contrastar con la conducta actual de cada

usuario. Las desviaciones que resultan de esta comparación son sometidas a técnicas que utiliza el sistema para decidir si ha habido o no indicios de intrusiones. Los *perfiles* están compuestos por conjuntos de métricas. Las *métricas* son medidas sobre aspectos concretos del comportamiento del usuario.

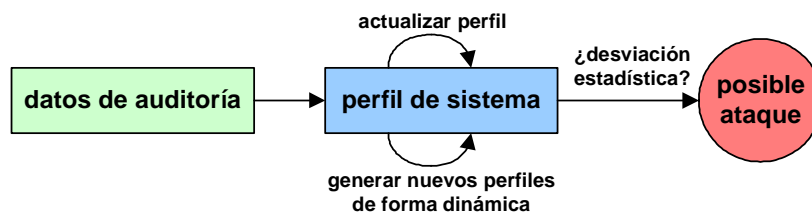


Figura 3-17 - Modelo general de un detector de anomalías

El principal inconveniente de la detección de anomalías es que los perfiles de conducta pueden ser gradualmente *educados*. Un atacante que sepa que sus actividades están siendo monitorizadas, puede cambiar paulatinamente su forma de comportamiento a lo largo del tiempo, para que cuando cometa una intrusión no sea reconocida como tal. Esta técnica es conocida como "session creep" (deslizamiento, o movimiento sigiloso de sesión).

Por otra parte, una de las ventajas la detección de anomalías consiste en su posibilidad de descubrir nuevos ataques, ya que se adapta y aprende de la conducta del sistema, al contrario que la detección de usos indebidos.

3.2.3.2.1 Modelo de Denning

Dorothy Denning indica en su documento "An Intrusion Detection Model" al menos cuatro modelos estadísticos que deben estar presentes en el sistema. Cada uno diseñado para adoptar un determinado tipo de métrica [37]. Estos modelos son:

Modelo operacional

Se aplica sobre eventos para determinar por ejemplo, el número de intentos de acceso fallidos al sistema. Compara su métrica con un valor umbral, que normalmente le indica si se ha cometido una intrusión. Este modelo también es aplicable a la detección de usos indebidos, y es el desarrollado en la detección de umbral, explicada más adelante.

Modelo de desviación media y estándar

Este modelo aplica el concepto de desviación media y estándar típico a la hora de elaborar los perfiles de comportamiento. Supone que a partir de las dos primeras medidas, se puede establecer un intervalo de confianza. Este intervalo sirve para determinar la desviación estándar. Si las siguientes medidas caen fuera de este intervalo revelarían un comportamiento anormal.

Modelo multivariable

Es una ampliación del modelo de desviación media y estándar. Utiliza correlaciones entre dos o más métricas para definir un comportamiento. Por ejemplo, en vez de determinar la conducta de un usuario exclusivamente por la duración de su sesión, también se tiene en cuenta la actividad de tráfico de red que genera durante la misma.

Modelo del Proceso Markov

Este es el modelo más complejo de los cuatro. Considera cada evento de auditoría como una variable de estados, y utiliza una matriz de transiciones de estados para describir la frecuencia de transiciones de estados (no la de cada evento, sino la de todos juntos). Un evento determinado indica una anomalía si su probabilidad es demasiado baja. Este modelo ayudó a desarrollar análisis basados en *flujos de eventos* o la generación patrones probables ("predictive pattern generation") comentada más adelante en esta sección.

3.2.3.2 Análisis cuantitativos

Probablemente, las técnicas basadas en análisis cuantitativos son las más utilizadas para la detección de anomalías. Mediante este método, las reglas de detección y los atributos de los objetos se expresan en forma numérica. Estos elementos se utilizan en análisis de diverso grado de complejidad. Los resultados se pueden usar para aportar patrones a la detección de usos indebidos, o para elaborar perfiles relativos a la detección de anomalías. A continuación se describen algunos ejemplos de este tipo de análisis.

Detección de umbral

Este sistema es conocido también como detección de umbral y disparador ("threshold and trigger"). En este caso, se cuenta el número de veces que ocurren los elementos que forman los perfiles de cada usuario, y se comparan estos datos con valores de umbral. Por ejemplo, si un usuario ha intentado entrar en el sistema más de cinco veces seguidas con una contraseña incorrecta, hay probabilidad de intrusiones. O si se realiza una serie de intentos de apertura de puertos en un intervalo de tiempo inferior a un límite, es posible que se esté intentando escanear el sistema.

Detección heurística de umbral

Esta detección desarrolla el punto de vista anterior, adaptando los valores de umbral a las medidas experimentadas. Por ejemplo, si un usuario suele entrar al sistema por lo general al primer o segundo intento durante un período de tiempo determinado, y en una determinada ocasión se produce un inusual número de intentos fallidos, se produce una anomalía. Los métodos que se utilizan para aproximar el valor de umbral son de diversa naturaleza. Uno de los más simples y conocidos puede ser el de la función de campana de Gauss. Todos aquellos valores que se alejen demasiado de lo definido como normal, revelan una posible intrusión.

Chequeos de integridad basados en objetivo

Consiste en utilizar alguna técnica que permita determinar si el objetivo monitorizado ha cambiado. Normalmente se utilizan funciones de resumen criptográfico ("hash") sobre los objetivos, para generar sumas de control ("checksum") almacenadas en una base de datos protegida. Si el objetivo en cuestión ha cambiado lo más mínimo, su suma de control habrá cambiado sustancialmente con respecto a la anterior. Uno de los productos comerciales más populares que utilizan este método es Tripwire®.

Reducción de auditoría

La reducción de eventos fue uno de los aplicaciones que tuvo el análisis cuantitativo. Este concepto, aparecido en el apartado 3.1.1.3, consiste en la eliminación de información redundante o irrelevante existente en registros de sistema de gran tamaño. Este proceso permite un análisis posterior más rápido y eficiente.

El sistema NADIR, desarrollado en la División Computacional y de Comunicaciones del Laboratorio Nacional de Los Alamos, es un ejemplo de un producto que utiliza este enfoque.

NADIR transforma los registros de auditoría en vectores, y los somete a un proceso de reducción de datos. El resultado es examinado mediante análisis estadísticos y sistemas expertos supervisados por un operador. [38]

3.2.3.2.3 Medidas estadísticas

Las medidas estadísticas constituyen uno de los primeros métodos utilizados para la detección de anomalías. Muchos de los productos de detección de intrusiones basados en red utilizan esta técnica. Suelen servirse de *filtros de anomalías de protocolo* [39], que en realidad son filtros de anomalías estadísticas, adaptados para trabajar con protocolos de red.

El término *filtro de anomalías estadísticas*, se utiliza para definir sistemas que buscan eventos cuyo valor estadístico es inusualmente bajo, haciéndolos sospechosos. Un filtro de anomalías de protocolo, es un tipo de filtro de anomalías estadísticas al que se ha añadido información específica sobre protocolos de red. En el diagrama siguiente se puede observar la relación entre las áreas identificadas como definiciones oficiales de protocolos, el uso práctico de estos protocolos, y los ataques o intrusiones. Gran parte de los ataques que se producen, son debidos a usos anómalos de los protocolos. Casi todos los ataques explotan las debilidades existentes, tanto en las definiciones oficiales como en algunas implementaciones de los protocolos. Los filtros de anomalías de protocolo son capaces de detectar aquellos ataques realizados fuera del área de uso normal.

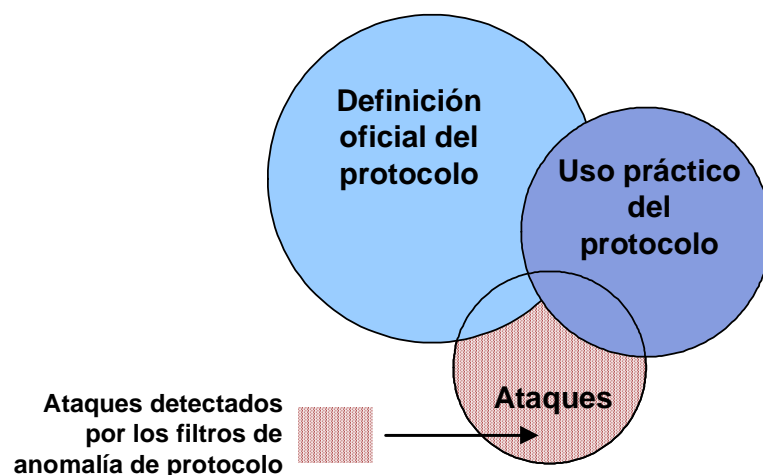


Figura 3-18 - Ataques que detecta un filtro de anomalías de protocolo

Algunos de los sistemas que utilizan medidas estadísticas están *basados en frecuencia*, en los que la frecuencia con la que se da un determinado evento determina su probabilidad. Algunos de los sistemas partidarios de esta idea son el sistema IDES (utilizado en el proyecto NIDES), el sistema Haystack, SPADE, o ADAM, descritos más adelante.

Más tarde comenzaron a hacer aparición acercamientos *basados en tiempo*, en los que lo que determina la probabilidad de un evento es el tiempo transcurrido desde que sucedió por última vez. Sistemas como PHAD, ALAD, LERAD o NETAD son algunos de los partidarios de este enfoque, explicados más abajo en esta sección.

Haystack

Haystack fue creado por Tracor Applied Sciences y los Laboratorios Haystack para las Fuerzas Aéreas norteamericanas [40]. Incorpora dos mecanismos estadísticos para la detección de anomalías. El primero investiga el grado de similitud entre una sesión de usuario y una intrusión conocida. Esto lo hace aplicando diversos algoritmos a vectores del comportamiento de usuario, y relacionado los resultados con puntuaciones obtenidas de intrusiones ya establecidas. El segundo, de forma complementaria, calcula la desviación entre las estadísticas obtenidas de la sesión actual del usuario y los datos de su perfil estadístico histórico. [41]

IDES/NIDES

Ambos sistemas fueron desarrollados por SRI International. Utilizan tanto detección de usos indebidos como de anomalías. En lo que respecta a la detección de anomalías, se hace uso de perfiles estadísticos históricos, para cada usuario y sistema. Estos perfiles se actualizan con el paso del tiempo según los cambios de comportamiento.

El sistema IDES utiliza una base de conocimiento estadística con los perfiles obtenidos. Cada perfil contiene su conjunto de métricas. Esta estadística, denominada puntuación IDES (IS), se calcula utilizando la fórmula:

$$IS = (S_1, S_2, \dots, S_n) C^{-1} (S_1, S_2, \dots, S_n)^t$$

En la que C^{-1} es la inversa de la matriz de correlaciones del vector (S_1, S_2, \dots, S_n) , y $(S_1, S_2, \dots, S_n)^t$ es la transpuesta de ese vector. Cada medida S_n representa elementos de comportamiento, tales como tiempo de sesión, número de intentos de acceso fallido, consumo de CPU, uso de la red. [42]

SPADE / SPICE

Ambos productos han sido desarrollados por Silicon Defense. SPADE (Statistical Packet Anomaly Detection Engine) es un preprocesador de datos que permite la detección de paquetes sospechosos utilizando técnicas de detección de anomalías. Se puede ejecutar como "plugin" del Snort.

El sistema SPICE (Stealthy Probing and Intrusion Correlation Engine), está formado por dos elementos: el ya mencionado detector de anomalías SPADE, y un correlacionador de eventos.

El funcionamiento consiste en la detección de anomalías por parte de SPADE, las cuales envía al correlacionador. Una vez relacionadas y agrupadas, el correlacionador las registra, comunicando las posibles actividades sospechosas. El correlacionador es capaz de detectar por ejemplo, escaneos de puertos en orden aleatorio, demasiado lentos, o incluso aquellos realizados desde distintas fuentes. [43]

ADAM

Este sistema, al igual que NIDES o SPADE, es un detector de anomalías basado en frecuencia, es decir, estima la probabilidad de cada evento según con la frecuencia que ocurra. Como el resto de los sistemas de su categoría, no necesita bases de datos de firmas para reconocer actividades inusuales.

Puede ser entrenado para identificar tanto ataques conocidos como los aún no publicados. Identifica las posibles intrusiones marcando cualquier actividad que no ha aprendido como anómala. [44]

PHAD [45], ALAD [46], LERAD [47]

Estos métodos de detección de anomalías de red están basados en tiempo, de modo que determinan la probabilidad de un evento según el tiempo que ha pasado desde que tuvo lugar por última vez. Fueron diseñados por Matthew V. Mahoney y Philip K. Chan en el Institute of Technology de Florida.

Utilizan el concepto de *atributo* para analizar el tráfico de red. Los atributos son normalmente datos obtenidos de los paquetes de red, como los campos de cabecera.

Para cada atributo, recopilan un conjunto de valores permitidos (cualquier cosa observada al menos una vez durante el entrenamiento o aprendizaje), y marcan aquellos valores nuevos como anómalos. Utilizan la siguiente ecuación para puntuar los atributos nuevos:

$$\sum tn/r$$

Donde t es el tiempo transcurrido desde que el atributo fue identificado como anómalo por última vez, n el número de entrenamientos realizados, y r el tamaño del conjunto de valores permitidos. De esto se puede deducir que r/n es el valor medio de anomalías detectadas durante entrenamientos. Además, el factor t hace que el modelo sea dependiente del tiempo, otorgando mayores valores a aquellos atributos que llevan más tiempo sin ocurrir.

Las diferencias entre los sistemas PHAD, ALAD y LERAD radican en los atributos que monitorizan. PHAD ("Packet Header Anomaly Detector") contempla 34 atributos correspondientes a las cabeceras de paquetes Ethernet, IP, TCP, UDP e ICMP. ALAD ("Application Layer Anomaly Detector") construye modelos a partir de las peticiones TCP entrantes, formados por las direcciones origen y destino, los puertos utilizados, los indicadores TCP, y los comandos existentes en el campo de datos de aplicación. Puede llegar a construir modelos independientes para cada "host", puerto o combinación de ambos. LERAD ("LEarning Rules for Anomaly Detection") también construye modelos TCP, pero utiliza los datos obtenidos para sugerir modelos. Por ejemplo, si registra dos peticiones FTP, destinadas al mismo "host", puede sugerir una regla que indica que todas las peticiones hacia ese "host" deben ser FTP y construye un modelo de puerto para ese "host".

NETAD

El sistema NETAD ("Network Traffic Anomaly Detector") fue escrito por Matthew V. Mahoney en el Institute of Technology de Florida. Es muy similar a PHAD, pero fue diseñado para mejorar algunas de sus características. [48]

Utiliza la detección de anomalías en paquetes de red, y está basado en tiempo. No obstante, cuenta con algunas características que lo diferencian de PHAD:

- Filtra el tráfico, examinando sólo el comienzo de las peticiones entrantes.
- Utiliza como atributos los primeros 48 bytes de los paquetes que comienzan con cabeceras IP.
- Cuenta con 9 modelos distintos correspondientes a los protocolos más utilizados, como IP, TCP o HTTP.
- La ecuación tn/r para determinar las anomalías ha sido modificada para indicar sucesos poco comunes, pero no necesariamente anómalos.

La mayoría de los ataques comienzan mediante el envío de peticiones a un servidor o sistema operativo, por lo que para su detección suele ser suficiente analizar el comienzo de las peticiones enviadas a la víctima. NETAD utiliza diferentes métodos para esta tarea: descarta paquetes que no sean IP, tráfico saliente (no detecta las respuestas anómalas de un servidor), las conexiones TCP entrantes que comienzan con indicadores SYN-ACK activados (indicando que la conexión ha sido iniciada por un "host" local), etc.

Los atributos de NETAD están compuestos por los primeros 48 bytes de los paquetes que comienzan con una cabecera IP. Por ejemplo, en un paquete TCP normal, los atributos son los 20 bytes de cabecera IP, otros 20 bytes de cabecera TCP, y los 8 primeros bytes del campo de datos de aplicación.

Los 9 modelos construidos por NETAD son 9 subconjuntos de tipos de paquetes comunes, obtenidos del tráfico de red. Algunos de estos modelos son: todos los paquetes IP (incluyendo TCP, UDP e ICMP), todos los paquetes TCP, todos los paquetes TCP SYN (sin otro indicador, es normalmente el primer paquete, con opciones TCP y sin campo de datos), todos los paquetes TCP ACK (normalmente el segundo y siguientes paquetes, que contienen campo de datos), paquetes TCP ACK a puertos 0-255, TCP ACK al puerto 21 (FTP), etc. Es posible que un paquete pueda pertenecer a más de un subconjunto. Por ejemplo, un paquete de datos FTP, también es TCP a puertos 0-255, TCP ACK, TCP e IP.

NETAD utiliza la siguiente versión modificada de la puntuación de anomalías tn/r (adoptada por PHAD, ALAD y LERAD):

$$\sum tn_a(1 - r/256) + t_i(f_i + r/256)$$

Donde n_a es el número de paquetes de entrenamiento transcurridos desde la última anomalía hasta el final del período de entrenamiento, t el período de prueba, r el número de valores permitidos (entre 1 y 256), t_i el tiempo transcurrido desde que fue contemplado el valor i (entre 0 y 255) tanto durante entrenamientos o pruebas, y f_i la frecuencia con la que ha ocurrido el valor i durante el entrenamiento. Una de las ideas destacables de este modelo de puntuación, es que tiene en cuenta el concepto de frecuencia, cosa que no hacen PHAD, ALAD o LERAD.

Ventajas e inconvenientes de las medidas estadísticas

Las medidas estadísticas para la detección de intrusiones, están entre las técnicas más utilizadas. Sin embargo, no por ello dejan de tener sus inconvenientes. Aquí se enumeran las características más relevantes, tanto positivas como negativas.

La ventaja más conocida, y quizás la más importante de los métodos basados en análisis estadísticos, es su capacidad para la detección de nuevos ataques. Los filtros de anomalías de protocolo, mencionados antes, pueden detectar nuevos ataques, aún sin haber sido registrados por las entidades oficiales de seguridad. No necesitan actualizar ninguna base de datos de firmas. En ese aspecto, son superiores a los sistemas de detección basados en patrones.

Otra de las características, derivada de la anterior, de los sistemas estadísticos es que no requieren el mantenimiento que necesitan los sistemas de detección de usos indebidos. No utilizan bases de firmas o patrones. Esto implica, por supuesto, contar con un modelo que utilice las métricas precisas, y que se adapte adecuadamente a los cambios de comportamiento de los usuarios.

Uno de los inconvenientes que más se achacan a la detección de anomalías mediante medidas estadísticas, es que puede ser paulatinamente entrenada, cosa que no ocurre con la detección de usos indebidos. Un usuario malicioso que supiera que está siendo monitorizado, podría cambiar intencionadamente su actitud para que, en un momento dado, el sistema identificara como normal un comportamiento hostil.

Otro de los inconvenientes de estos sistemas radica en el gran consumo de recursos que utilizan frente a otros modelos propuestos. Los análisis estadísticos normalmente requieren más tiempo de proceso que los sistemas de detección de usos indebidos. Esta limitación fue una de las razones por las que los primeros sistemas estadísticos se ejecutaban en modo "batch" antes de poderse realizar en tiempo real.

Por otra parte, la naturaleza de los análisis estadísticos les impide tener en cuenta relaciones entre eventos secuenciales. El orden de ocurrencia de eventos no suele ser uno de los atributos utilizados por los modelos de análisis estadísticos. No pueden reconocer directamente ataques realizados mediante sucesiones de eventos en un determinado orden. Esto representa una seria limitación, dado el elevado número de intrusiones basadas en estas características.

En muchos sistemas basados en medidas estadísticas, el número de falsas alarmas es demasiado alto, lo que hace que muchos usuarios las ignoren.

3.2.3.2.4 Medidas estadísticas no paramétricas

Los sistemas estadísticos pueden ser de tipo paramétrico o no paramétrico. Los primeros son aquellos cuyas distribuciones son conocidas de antemano. Por ejemplo, en las primeras versiones de IDES, la distribución utilizada era la Gaussiana o normal. La mayoría de los primeras soluciones basadas en medidas estadísticas utilizaban aproximaciones de tipo paramétrico.

Los enfoques estadísticos no paramétricos, por tanto, trabajan con perfiles de comportamiento que no se basan en distribuciones preestablecidas.

El inconveniente de los métodos paramétricos es que las tasas de error en algunos casos es demasiado alta. Si una determinada métrica, como el consumo de CPU, no se ajustaba a una distribución conocida como, por ejemplo, la normal, se producían demasiados errores.

En la Universidad de Tulane, Linda Lankewicz y Mark Benard propusieron un modelo de detección de anomalías basado en técnicas no paramétricas [49]. Esta solución utiliza patrones de uso menos predecibles y permite tener en cuenta medidas difíciles de utilizar en enfoques paramétricos.

Los datos se clasifican mediante una técnica denominada "clusterig analysis" (agrupación de datos). La información histórica se agrupa y organiza en "clusters" (grupos) según diversos criterios de evaluación (llamados "features"). Los datos asociados a un determinado usuario u objeto son preprocesados y convertidos en representaciones vectoriales (por ejemplo $X_i = [f_1, f_2, \dots, f_n]$). Luego, mediante un algoritmo de agrupación, esos vectores se reúnen en clases de comportamiento. La idea es que aquellos miembros de una clase estén muy relacionados unos con otros, mientras que los de clases diferentes sean lo más distintos posible.

Siguiendo este modelo no paramétrico, la actividad de un usuario, expresada en términos de criterios de evaluación, se divide en dos grupos principales: uno que indica actividad anómala, y otro que indica actividad normal.

Una de las ventajas de modelos no paramétricos es que hacen una efectiva reducción de datos de auditoría, mediante la transformación de eventos en vectores. Por otra parte, tienen mayor velocidad de detección que los análisis estadísticos paramétricos.

No obstante, un consumo de recursos excesivo, por parte de los criterios de evaluación, reduciría de forma notable la eficiencia de sistemas basados en este tipo de análisis.

3.2.3.2.5 Sistemas basados en reglas

Otra de las aproximaciones de la detección de anomalías es la basada en reglas ("rule-based"). Utiliza los mismos métodos que la detección de anomalías estadística. La diferencia radica en que la detección de intrusiones basada en reglas hace uso de conjuntos de reglas para representar y almacenar patrones de uso. Algunos de los sistemas que utilizan esta metodología son Wisdom and Sense, o TIM (Time-Based Inductive Machine).

Wisdom and Sense

"Wisdom and Sense" (Sabiduría y sentido), fue desarrollado en el Laboratorio Nacional de Los Alamos y el Laboratorio Nacional de Oak Ridge [50]. Soporta gran variedad de plataformas y puede trabajar a nivel de sistema operativo y de aplicación.

Tiene dos formas de añadir reglas a la base de datos: de forma manual, o mediante generación automática a partir de los datos históricos de auditoría. Las reglas reflejan el comportamiento de los usuarios y objetos y están almacenadas en una estructura de árboles denominada *bosque*. Algunos datos específicos dentro de los registros de auditoría se agrupan en "thread classes" (clases de hilo) a las que se asocian conjuntos de operaciones o reglas. Una "thread class" podría ser "los registros asociados a un determinado usuario".

Las anomalías se detectan al comparar los nuevos eventos con aquellos asociados a los "threads" correspondientes, y evaluando su relación con los patrones históricos de actividad.

Generación de patrones probables

La generación de patrones probables o "Predictive Pattern Generation" intenta predecir eventos futuros basándose en aquellos que ya han tenido lugar [51]. Utiliza una base de reglas de perfiles de usuarios definida a partir de estadísticas de eventos. La Figura 3-19 muestra un ejemplo de una posible regla creada mediante esta metodología:

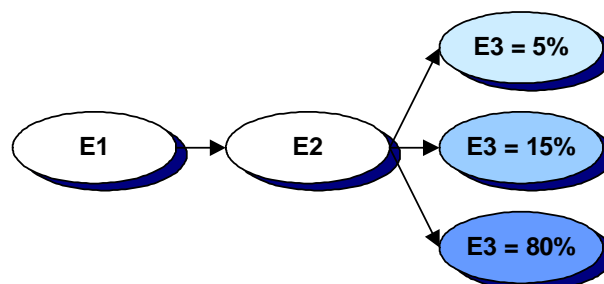


Figura 3-19 - Generación de patrones probables

Esto significa que, contando con la premisa de que los eventos E1 y E2 han ocurrido en ese orden, hay un 80% de probabilidades de que siga E3, un 15% de que ocurra E4, y un 5% de que tenga lugar E5. No obstante, algunas de las intrusiones que no figuren en las reglas no se reconocerán como tales. Por lo tanto, si una secuencia de eventos determinada existe, y pertenece a una intrusión, pero no está en la base de reglas, se marcará como no reconocida. Esto se puede arreglar marcando los eventos no reconocidos como intrusiones, aunque esta medida también aumentará el número de falsos positivos. Normalmente, un evento se marca como intrusión si la parte izquierda de la regla coincide, pero la parte derecha es muy diferente de la predicción estadística.

Este método tiene numerosas ventajas. Una de ellas, es que puede detectar anomalías en secuencias de eventos, difíciles de detectar mediante otros métodos menos flexibles. Por otra parte, adapta con facilidad a los cambios de comportamiento. Los patrones menos utilizados son paulatinamente eliminados, quedando a largo plazo aquellos de más calidad. Mediante esta técnica es más fácil identificar a aquellos usuarios que intentan entrenar al sistema durante su fase de aprendizaje (método ya comentado, denominado "session creep"). Esto es así, debido a que la semántica se construye en las propias reglas de detección. Además, este método tiene pocos requerimientos de recursos de sistema, haciendo posible el proceso de los datos de auditoría e identificación de anomalías en poco tiempo.

Uno de los inconvenientes de estos sistemas es que, como ocurre con todos los sistemas de aprendizaje, dependen de la calidad del material de entrenamiento que usan. Los datos de aprendizaje deben ser lo más fieles posibles a la actividad normal de los usuarios. Además, el número de falsos positivos es grande principalmente al comienzo de la operación, debido a que la mayoría de los eventos existentes no suelen coincidir con los comienzos de las reglas.

TIM

El sistema TIM ("Time-Based Inductive Machine") fue desarrollado por Teng, Chen y Lu en asociación con Digital Equipment Corporation [52]. Utiliza métodos inductivos para generar secuencias de patrones. Implementa un tipo de modelo de probabilidad de transiciones de Markov de Denning, buscando patrones en secuencias de eventos.

Su funcionamiento y características están basados en la técnica de generación de patrones probables, explicada antes. Como TIM utiliza grupos de secuencias de eventos, el espacio que necesita para su base de reglas es menor que el que necesitan aquellos sistemas que trabajan con eventos aislados, como Wisdom and Sense.

Este producto sentó las bases del producto de detección de intrusiones Digital Equipment Corporation Polycenter y sirvió de referencia para muchos trabajos sobre detección de anomalías posteriores.

3.2.3.2.6 Redes neuronales

Una de las formas de detección de anomalías más prometedoras es la basada en redes neuronales. El concepto consiste en aprovechar las características de aprendizaje de estas redes, para predecir las acciones de los usuarios, dado un número determinado n de acciones previas conocidas. [53]

Las redes neuronales están formadas por numerosos elementos de procesamiento simple denominados *unidades* que se interactúan entre sí mediante *conexiones* con peso. El conocimiento de una red neuronal se almacena mediante la indicación de las conexiones entre las unidades y sus

pesos. El proceso de aprendizaje se realiza cambiando pesos y aumentando o disminuyendo el número de conexiones.

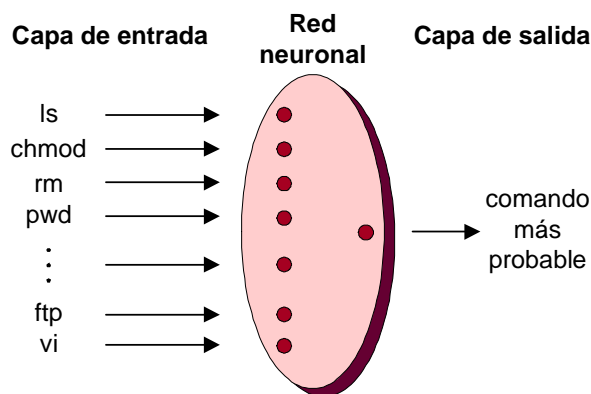


Figura 3-20 - Redes neuronales para la detección de anomalías

La red neuronal es entrenada mediante conjuntos de los comandos más significativos. Después de un período de entrenamiento, la red contrasta los comandos obtenidos con el perfil de usuario. Cualquier comando o acción predicha incorrectamente permiten determinar el grado de desviación entre el usuario y el perfil establecido.

En el Departamento de Tecnología Informática y Computación de la Universidad de Alicante se ha hecho un estudio sobre la utilización de redes neuronales en el que se han obtenido extraordinarios resultados [54]. Aplicaron dos métodos en la experimentación: una red de perceptrón multicapa y un mapa auto-organizativo (MAO) [55].

En el estudio se utilizaron como datos de entrada 29 elementos consistentes en elementos de cabeceras de paquetes IP, TCP, UDP, TCMP y datos obtenidos a partir de los contenidos de los paquetes. Luego, en una red local de dos ordenadores, y con la ayuda de la información proporcionada por el escáner de vulnerabilidades Nessus y el IDS Snort, se capturaron un total de 443 paquetes "peligrosos" utilizados en las pruebas. Utilizaron una parte de estos paquetes para entrenar a los sistemas y la otra para probarlos, con unos porcentajes de aciertos de más del 90%.

Las redes neuronales tienen la ventaja de que funcionan bien con datos con ruido. No hacen suposiciones estadísticas sobre la naturaleza de los datos. Son capaces de detectar nuevas formas de ataque no conocidas, sin necesidad de reglas introducidas manualmente. Además, son fáciles de modificar para soportar nuevos conjuntos de usuarios.

Uno de los inconvenientes de estos sistemas está relacionado con la cantidad de datos a utilizar durante el entrenamiento. Las redes neuronales provocarían muchos falsos positivos si los datos son insuficientes, y si son demasiados, el número de datos irrelevantes sería alto, aumentando los falsos negativos. Otro de los puntos en contra de este modelo es que no aporta ninguna explicación sobre las anomalías que identifica, dificultando la posibilidad de corregir las raíces del problema de seguridad en el sistema. Por otra parte, un intruso podría ser capaz de entrenar a la red durante la fase de aprendizaje ("session creep"). Algunos desarrolladores han aportado soluciones híbridas que intentan solventar algunos de estos inconvenientes. [56]

3.2.3.3 Métodos alternativos

Aparte de los métodos de detección mencionados, han ido apareciendo nuevas soluciones, aplicables bien a la detección de usos indebidos, bien a la detección de anomalías.

Con toda seguridad, en un futuro cercano aparecerán más técnicas como las descritas a continuación. Como se comprobará, las posibilidades en el campo de la detección son enormes. Cualquier sistema relacionado con técnicas de aprendizaje, reducción de datos, o toma de decisiones se puede aplicar de algún modo a la detección de intrusiones, bien en la detección de usos indebidos o en la detección de anomalías.

Si bien es cierto que estas técnicas son utilizadas en muchas ocasiones en conjunción con otras más tradicionales, para refinar los procesos de detección, también se encuentran como propuestas independientes.

3.2.3.3.1 Sistema inmune

Esta propuesta consiste en aprovechar las similitudes existentes entre el sistema inmune del organismo y la detección de intrusiones, basadas en la identificación de lo que es propio al sistema y lo ajeno al mismo.

El sistema inmune es capaz de reconocer comportamientos extraños al organismo (antígenos). Los antígenos, en el contexto de un sistema computacional con usuarios y comportamientos individuales, pueden estar relacionados:

- Con comportamientos anómalos del usuario.
- Con ataques conocidos de antemano.

Estas características se pueden utilizar para la detección de anomalías y la de usos indebidos.

Algunos desarrolladores de la Universidad de México (Forrest, Hofmeyr, y Somayagi, entre otros) han aplicado este método a las secuencias de llamadas al sistema, bajo UNIX [57]. Su sistema utiliza pequeñas secuencias de llamadas al sistema, teniendo en cuenta sólo la relación temporal entre las mismas.

El proceso de detección de intrusiones se completa en dos fases. La primera consiste en la creación de un perfil de comportamiento normal. La diferencia de este perfil con respecto a otros comentados en este capítulo es que está centrado en procesos de sistema, no en usuarios. En la segunda fase se utiliza dicho perfil para, a través de desviaciones de comportamiento de sistema, identificar anomalías.

Los resultados obtenidos fueron sorprendentes, permitiendo la detección de anomalías en el comportamiento de varios programas de UNIX históricamente problemáticos. Además, las secuencias de ejecución utilizadas eran bastante reducidas [58].

Por otra parte, en la Universidad de Chile y Concepción se ha propuesto un sistema de detección de intrusiones basado en la Biología, más concretamente en el sistema inmune. [59]

Esta propuesta analiza el comportamiento de usuarios analizando las secuencias de comandos que ejecutan [60][61][62], en un entorno dinámico, en el que el sistema se adapta

gradualmente a los cambios de comportamiento de cada usuario. Utilizan dos criterios asociados a las secuencias estudiadas: *comandos sucesores*, que caracteriza a los usuarios a través de la relación causal de los comandos ejecutados; y *secuencia segmentada*, mas compleja que la anterior, que busca patrones extensos y tal vez segmentados en las cadenas recogidas. Estas secuencias sirven para crear los perfiles de usuario.

Para el tratamiento y respuesta ante intrusiones, el sistema utiliza una arquitectura distribuida basada en agentes, que se comenta más a fondo en un apartado posterior. Los principales componentes de esta estructura son los denominados *AgentesT*, nombre derivado de Linfocitos T.

La recopilación de los comandos de consola se realizó a través de un "wrapper" (envoltura) de consola, programada específicamente para esta propuesta, denominada JudaShell (jsh). Esta herramienta no sólo proporciona los servicios usuales de una consola normal y registra los comandos ejecutados, sino que permite el análisis de los datos para facilitar la labor a los agentes, y restringir las operaciones de los usuarios.

Los resultados de la propuesta fueron más que satisfactorios, haciendo este sistema útil tanto en ambientes monousuario como en entornos multiusuario. No obstante, como ya se apuntó, los sistemas susceptibles de ser entrenados corren el riesgo de ser atacados por usuarios que modifican maliciosamente su actitud a largo plazo ("session creep").

En general, el uso de teorías basadas en el sistema inmune para la detección de intrusiones, permite abordar sistemas complejos de forma más simple que muchas otras alternativas, y sin diferencias significativas en los resultados.

No obstante, estas técnicas no deberían utilizarse de forma única, sin el apoyo de algún otro mecanismo de detección complementario. Algunos ataques, tales como los de condición de carrera, enmascaramiento, o violaciones de políticas de sistema no hacen uso de procesos privilegiados, por lo que no son detectados por este enfoque.

3.2.3.3.2 Genética

Los *algoritmos genéticos* también son de gran utilidad en la detección de intrusiones, como avanzado método de análisis de datos.

Un algoritmo genético es un algoritmo de búsqueda basado en la mecánica de la selección natural y de la genética natural. Combina la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorio, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas [63].

El algoritmo genético es uno de los algoritmos englobados en el conjunto que recibe el nombre de *algoritmos evolutivos*; los cuales, utilizan las nociones de la selección natural formulada por Darwin para solucionar problemas.

Cada solución será representada a través de una cadena de 0s y de 1s ó *cromosomas* que se verán entonces sometidos a una imitación de la evolución de las especies: mutaciones y reproducción por combinación. Como se favorece la supervivencia de los más "aptos" (las soluciones más correctas), se provoca la aparición de híbridos cada vez mejores que sus padres. Al despejar los elementos más aptos, se garantiza que las generaciones sucesivas serán cada vez más adaptadas a la resolución del problema.

Para utilizar algoritmos genéticos en la detección de intrusiones, los desarrolladores han definido *vectores de hipótesis* para los datos de eventos, donde los vectores pueden indicar si ha habido intrusión o no. Entonces, la hipótesis se somete a prueba para determinar si es válida. Con los resultados de la prueba, se desarrolla una versión mejorada (*evolucionada*) de la hipótesis. Este proceso se repite hasta encontrar una solución.

En el sistema GASSATA, desarrollado por Ludovic Mé, en Francia, aplica el algoritmo genético al problema de la clasificación de eventos mediante el uso de vectores de hipótesis H (uno por cada flujo de eventos relevante) de n dimensiones (donde n representa el número de ataques potenciales). [64]

Los resultados experimentales fueron asombrosos. Se obtuvo una probabilidad de verdaderos positivos del 0.996, siendo de 0.0044 la de falsos positivos. El tiempo necesitado para construir los filtros también fue mínimo. Para un conjunto de 200 ataques, al sistema le llevó 10 minutos y 25 segundos analizar los eventos recopilados durante 30 minutos de intensa actividad de usuario.

En los algoritmos genéticos, los mecanismo de evolución y selección son independientes del problema a resolver: sólo varían la función que descodifica el genotipo en una solución posible y la función que evalúa el ajuste de la solución. Esta técnica es de aplicación general.

Por último, hay que mencionar que el algoritmo genético puede ayudar en la producción de una variedad de objetos, mientras sea posible obtener una calificación que permita expresar la solución. De esta forma, es posible fabricar previsores estadísticos, no a través de cálculos de datos como en la estadística clásica, sino haciendo evolucionar los datos mediante algoritmo genético ("inducción"). El mecanismo de estimulación de los más aptos, permite la aparición del previsor, que reordenará los datos lo mejor posible. Este tipo de construcción de previsor forma parte de las llamadas técnicas de "data mining" (minería de datos), comentada a continuación.

Los previsores producidos pueden tener formas muy diversas, como: bases de reglas, evaluación por puntuación, árboles de decisión e incluso redes neuronales.

3.2.3.3 "Data mining" (minería de datos)

Según apuntan algunos, la minería de datos es el sucesor de la estadística clásica. Tanto uno como otra, llevan al mismo fin, construir *modelos* compactos y comprensibles que relacionen la descripción de una simulación y un resultado relacionado con dicha descripción. A grandes rasgos, la mayor diferencia entre ambos, reside en que las técnicas de minería de datos construyen su modelo de forma automática, mientras que las técnicas estadísticas clásicas deben ser manejadas por un profesional.

La detección de intrusiones que utiliza técnicas de minería de datos es similar a la basada en reglas. Intenta descubrir patrones fiables de características de sistema que puedan definir pautas de comportamiento de sistema y usuario. Estos conjuntos de características de sistema son procesados mediante métodos de inducción por motores de detección que identifican tanto anomalías como usos indebidos.

La minería de datos extrae modelos a partir de grandes cantidades de información. Tiene la peculiaridad de encontrar relaciones ente los datos que serían más difíciles de detectar mediante otros métodos de análisis. Entre los algoritmos disponibles para aplicar la minería de datos sobre datos de auditoría predominan tres: *clasificación*, *análisis de enlace*, y *análisis de secuencia*.

- **Clasificación:** Asigna los datos a una serie de categorías predefinidas. Los algoritmos de clasificación devuelve *clasificadores*, tales como árboles de decisión o reglas. En la detección de intrusiones, los clasificadores deciden si los datos de auditoría pertenecen a una categoría normal o anómala.
- **Análisis de enlace:** Identifica las relaciones y correlaciones entre los campos en el cuerpo de los datos. Un algoritmo de análisis de enlace óptimo reconoce el conjunto de características de sistema más adecuado para revelar intrusiones.
- **Análisis de secuencia:** Crea patrones de secuencias. Estos algoritmos pueden identificar aquellos eventos que suelen ocurrir juntos, y proporcionar medidas estadísticas de tiempo para mejorar la detección de intrusiones. Estas medidas ayudan en la detección de ataques basados en denegación de servicio.

El número de propuestas sobre de detección de intrusiones basadas en la minería de datos es abundante. Muchas de ellas han sido desarrolladas en la Universidad de Columbia, por Wenke Lee y Salvatore J. Stolfo. Algunos de sus trabajos son "A Data Mining Framework for Building Intrusion Detection Models" [65], "Adaptative Intrusion Detection: a Data Mining Approach"[66], "Mining Audit Data to Build Intrusion Detection Models" [67], o "A Data Mining and CIDF Based Approach for Detecting Novel and Distributed Intrusions" [68].

Una de las ventajas de la minería de datos es que mejoran el rendimiento, la manejabilidad y reducen el tiempo de trabajo. Su capacidad de realizar modelos propios, a partir de algunos tipos de datos (como el consumo de CPU), difíciles de encajar en distribuciones conocidas, les hace idóneos para la detección de intrusiones. Además de su rapidez, también son valorados por su sencillez. Estas técnicas permiten trabajar con importantes cantidades de información sin problemas.

3.2.3.3.4 Detección basada en agentes

Los agentes son aplicaciones de software que realizan funciones de monitorización en máquinas. Funcionan de forma autónoma, es decir, son sólo controlados por el sistema operativo, no por otros procesos. Los agentes están siempre operativos, siendo posible la comunicación y cooperación entre ellos si es necesario.

El grado de complejidad de los agentes es variable. Pueden realizar tareas sencillas como registrar el número de ocasiones en que un usuario entra al sistema, o más complejas como la búsqueda de evidencias de ciertos ataques, de acuerdo con determinados parámetros. Además, tienen la capacidad de responder de forma muy precisa ante posibles intrusiones, por ejemplo, modificando prioridades de procesos.

Dadas sus características, los agentes se pueden utilizar tanto en detección de anomalías como de usos indebidos.

Agentes autónomos

Muchos sistemas de detección de intrusiones utilizan agentes. El sistema denominado Agentes Autónomos para la Detección de Intrusiones (AAFID), propuesto en la Universidad de Purdue, fue uno de los primeros en utilizar estos elementos.

Su arquitectura está organizada de forma jerárquica, en la que existen mecanismos de control y divulgación, tal como se observa en la Figura 3-21 siguiente. Pueden existir varios agentes

por máquina ("host"), y cada uno de ellos envía sus datos a un transmisor-receptor. Estos últimos, coordinan las operaciones realizadas por los agentes y se encargan de su funcionamiento y configuración. Además, realizan funciones de reducción de datos y envían sus resultados al siguiente nivel jerárquico, formado por uno o más monitores.

Los monitores, que pueden estar estructurados en distintos niveles, procesan la información que reciben de los transmisores-receptores. Los monitores pueden obtener datos de toda la red, de forma que pueden detectar ataques multi-máquina ("multi-host"), realizados desde diferentes sistemas. Los monitores son controlados a través de una interfaz de usuario, y utilizan esa información para enviar órdenes a los transmisores-receptores.

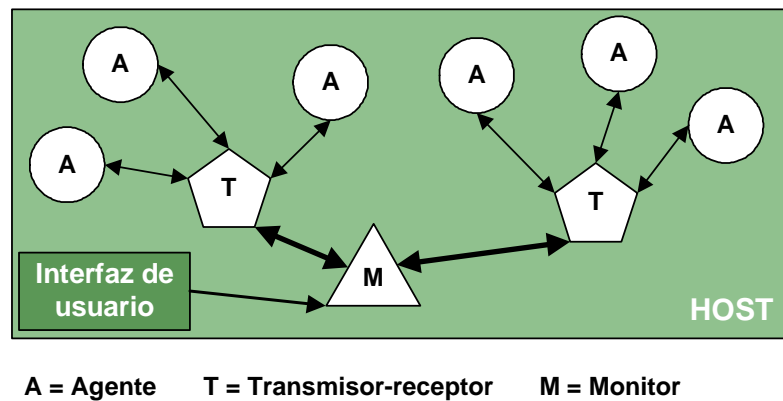


Figura 3-21 - Arquitectura del sistema AAFID

Entre las ventajas de AAFID cabe destacar que es un sistema más robusto que otros mecanismos de detección de intrusiones. Su arquitectura estructurada en niveles, permite la adición o extracción de componentes de forma sencilla. Los agentes pueden ser probados de forma independiente antes de implantarlos en el sistema. Como estos agentes se pueden comunicar entre sí, pueden realizar de forma individual tareas simples, de forma que la tarea conjunta sea más compleja.

No obstante, este sistema también tiene sus desventajas. Si un monitor falla, los datos enviados por los transmisores-receptores conectados a él, dejarán de llegar. Una de las alternativas propuestas más fáciles de imaginar para solventar este problema, es la duplicación de los monitores, análogamente a como se hace con servidores de "backup" en redes. Desgraciadamente, esta situación provoca problemas de consistencia de datos y duplicación de información que requieren técnicas adicionales. Hay problemas, comunes a todos los sistemas de detección de intrusiones distribuidos, asociados al retardo entre la ocurrencia de una incidencia y su llegada al monitor. Otro inconveniente no menos importante, y existente en muchos otros sistemas de detección de intrusiones, es el de contar con una interfaz de usuario demasiado pobre en contenidos. La forma de representar las reglas de sistema, así como las políticas de seguridad es mejorable.

3.2.3.3.5 Lógica difusa

La lógica difusa ("fuzzy logic") es una forma de razonamiento que incorpora criterios múltiples para tomar decisiones y valores múltiples para evaluar posibilidades.

En lógica dicotómica (método de razonamiento, basado en que cada restricción del problema puede ser considerada verdadera o falsa), se espera derivar una solución decidiendo entre sí o no, dependiendo de si cada una de las restricciones o parámetros es verdadero o falso. En cambio, en lógica difusa es posible utilizar escalas de condiciones (restricciones) y matices (flexibilidad) en los valores numéricos. En el intervalo $[0..1]$ puede caber cualquier valor de verdad, sin necesitar ser un número entero. Permite volcar numéricamente expresiones del tipo *muy caliente*.

Este método se suele utilizar para mejorar y afinar el funcionamiento de otros sistemas. Por ejemplo, en la Universidad de Mississippi, los profesores Susan M. Bridges y Rayford B. Vaughn propusieron un prototipo de sistema de detección de intrusiones inteligente (IIDS) basado en técnicas de "fuzzy data mining" y algoritmos genéticos [69]. Combinaron las capacidades de representación de datos de la lógica difusa, y la capacidad de creación de modelos del "data mining" a partir de grandes cantidades de datos. De esta forma les fue posible utilizar patrones más flexibles y abstractos, más cercanos a la realidad, para la detección de intrusiones.

La lógica difusa permite representar conceptos que pueden pertenecer a varias categorías (categorías solapadas). En la teoría de conjuntos estándar, cada elemento es miembro de una categoría o no lo es. No obstante, la lógica difusa permite la pertenencia parcial a varias categorías.

Detección de Intrusiones Difusa

John E. D., Jukka J., Ourania K y Julie A. D., de la Universidad de Iowa, han extendido el concepto de *difuso*, desarrollando un detector de intrusiones basado en red bajo el nombre FIRE ("Fuzzy Intrusion Recognition Engine") [70]. Este sistema utiliza una arquitectura basada en agentes, mediante el sistema AAFID, mencionado en el apartado anterior, para distribuir la monitorización. Cada uno de ellos difumina sus entradas de datos. Se comunican con un motor de evaluación difuso, que relaciona los datos recibidos de los agentes mediante reglas difusas para producir diferentes grados de alarmas.

Los sistemas difusos son muy útiles en la detección de escaneo de puertos, y ataques de denegación de servicio, así como la actividad de diversos troyanos o puertas traseras.

Estas son algunas de las ventajas más importantes de los sistemas difusos:

- Pueden obtener y relacionar entradas de datos de orígenes variados.
- Tienen más matices para definir intrusiones difíciles de clasificar por otros sistemas más estrictos.
- Pueden emitir diferentes grados de alarmas.

Los sistemas difusos parecen tener un futuro prometedor en la detección de intrusiones. Sin embargo, es importante tener un alto grado de conocimiento en esta materia para poder afinar correctamente las reglas de detección. Por otra parte, el diseño de los agentes es fundamental en sistemas de este tipo. Su especialización y grado de detalle pueden determinar la identificación de nuevas formas de ataque con éxito.

3.2.3.3.6 Anomalías artificiales

La generación de anomalías artificiales es un modelo propuesto por Wenke Lee, Salvatore J. Stolfo, W. Fan, M. Miller, y P. K. Chan [71]. Consiste en utilizar algoritmos para fabricar anomalías nuevas, a partir de ataques ya conocidos de antemano. De esta forma, se intenta mejorar las capacidades de detección tanto de usos indebidos como de anomalías (detección de intrusiones conocidas y no conocidas). El detector de intrusiones propuesto está basado en red.

Uno de los aspectos que se destacan en el modelo es la necesidad de combinar los dos métodos de detección de intrusiones más comunes: el de clasificaciones, o usos indebidos y el de anomalías.

En circunstancias normales, para la generación de modelos de clasificación, se utiliza información de entrenamiento basada en clases de datos conocidas. La limitación de estos modelos es precisamente, que sólo reconocerán anomalías relacionadas con esas clases. En la detección de anomalías, la limitación es que sólo se utiliza una clase conocida (equivalente a pautas de comportamiento normal; por ejemplo, el número medio de conexiones por sesión), o algunas instancias limitadas de clases conocidas, con el objetivo de reconocer clases no conocidas.

Esta propuesta presenta formas de generar anomalías artificiales, basadas en clases conocidas, para proporcionar un algoritmo de aprendizaje que permita hallar formas de separar las clases conocidas de las no conocidas. Discute formas de generar modelos de detección de anomalías, a partir únicamente de datos normales. Y estudia el proceso de creación de modelos de detección combinada de usos indebidos y anomalías, a partir de datos que contienen clases conocidas.

Los resultados de los experimentos son muy positivos. Demuestran que el detector de anomalías, entrenado con anomalías conocidas y generadas artificialmente, es capaz de detectar más del 77% de todas las clases de intrusiones no conocidas, con más del 50% eficacia. Por otra parte, el detector combinado de usos indebidos y anomalías, es tan eficaz como el detector de usos indebidos simple, identificando intrusiones conocidas, y es capaz de detectar más del 50% de las clases de intrusiones no conocidas, con una eficacia de entre el 75% y el 100% por clase.

3.3 Respuesta

Los resultados obtenidos de la fase de análisis, se utilizan para tomar las decisiones que conducirán a una respuesta. Esta es la tercera y última fase del modelo de un sistema de detección de intrusiones. El conjunto de acciones y mecanismos que se pueden efectuar en esta etapa es amplio. A continuación se describirán los requisitos y tipos de respuesta más comunes.

3.3.1 Primeras consideraciones

Cuando se diseña un plan de respuesta ante posibles intrusiones hay que tener en cuenta el marco de trabajo. Una empresa puede estimar necesario cumplir con los estándares en gestión de seguridad y manejo de incidencias, mientras que un investigador de seguridad que experimenta con una red de laboratorio puede necesitar registrar de forma exhaustiva las actividades de la misma para su investigación. Los productos comerciales deberían ser lo suficientemente versátiles para poder atender las necesidades de los usuarios en este aspecto.

Una de las consideraciones a tener en cuenta al diseñar un mecanismo de respuesta es el entorno operacional en el que se va a utilizar. Un sistema de detección que deba coordinar la información de múltiples agentes, distribuidos a lo largo de una red de producción, no tiene las mismas necesidades de alarma y notificación que un sistema no distribuido, instalado en un ordenador personal.

El elemento monitorizado juega un papel importante en el modelo de respuesta. Una de las razones por las que se proporcionan respuestas activas, como el bloqueo de las conexiones del atacante, se debe a la existencia de sistemas que proporcionan funciones o servicios críticos a los usuarios. Un ejemplo de este caso es el de bancos o comercios electrónicos. Un ataque de denegación de servicio podría ser graves consecuencias en esos casos.

En determinados entornos, se cuenta con procedimientos preestablecidos de obligatorio cumplimiento. Algunas fuerzas militares poseen normas que definen los requisitos y funciones que deben satisfacer los detectores de intrusiones. Si en un determinado momento, el detector no está activo, se indica que dicho sistema no debe trabajar con información clasificada.

Por otra parte, las alarmas y avisos proporcionados por un mecanismo de respuesta deberían presentar suficiente información adicional para indicar las acciones a tomar en cada situación. Algunos productos de seguridad sólo indican el identificador del error mediante un lacónico mensaje, siendo más útil añadir el posible origen del problema y cómo solucionarlo.

3.3.2 Tipos de respuestas

Las respuestas de un sistema de detección de intrusiones pueden ser de dos tipos: *pasivas* o *activas*. Las pasivas consisten en la emisión de informes, o el registro de las intrusiones ocurridas. Las activas son las que implican alguna acción en particular, como el bloqueo de conexión, el cierre inmediato de una cuenta, o la prohibición de ejecución de determinados comandos.

Se pueden soportar ambos tipos de respuestas en un sistema de detección. Una no excluye a la otra. Es posible que en determinadas anomalías sólo sea necesario registrar la actividad ocurrida para su posterior examen, mientras que en intrusiones a sistemas críticos haga falta una actuación más activa y urgente. Como se comentó antes, todo depende de las necesidades de cada caso.

3.3.2.1 Respuestas activas

Las respuestas activas, como ya se mencionó, afectan al progreso del ataque, pueden ser llevadas a cabo de forma automática por el sistema, o mediante intervención humana.

Estas acciones pueden ser de diversa naturaleza; no obstante, la mayoría se pueden clasificar en estas tres categorías principales: ejecutar acciones contra el intruso, corregir el entorno, y recopilar más información.

3.3.2.1.1 Ejecutar acciones contra el intruso

La más famosa de las respuestas activas, es la de tomar acciones contra el intruso. La forma más directa consiste en identificar el origen de ataque, e impedirle el acceso al sistema. Por ejemplo, desactivando una conexión de red, o bloqueando la máquina comprometida.

Sin embargo, tomar decisiones tan agresivas no es siempre una buena solución. Hay situaciones en las que esto podría causar serios problemas:

- Los ataques recibidos a menudo son realizados, no desde la propia máquina del intruso, sino desde una víctima controlada por aquel. El intruso puede haber utilizado a su víctima mediante algún programa de control remoto, o como resultado de un fallo de seguridad que le permitiera entrar. Si se bloquea o incluso devuelve el ataque en esta situación, se estaría perjudicando a alguien inocente.
- En muchas ocasiones, los atacantes utilizan técnicas de ocultación de su dirección IP ("spoofing"). En este tipo de ataques, las direcciones IP de origen no tienen nada que ver con el atacante. Incluso pueden no existir, hecho beneficioso en casos de ataque de denegación de servicio, en los que el servidor pierde demasiado tiempo esperando la respuesta de direcciones IP que no responden nunca.
- Por otra parte, responder de forma automática a intrusiones puede provocar penalizaciones legales. Si se devuelve el ataque a una entidad inocente, puede efectuar una demanda por daños y perjuicios. Además, en algunas organizaciones, tomar este tipo de decisiones sin la autorización adecuada puede ser razón de despido.

Se pueden realizar acciones contra intrusos menos drásticas. Por ejemplo, terminando la sesión TCP problemática, o bloqueando durante un intervalo de tiempo el origen de las intrusiones, o enviando un correo electrónico al administrador.

Hay que tener en cuenta que ejecutar una respuesta definitiva, de forma automática, es algo delicado. Si en un determinado entorno se bloquean permanentemente las direcciones IP que intentan demasiadas conexiones con el servidor, y un intruso se percata de ello, puede elaborar un ataque que consista en realizar sucesivos intentos de conexión con direcciones falseadas, pertenecientes a los clientes más importantes. De esta forma, el propio detector de intrusiones estaría bloqueando a sus propios clientes, ayudando a un ataque de denegación de servicio.

3.3.2.1.2 Corregir el entorno

Esta opción, como su nombre indica, consiste en efectuar las acciones pertinentes para restaurar el sistema y corregir los posibles problemas de seguridad existentes. En la mayoría de las ocasiones, esta respuesta activa suele ser la acción más acertada.

Aquellos sistemas que cuentan con métodos de *auto-curación* ("self-healing"), son capaces de identificar el problema y proporcionar los métodos adecuados para corregirlo.

En muchas ocasiones, este tipo de respuestas puede provocar cambios en el motor de análisis o en los sistemas expertos, generalmente aumentando su sensibilidad e incrementando el nivel de sospecha ante posibles intrusiones.

3.3.2.1.3 Recopilar más información

Recoger información adicional es otra de las respuestas activas. Esta opción es utilizada en ocasiones para cumplir los requisitos de información necesarios para poder tomar acciones legales contra posibles criminales. Normalmente se aplica en sistemas que proporcionan servicios críticos.

Otra de las situaciones en las que se puede dar este tipo de respuesta, es en máquinas o redes que imitan comportamientos y servicios reales, para engañar a los intrusos. Tal es el caso de

servidores "decoy" (trampa), "fishbowl" (peceras), o "honeypots" (tarros de miel). Este último tipo, por su creciente rango de características, que lo hacen útil en el ámbito de la detección de intrusiones, es comentado más a fondo en el siguiente capítulo.

La puesta en práctica de un servidor "decoy" fue descrita primera vez a través de un artículo escrito por Bill Cheswick [72]. En él detallaba cómo había creado un servidor "decoy" para redirigir las acciones de un "cracker" holandés que atacaba sus sistemas. El uso de un servidor "decoy" fue comentado por Cliff Stoll en su libro "The Cuckoo's Egg" [73].

Las posibilidades derivadas del uso de servidores "decoy" o "honeypots" son muy amplias, y se explican con mayor detalle en el siguiente capítulo. Por describir algunas de ellas, pueden ser usados para registrar de forma exhaustiva las actividades de un intruso, y aprovechar los resultados para tomar medidas legales. Por otra parte, un servidor trampa puede utilizarse, como ya se ha hecho en alguna empresa, para detectar vulnerabilidades utilizadas por intrusos, aún no publicadas o ni siquiera descubiertas. Otra de las posibilidades que tienen estos sistemas, es la de ayudar a diseñar mejores patrones y reglas de detección, gracias a los registros de actividad obtenidos.

3.3.2.2 Respuestas pasivas

Son aquellas respuestas que consisten en el envío de información al responsable correspondiente, dejando recaer en él la toma de decisiones. En los primeros detectores de intrusiones, todas las respuestas eran pasivas. Aunque la tecnología ha evolucionado mucho desde entonces, las respuestas pasivas siguen existiendo. Y es que, por muy afinados que sean los mecanismos de respuesta automática, hay ocasiones en que un sistema no tiene la responsabilidad suficiente para tomar una decisión.

Las alarmas por pantalla son una de las alarmas más comunes entre los sistemas de detección de intrusiones. Un mensaje en una ventana indica al usuario que se ha cometido una posible intrusión, acompañando a veces el mensaje con información adicional, como la dirección del posible atacante, el protocolo usado, etc. Muchas veces, el contenido de estas alarmas puede ser configurado.

Otra de las posibles formas de recibir respuestas pasivas es a través del correo electrónico o mensajes a un teléfono móvil. La ventaja del segundo caso sobre el primero, es que puede ser recibido en cualquier lugar, cualidad muy apreciada entre administradores. No obstante, un correo electrónico puede contener más información, proporcionando un mensaje más extenso, y menos ambiguo, sobre la incidencia.

Por otra parte, algunos sistemas de detección de intrusiones están integrados con mecanismos de gestión de redes. En estos casos se suele hacer uso de mensajes SNMP (Protocolo Simple de Gestión de Red). La integración con este sistema de comunicación permite utilizar canales ya existentes para el envío de incidencias. No obstante, un uso excesivo por parte de detectores de intrusiones de estos canales, podría perjudicar a otros sistemas que también hicieran uso de ellos.

3.3.3 Observaciones sobre las respuestas

Para que la fase de respuesta de un detector de intrusiones tenga relativo éxito, hay que tener ciertos aspectos en cuenta. Desde que se detecta una actividad sospechosa, hasta que se comunica a la entidad correspondiente, se producen una serie de pasos en los que hay que salvar

diversos obstáculos. Mantener cierta seguridad en la comunicación de las respuestas, configurar adecuadamente el detector de intrusiones para minimizar los falsos positivos, o proporcionar métodos de almacenamiento apropiados para las notificaciones, son algunos de los que se describen a continuación.

3.3.3.1 Aspectos de seguridad

Los propios sistemas de respuesta pueden ser importantes objetivos de ataque. El bloqueo de las respuestas de un detector de intrusiones, dejaría a este sistema mudo, anulando toda su eficacia. Por ello, los productos desarrollados suelen dedicar especial atención a este apartado.

3.3.3.1.1 Comunicación oculta

Una de las cuestiones que los detectores de intrusiones tienen presente, es la de no ser reconocidos por atacantes. Cuando un sistema está siendo comprometido, no conviene que el intruso se percate de que está siendo monitorizado. Por esta razón, se utilizan técnicas que permitan al detector de intrusiones registrar todo lo sucedido y comunicar las incidencias a los responsables, todo ello sin ser interceptado.

En la mayoría de las ocasiones se suele utilizar canales de comunicación cifrados. De esta forma, el intruso no puede detectar ni modificar el contenido de las comunicaciones.

3.3.3.1.2 Redundancia

Otra de las soluciones recomendadas, es el uso de la redundancia en las comunicaciones. Es decir, en situaciones de alarma críticas, conviene utilizar más de una vía de comunicación para transmitir la misma notificación. Se puede enviar mensaje por un canal cifrado y otra a través de mensajes de gestión de red. Así, se reducen las probabilidades de que los mensajes sean bloqueados.

3.3.3.1.3 Protección de registros

Una vez comunicadas las alarmas hay que almacenarlas de forma segura, protegiéndolas ante alteraciones o eliminaciones. Esto es especialmente importante cuando se pretende utilizar el material obtenido para asuntos legales. También aquí se utiliza redundancia, en los casos más importantes.

Una de las soluciones practicadas es el almacenamiento de los registros en medios de una sola escritura, como un CD-ROM o incluso una impresora de papel continuo.

3.3.3.2 Falsas alarmas

Uno de los problemas asociados a los mecanismos de respuesta de los sistemas de detección de intrusiones es el de las falsas alarmas. Son de dos tipos: falsos positivos, que indican posibles intrusiones cuando en realidad no los hay; y falsos negativos, que no notifican intrusiones cuando realmente han tenido lugar.

Los falsos negativos representan un problema, en el sentido de que al sistema se le escapan muchas actividades sospechosas. La solución pasa por mejorar la sensibilidad del detector, afinando la configuración o desarrollando mejores técnicas de detección.

Los falsos positivos pueden ser también peligrosos si se producen con demasiada frecuencia. Pueden hacer que el responsable de seguridad acabe ignorándolos, o no distinguiendo

entre ellos las verdaderas alarmas. En el peor de los casos, pueden llegar a colapsar al sistema. Una de las posibles soluciones a esta situación consiste en disminuir la sensibilidad del motor de detección de anomalías, adaptando los resultados al comportamiento normal del sistema.

3.3.3.3 Almacenamiento de registros

Casi todos los detectores de intrusiones comerciales tienen métodos especiales para conservar los registros generados a través de sus mecanismos de respuesta pasiva. Buena parte de ellos contempla la posibilidad de almacenarlos en bases de datos. Esto simplifica en gran medida su análisis ulterior. Además, permite a los responsables de seguridad entregar informes detallados de la actividad del sistema a los encargados de la gestión ejecutiva.

Conviene que los registros de los sistemas de detección de intrusiones sean almacenados de forma similar a los registros de sistema, y en medios seguros, como ya se comentó en apartados anteriores. Uno de los motivos por los que se hace esto, es para ayudar en operaciones legales e investigaciones forenses.

3.3.4 Adopción de políticas de respuesta

La correcta gestión de la seguridad de un sistema de detección de intrusiones conlleva el cumplimiento de una serie de procedimientos, que definan las acciones a tomar en caso de problemas. Estos procedimientos deben estar incluidos en las políticas de seguridad de la organización. Las actividades contempladas por las políticas de seguridad, en caso de intrusiones o violaciones de seguridad, están divididas en cuatro categorías: intermedia o crítica, oportuna, largo plazo - local, y a largo plazo - global [23].

3.3.4.1 Intermedia o crítica

Estas son algunas de las acciones a realizar justo después de percibir un ataque o intrusión:

- Procedimientos de manejo de incidencias.
- Contención y control de daños.
- Notificación a las autoridades legales y otras organizaciones.
- Restaurar el servicio en los sistemas afectados.

3.3.4.2 Oportuna

A continuación se indican las acciones a tomar después de un ataque o violación de seguridad. El tiempo en que se llevan a cabo puede variar entre unas horas y varios días, dependiendo de la organización y el grado de importancia:

- Investigar manualmente patrones inusuales de uso del sistema.
- Investigar y aislar las causas del problema.
- Corregir estos problemas cuando sea posible, aplicando parches o corrigiendo la configuración del sistema.

- Notificar los detalles del incidente a los responsables adecuados.
- Mejorar las firmas o patrones de detección del sistema de detección de intrusiones.
- Tomar acciones legales contra el intruso.

3.3.4.3 Largo plazo - local

Las acciones de largo plazo local son menos críticas que las pertenecientes a las dos categorías anteriores. No obstante, no deben dejar de realizarse, ya que desempeñan un papel imprescindible en la seguridad del sistema. Se llevan a cabo a de forma periódica, durante la administración de un sistema.

Actividades como las descritas a continuación, permiten la prevención de posibles ataques, y mejoran de forma constante la configuración de los elementos de seguridad del sistema:

- Compilar estadísticas y realizar análisis de las tendencias de uso y comportamiento.
- Seguir la pista de patrones de forma continua.

3.3.4.4 Largo plazo - global

Estas acciones corresponden a actividades no críticas, pero que no deben ser ignoradas. El impacto de estas acciones no está limitado a la organización. Algunos aspectos de la seguridad de sistema no pueden ser resueltos de forma local. Acudir a otras entidades puede hacer que la organización sea partícipe de una solución más completa. Algunas de estas acciones son:

- Notificar a los vendedores de los posibles problemas de seguridad encontrados en sus productos.
- Acudir a entidades legisladoras y al gobierno para solicitar mejores medidas legales contra violaciones de seguridad de sistemas.
- Enviar estadísticas sobre incidentes de seguridad a autoridades legales u otras organizaciones que mantengan este tipo de estadísticas.

3.4 Referencias

- [1] National Computer Security Center. *Department of Defense Trusted Computer System Evaluation Criteria*. Orange Book, DOD 5200.28-std, December 1985.
- [2] Price, Katherine E. *Host-Based Misuse Detection and Conventional Operating Systems' Audit Data Collection*. Master thesis, Purdue University, W. Lafayette, IN, December 1997.
- [3] Sun Microsystems, Inc. *System Administration Guide: Security Services*. BSM (Overview), 2002. Santa Clara, CA.
- [4] Josué Kuri, Gonzalo Navarro, Ludovic Mé, Laurent Heye. *A Pattern Matching Based Filter for Audit Reduction and Fast Detection of Potential Intrusions*. Springer-Verlag Berlin Heidelberg, 2000.
- [5] IBM Research, Zurich Research Laboratory. Andreas Wespi, Marc Dacier, and Hervé Debar. *Intrusion Detection Using Variable-Length Audit Trail Patterns*. Springer-Verlag Berlin Heidelberg, 2000.

- [6] Garfinkel, S. and E.H. Spafford. *Practical UNIX and Internet Security*, Second Edition. O'Reilly and Associates, 1996: 290.
- [7] Schaefer, Marvin, B. Hubbard, D. Sterne, T. K. Haley, J. N. McAuliffe, and D. Woolcott. *Auditing: A Relevant Contribution To Trusted Database Management Systems*. Proceedings of the Fifth Annual Computer Security Applications Conference, Tucson, AZ, December 1989.
- [8] World Wide Web Consortium. *Logging Control In W3C httpd - The Common Logfile Format*. [en línea]. Julio 1995 [consultado en febrero 2003]. Disponible en <<http://www.w3.org/Daemon/User/Config/Logging.html>>.
- [9] World Wide Web Consortium. *HTTP - Hypertext Transfer Protocol*. [en línea]. Julio 1995 [consultado en febrero 2003]. Disponible en <<http://www.w3.org/Protocols/>>.
- [10] RENT-A-GURU. *HTTP-ANALYZE - A Log Analyzer for web servers*. [en línea]. 27 de septiembre, 2002 [consultado en febrero 2003]. Disponible en <<http://www.http-analyze.org/>>.
- [11] Sanctum Inc. *Appshield*. [en línea]. Fecha no disponible [consultado en abril 2003]. Disponible en <<http://www.sanctuminc.com>>.
- [12] Kim Gene H. and E. H. Spafford. *Tripwire: A Case Study in Integrity Monitoring*. Internet Beseiged: Countering Cyberspace Scofflaws; edited by Dorothy and Peter Denning, Addison-Wesley, 1997.
- [13] González Gómez, Diego. *Cables UTP de sólo recepción*. [en línea]. Junio, 2003 [consultado en junio 2003]. Disponible en <<http://www.dgonzalez.net/secinf>>.
- [14] The Moschovitis Group. *History of the Internet: Birth of TCP/IP Networking Protocol*. [en línea]. 1999 [consultado en febrero 2003]. Extractos del Capítulo 4 - Because It's there: 1979-1984. Disponible en <<http://www.historyoftheinternet.com/chap4.html>>.
- [15] Stevens, W. R. *TCP/IP Illustrated*, Volume 1. Addison-Wesley, Reading, Massachusetts, 1994.
- [16] Kirkpatrick S., Stahl M., Recker M., *Internet Numbers*. Internet Engineering Task Force. [en línea]. Julio, 1990 [consultado en febrero, 2003]. Request for Comments: 1166. [182 pp.] Disponible desde Internet <<http://www.ietf.org/rfc/rfc1166.txt>>
- [17] Postel, J., *Internet Protocol*. Internet Engineering Task Force. [en línea]. Septiembre, 1981 [consultado en febrero, 2003]. Request for Comments: 791. [45 pp.] Disponible desde Internet: <<http://www.ietf.org/rfc/rfc791.txt>>. También disponible en castellano en <<http://www.rfc-es.org/rfc/rfc0791-es.txt>>.
- [18] Postel, J., *Transmission Control Protocol*. Internet Engineering Task Force, [en línea]. Septiembre, 1981 [consultado en febrero, 2003]. Request for Comments: 793. [85 pp.] Disponible desde Internet: <<http://www.ietf.org/rfc/rfc793.txt>>. También disponible en castellano en <<http://www.rfc-es.org/rfc/rfc0793-es.txt>>.
- [19] McCanne, S. and V. Jacobson. *The BSD Packet Filter: A New Architecture for User Level Packet Capture*. 1993 Winter USENIX Conference, San Diego, CA, Enero 1993.
- [20] Ranum, M. J., K. Landfield, M. Stolarchuk, M. Sienkiewicz, A. Lambeth, and E. Wall. *Implementing a Generalized Tool for Network Monitoring*. Proceedings of the Eleventh Systems Administration Conference (LISA '97). San Diego, CA, October 1997.
- [21] Driver Development for SCO Systems. *Streams Driver Overview*. The Santa Cruz Operation, Santa Cruz, CA, 1999.
- [22] Sunstrom, Peter. *fwlogsum - Firewall Report Summariser*. [en línea]. Actualizado el 1 de marzo 2003 [consultado en Marzo de 2003] Actualizado continuamente. Disponible desde Internet en <<http://www.ginini.com.au/tools/fw1/>>
- [23] Bace, Rebecca. *Intrusion Detection*. Macmillan Technical Publishing, 2000.

- [24] Lunt, T., A. Tamaru, and F. Gilham. *IDES: A Progress Report*. Proceedings of the Sixth Annual Computer Security Applications Conference, Tucson, AZ, December 1990.
- [25] Jackson, P. *Introduction to Expert Systems*. International Computer Science Series. Addison Wesley, 1986.
- [26] Roesch, Marty et al. *Snort.org*. [en línea]. Actualizado semanalmente [consultado en marzo de 2003]. Disponible en <<http://www.snort.org>>.
- [27] Paxson, Vern. *Bro: A System for Detecting Network Intruders in Real-Time*. Lawrence Berkeley National Laboratory, Berkeley, CA and AT&T Center for Internet Research at ICSI, Berkeley, CA. [en línea]. 14 de diciembre de 1999 [consultado en marzo de 2003]. Disponible desde Internet en <<http://www.icir.org/vern/bro-info.html>>
- [28] Porras, Phiffip. *STAT, a State Transition Analysis Tool for Intrusion Detection*. Master thesis, Computer Science Department, University of California, Santa Barbara, CA, July 1992.
- [29] IBM Research, *Zurich Research Laboratory*. *Andreas Wespi, Marc Dacier, and Hervé Debar*. Intrusion Detection Using Variable-Length Audit Trail Patterns. Springer-Verlag Berlin Heidelberg, 2000.
- [30] Ilgun, Koral. *USTAT: A Real-Time Intrusion Detection System for UNIX*. Master thesis, University of California, Santa Barbara, CA, November 1992.
- [31] Kumar, S. and E.H. Spafford. *A Pattern Matching Model for Misuse Intrusion Detection*. Proceedings of the Seventeenth National Computer Security Conference, Baltimore, MD, October 1994.
- [32] Mounji, A. *Languages and Tools for Rule-Based Distributed Intrusion Detection*. Thesis, Faculte's Universitaires Notre-Dame de la Paix, Namur, Belgium, September 1997.
- [33] Smaha, Stephen E. and S. Snapp. *Method and System for Detecting Intrusion into and Misuse of a Data Processing System*. US555742, U.S. Patent Office, September 17, 1996.
- [34] NFR. *Network Flight Recorder*. [en línea]. Fecha no disponible [consultado en marzo, 2003]. Disponible desde Internet <<http://www.nfr.net>>.
- [35] Yao-Tsung Lin, Shian-Shyong Tseng And Shun-Chieh Lin. *An Intrusion Detection Model Based Upon Intrusion Detection Markup Language (IDML)*. Department of Computer and Information Science, National Chiao Tung University, Taiwan, agosto de 2001.
- [36] Anderson, Ross, and A. Khattak. *The Use of Information Retrieval Techniques for Intrusion Detection*. Presentation, First International Workshop on the Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium, September 1998.
- [37] Denning, Dorothy E. *An Intrusion Detection Model*. Proceedings of the 1986 IEEE Symposium on Security and Privacy, Oakland, CA, April 1986.
- [38] Hochberg, Judith, K. Jackson, C. Stallings, J. F. McClary, D. DuBois, and J. Ford. *NADIR: An Automated System for Detecting Network Intrusion and Misuse*. Computers and Security 12, no. 3 (May 1993): 235 - 248.
- [39] Lemmonier, E. *Protocol Anomaly Detection in Network-based IDSs*. Defcom, Sweden, Stockholm, 28 de junio de 2001.
- [40] Smaha, Stephen E. *Haystack: An Intrusion Detection System*. Proceedings of the IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL, December 1988.
- [41] Mukherjee, Biswanath, L. T. Heberlein, and K. N. Levitt. *Network Intrusion Detection*. IEEE Network 8, no. 3 (May - June 1994): 26 - 41.
- [42] Javitz, Harold S. and Valdes, A. *The SRI IDIES Statistical Anomaly Detector*. Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1991.

- [43] Hoagland, J., S. Staniford. Silicon Defense. *SPICE / SPADE*. [en línea]. Actualizado con frecuencia [consultado en marzo, 2003]. Disponible desde Internet en <<http://www.silicondefense.com/software/spice/>>.
- [44] Sekar, R., M. Bendre, D. Dhurjati, P. Bollineni, *A Fast Automaton-based Method for Detecting Anomalous Program Behaviors*. Proceedings of the 2001 IEEE Symposium on Security and Privacy.
- [45] Mahoney, Matthew, P. K. Chan. *PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic*. Florida Tech. technical report 2001-04. Disponible en <<http://cs.fit.edu/~tr/>>.
- [46] Mahoney, Matthew, P. K. Chan. *Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks*. Edmonton, Alberta: Proc. SIGKDD, 2002, 376-385.
- [47] Mahoney, Matthew, P. K. Chan. *Learning Models of Network Traffic for Detecting Novel Attacks*. Florida Tech. technical report 2002-08. Disponible en <<http://cs.fit.edu/~tr/>>
- [48] Mahoney, Matthew. *Network Traffic Anomaly Detection Based on Packet Bytes*. Florida Institute of Technology. Melbourne, Florida, 2003.
- [49] Lankewicz, Linda and M. Benard. *Real-Time Anomaly Detection Using a Nonparametric Pattern Recognition Approach*. Proceedings of the Seventh Computer Security Applications Conference, San Antonio, TX, December 1991.
- [50] Liepins, G. E. and H. S. Vaccaro. *Intrusion Detection: Its Role and Validation*. Computers and Security, v 11, Oxford, UK: Elsevier Science Publishers, Ltd, 1992: 347 - 355.
- [51] Teng H. S., Kaihu Chen and Stephen C. Lu. *Security Audit Trail Analysis Using Inductively Generated Predictive Rules*. Proceedings of the 11th National Conference on Artificial Intelligence Applications, pages 24-29, IEEE, IEEE Service Center, Piscataway, NJ, March 1990.
- [52] Teng, H. S., K. Chen, and S. C. Y. Lu. *Adaptive Real-Time Anomaly Detection Using Inductively Generated Sequential Patterns*. Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1990.
- [53] Lunt, Teresa F. *A Survey of Intrusion Detection Techniques*. Computers and Security 12, 4 (June 1993): 405-418.
- [54] Grediaga, A., Ibarra, F., Ledesma, B., Brotons, F. *Utilización de redes neuronales para la detección de intrusos*. Departamento de Tecnología Informática y Computación. Universidad de Alicante.
- [55] Fox K. L., R. Henning, J. Reed. *A Neural Network Approach Towards Intrusion Detection*. Proceedings of the 13th National Computer Security Conference. Pp 125-134 Washintong, DC, October 1990.
- [56] Debar, Herve, M. Becker, and D. Siboni. *A Neural Network Component for an Intrusion Detection System*. Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1992.
- [57] Hofmeyr, Steven A., S. Forrest, and A. Somayaji. *Intrusion Detection Using Sequences of System Calls*. Journal of Computer Security 6, no. 3 (1996): 151 - 180.
- [58] Warrender, C., S. Forrest, and B. Pearimutter. *Detecting Intrusions Using System Calls: Alternative Data Models*. Proceedings of Twenty-Fifth IEEE Symposium on Security and Privacy, Oakland, CA, May 1999.
- [59] Pinacho, P., Contreras, R. *Una propuesta de Sistemas para Tratamiento de Intrusos Inspirado en la Biología*. Universidad de Santiago de Chile. Facultad de Ingeniería. Universidad de Concepción, Facultad de Ingeniería.
- [60] Lane, Terran and Carla E. Brodley. *An Application of Machine Learning to Anomaly Detection*. Proceedings of the Twentieth National Information System Security Conference, Baltimore, MD, October 1997.

- [61] Lane, Terran and Carla E. Brodley. *Detecting the Abnormal: Machine Learning in Computer Security*. Purdue University, January 1997.
- [62] Lane, Terran and Carla E. Brodley. *Sequence Matching and Learning in Anomaly Detection for Computer Security*. Purdue University, 1997.
- [63] Goldberg, David E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley January 1989.
- [64] Mé, Ludovic. *GASSATA, a Genetic Algorithm as an Alternative Tool for Security Audit Trails Analysis*. First International Workshop on the Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium, September 1998.
- [65] Lee, Wenke, S. J. Stolfo, and K. W. Mok. *A Data Mining Framework for Building Intrusion Detection Models*. Proceedings of the Twentieth IEEE Symposium on Security and Privacy, Oakland, CA, 1999.
- [66] Lee, Wenke, and Salvatore J. Stolfo. *Adaptive Intrusion Detection: a Data Mining Approach*. Computer Science Department, Columbia University, 2000.
- [67] Lee, Wenke, Salvatore J. Stolfo, Kui W. Mok. *Mining Audit Data to Build Intrusion Detection Models*. Computer Science Department, Columbia University, August 1998.
- [68] Lee, Wenke, Rahul A. Nimbalkar, Kam K. Yee, Sunil B. Patil, Pragneshkumar H. Desai, Thuan T. Tran, and Salvatore J. Stolfo. *A Data Mining and CIDF Based Approach for Detecting Novel and Distributed Intrusions*. Computer Science Department, North Carolina State University. Computer Science Department, Columbia University
- [69] Bridges, Susan M. and Rayford B. Vaughn. *Fuzzy Data Mining and Genetic Algorithms applied to Intrusion Detection*. Mississippi State University.
- [70] Dickerson, John E., Jukka Juslin, Ourania Koukousoula, Julie A. Dickerson. *Fuzzy Intrusion Detection*. Electrical and Computer Engineering Department. Iowa State University.
- [71] Lee, Wenke, Wei Fan, Matthew Miller, Salvatore J. Stolfo, Philip K. Chan. *Using Anomalies to Detect Unknown and Known Network Intrusions*. College of Computing. Georgia Tech. IBM T.J. Watson Research. Columbia University. Computer Science, Florida Tech. November 2001.
- [72] Cheswick, William. *An Evening with Berferd in Which a Cracker Is Lured, Endured, and Studied*. Proceedings of USENIX Security Conference, San Francisco, CA, Winter 1992.
- [73] Stoll, Clifford. *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*. New York, NY: Doubleday, 1989.

Complementos y casos especiales

La gran cantidad de formas de abordar el problema de la detección de intrusiones, ha dado lugar a numerosas y variadas propuestas y soluciones. Algunas herramientas de seguridad pueden ayudar a complementar la tarea de los detectores de intrusiones, como los comprobadores de integridad de ficheros en los detectores de intrusiones basados en máquina. Por otro lado, algunos sistemas de seguridad ofrecen soluciones tan similares, que muchos los consideran casos especiales de detectores de intrusiones.

En este capítulo se describirán algunos mecanismos y herramientas de seguridad que por sus características, poseen, de una u otra forma, una estrecha relación con la detección de intrusiones.

4.1 Escáner de vulnerabilidades

Un escáner de vulnerabilidades ("vulnerability scanner" o "assessment system") es una herramienta que realiza un conjunto de pruebas (generalmente ataques) para determinar si una red o un "host" tiene fallos de seguridad.

Este tipo de sistemas se podría considerar como un caso especial de detectores de intrusiones. Ya en apartado sobre información recogida de objetivos del capítulo 2 "Definiciones", se comentaba la diferencia entre un enfoque *estático* y uno *dinámico*. Un escáner de vulnerabilidades es un ejemplo de enfoque estático. Un análisis estático, o *basado en intervalo*, no trabaja de forma continua (un vídeo), sino en intervalos de tiempo (una imagen).

Es fácil darse cuenta de las debilidades que tiene un sistema como este. Sólo puede detectar aquellas vulnerabilidades contenidas en su base de datos. Además, sólo es capaz de identificar fallos de seguridad en los intervalos en que se ejecuta. No obstante, ello no le impide ser de inestimable ayuda a la hora de mejorar la seguridad de un sistema.

4.1.1 Proceso de análisis

A continuación se describe el proceso general de funcionamiento de un escáner de vulnerabilidades:

- Se muestrea un conjunto específico de atributos de sistema.
- Los resultados del muestreo se almacenan en un recipiente de datos seguro.
- Los resultados se organizan y comparan con al menos un conjunto de referencia de datos (este conjunto puede ser una plantilla de "configuración ideal" generada manualmente, o bien ser una imagen del estado del sistema hecha con anterioridad)
- Se genera un informe con las diferencias entre ambos conjuntos de datos.

Algunos programas comerciales optimizan el proceso mediante:

- Ejecución de motores de de comparación en paralelo.
- Utilización de métodos criptográficos para detectar cambios en los objetos monitorizados.

4.1.2 Tipos de análisis de vulnerabilidades

Existen dos formas de clasificar los análisis de vulnerabilidades; bien mediante la localización desde la que se obtienen datos o bien mediante el nivel de confianza del que hace uso el analizador de vulnerabilidades. Según el primer método, los análisis pueden ser *basados en máquina* o *basados en red*. Según el segundo método, los análisis pueden ser *con acreditaciones* ("credentialed") o *sin acreditaciones* ("non credentialed") [1]. Estos últimos análisis se refieren al hecho de utilizar credenciales de sistema, tales como contraseñas, durante el proceso de análisis [2]. La Figura 4-1 ilustra varias formas de clasificar estas herramientas de seguridad.

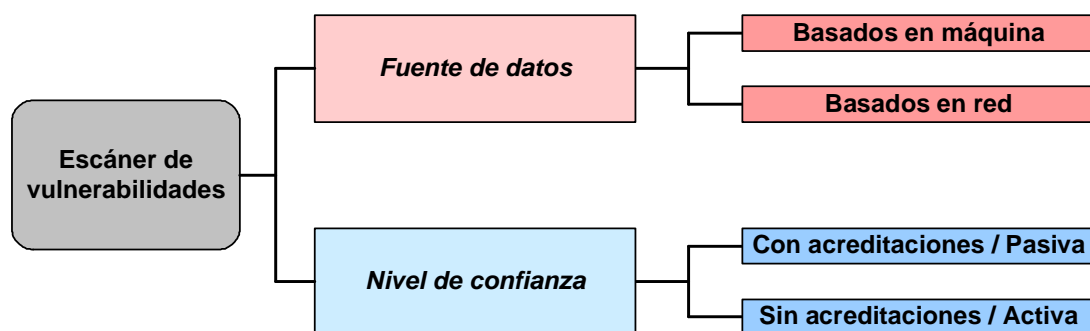


Figura 4-1 - Clasificación de analizadores de vulnerabilidades

A continuación se utilizará el primer enfoque descrito (basado en el origen de los datos), como método de clasificación para la descripción de estos sistemas.

4.1.2.1 Análisis de vulnerabilidades basado en máquina

Este tipo de análisis utiliza fue el primero en utilizarse en la evaluación de vulnerabilidades. Utiliza elementos tales como ajustes de configuración, contenidos de ficheros, u otro tipo de información de un sistema para la detección de vulnerabilidades. Esta información se puede obtener básicamente mediante consultas al sistema, o a través de la revisión de diferentes atributos del sistema.

En este caso se asume que el analizador de vulnerabilidades tiene acceso autorizado al sistema, por lo que también recibe el nombre de análisis *con acreditaciones*. También se denomina evaluación *pasiva*.

Bajo sistemas UNIX, la información es obtenida a nivel de host o de dispositivo. No obstante, los sistemas Windows permiten realizar muchas llamadas nativas de forma local o remota, según las credenciales utilizadas. La correspondencia entre sistemas *basados en máquina y autenticados* no siempre se cumple.

Las vulnerabilidades que se suelen encontrar mediante la evaluación basada en máquina, suelen ser estar relacionadas con ataques de *escalamiento de privilegios*. Estos ataques persiguen obtener permisos de *root* o "superusuario" en sistemas UNIX, o administrador en sistemas Windows.

Los motores de análisis de vulnerabilidades basados en máquina están muy relacionados con el sistema operativo que evalúan, lo cual hace su mantenimiento costoso y complica su administración en entornos heterogéneos. Las credenciales utilizadas deben ser protegidas convenientemente, así como la información accedida mediante las mismas, para evitar que sean objeto de ataques.

Un ejemplo de analizador de vulnerabilidades basado en máquina es el "Cerberus Internet Scanner (CIS)" [3]. Es un escáner gratuito desarrollado por Cerberus Information Security, Ltd. Es un programa simple pero eficaz, que comprueba una completa lista de vulnerabilidades conocidas para sistemas Windows, mostrando en poco tiempo los fallos existentes.

4.1.2.2 Análisis de vulnerabilidades basado en red

Los análisis de vulnerabilidades basados en red han aparecido hace algunos años, y se han ido haciendo cada vez más populares. Obtienen la información necesaria a través de las conexiones de red que establecen con el objetivo. Realizan conjuntos de ataques y registran las respuestas obtenidas. No se debe confundir los analizadores de vulnerabilidades basados en red con los sistemas de detección de intrusiones. Aunque un analizador de estas características puede ser muy similar a un detector de intrusiones, no representa una solución tan completa.

Esta forma de análisis normalmente no tiene requerimientos de autenticación; los ataques se realizan sin tener necesariamente permiso de acceso al sistema. Por esta razón, este tipo de enfoque también suele denominar *sin acreditaciones*. Además, el análisis se realiza atacando o sondeando efectivamente al objetivo, por lo que también recibe el nombre de evaluación *activa*.

La correlación descrita entre los enfoques basados en máquina y los autenticados, también se aplica a los basados en red y los no autenticados.

Se suelen utilizar dos técnicas, descritas a continuación, para la evaluación de vulnerabilidades basadas en red:

- Prueba por explotación ("Testing by exploit"): Esta técnica consiste en lanzar ataques reales contra el objetivo. Estos ataques están programados normalmente mediante guiones de comandos. En vez de aprovechar la vulnerabilidad para acceder al sistema, se devuelve un indicador que muestra si se ha tenido éxito o no. Obviamente, este tipo de técnicas es bastante agresivo, sobretodo cuando se prueban ataques de denegación de servicio. No obstante, no sólo ese tipo de ataques puede provocar la caída del sistema.
- Métodos de inferencia: El sistema no explota vulnerabilidades, sino que busca indicios que indiquen que se han realizado ataques. Es decir, busca resultados de posibles

ataques en el objetivo. Este método es menos agresivo que el anterior. No obstante, los resultados obtenidos son menos exactos que los que utilizan "exploits" [1]. Ejemplos de técnicas de inferencia pueden ser la comprobación de versión de sistema para determinar si existe una vulnerabilidad, la comprobación del estado de determinados puertos para descubrir cuáles están abiertos, y la comprobación de conformidad de protocolo mediante solicitudes de estado.

Uno de los productos más conocidos entre los analizadores de vulnerabilidades basados en red es Nessus [3]. Es un escáner de seguridad de red de Software Libre, licenciado bajo GNU "General Public License" (más tarde Lesser GPL). Fue desarrollado por Renaud Deraison en Europa en 1998. Su precursor fue SATAN, otro analizador, escrito por Wietse Venema y Dan Farmer [5].

Nessus es una herramienta basada en el modelo cliente-servidor que cuenta con su propio protocolo de comunicación. De forma similar a SATAN, el trabajo correspondiente a escanear y atacar objetivos es llevada a cabo por el servidor, mientras que las tareas de control y presentación de los datos son gestionadas por el cliente.

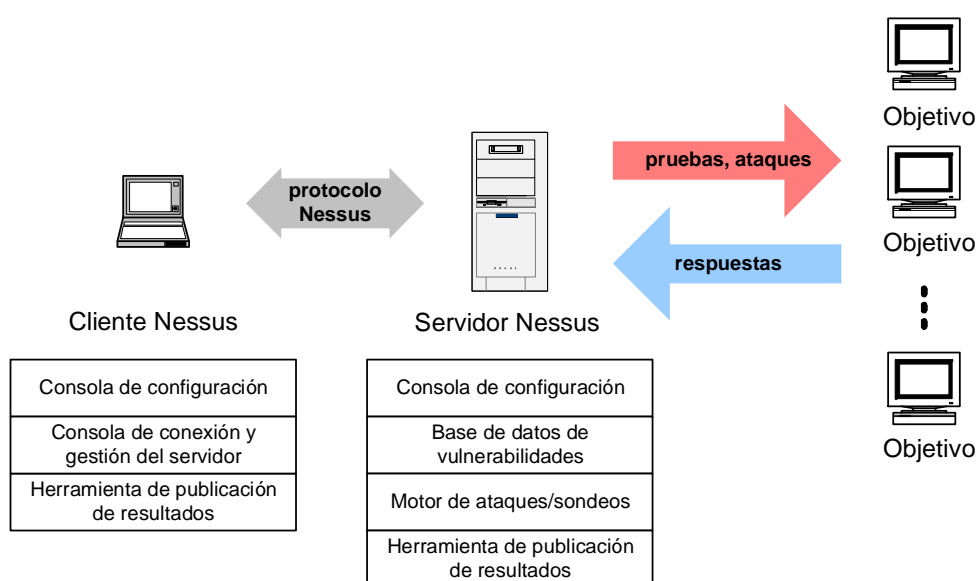


Figura 4-2- Arquitectura de Nessus

Nessus utiliza su propio lenguaje, denominado NASL (Nessus Attack Scripting Language), para definir las pruebas de vulnerabilidad. Su sencillez permite mantener la base de vulnerabilidades actualizada sin demasiado esfuerzo. Cada prueba de seguridad se escribe como un módulo externo, haciendo posible su programación sin conocer los detalles del motor de análisis.

Una de las ventajas de Nessus frente a otros competidores, es que además de ser de libre distribución, está escrito con interfaces claras y APIs (Application Program Interface), lo que le permite su integración con otras herramientas.

4.1.2.3 "Password cracking"

Otro de los métodos utilizados por los analizadores de vulnerabilidades (que no tiene por qué encajar en las anteriores clasificaciones) es el conocido como "password cracking" (romper/adivinar contraseñas). Esta técnica la incluyen muchos productos de análisis de vulnerabilidades.

El proceso de romper contraseñas permite determinar el grado de calidad de las contraseñas de los usuarios de sistema. En este proceso se utilizan funciones relacionadas con el sistema de autenticación de usuario del sistema operativo. Para adivinar las contraseñas, se prueban diferentes combinaciones de caracteres, o una serie de palabras tomadas de una lista (por ejemplo, un diccionario).

La calidad de las contraseñas dependerá de factores como la longitud de las mismas o la variedad de caracteres utilizados (mayúsculas, minúsculas, números, símbolos especiales, etc.). Cuanto más larga y mayor sea la variedad de caracteres utilizados en una contraseña, más difícil será de romper y, por tanto, de mayor calidad será.

4.1.2.4 Ventajas e inconvenientes

Un escáner de vulnerabilidades es una herramienta muy útil a la hora de ampliar la seguridad de un sistema. Tiene valiosas características de las que carecen otros enfoques más dinámicos. Sin embargo, como siempre que se habla de seguridad, no es la solución definitiva. Siempre necesitará complementarse con alguna herramienta o sistema que contrarreste sus debilidades.

4.1.2.4.1 Ventajas

- Los analizadores de vulnerabilidades mejoran de forma significativa la seguridad de un sistema, especialmente en entornos en los que no se cuenta con un sistema de detección de intrusiones.
- Un escáner de vulnerabilidades reduce eficazmente los fallos de seguridad más comunes de un sistema. Alarma de forma precisa muchos problemas de configuración que se le pueden pasar por alto a un administrador de sistemas o a un gestor de seguridad.
- Los analizadores de "host", que son dependientes del sistema operativo, están mejor adaptados a su objetivo. Esto les permite identificar de forma más eficaz que otros sistemas, más generales, ataques o signos de intrusiones particulares.
- El uso periódico de análisis de vulnerabilidades permite detectar cambios en las sucesivas configuraciones del sistema, informando de los cambios a los responsables de seguridad.

4.1.2.4.2 Inconvenientes

- Los análisis de vulnerabilidades basados en máquina, debido a su dependencia del sistema operativo que evalúan, son más costosos y complicados de gestionar.
- Los analizadores de vulnerabilidades basados en red son independientes de la plataforma, pero también son menos exactos y propensos a emitir falsas alarmas en sus resultados.

- En escenarios en los que se están ejecutando sistemas de detección de intrusiones, los análisis de vulnerabilidades pueden ser bloqueados por aquellos. Esta situación se agrava cuando se corre el riesgo de entrenar de forma errónea a los mecanismos de detección de intrusiones, haciendo que ignoren ataques reales.
- Algunas pruebas basadas en red, como los ataques de denegación de servicio pueden llevar a provocar la caída del objetivo. Este tipo de pruebas deben hacerse de forma controlada, conociendo de antemano los posibles efectos negativos que puedan tener.

4.2 "Honeypot", "Honeynet" y "Padded Cell"

Los sistemas descritos a continuación presentan un enfoque innovador con respecto a los sistemas de seguridad tradicionales. En vez de repeler las acciones de los atacantes, utilizan técnicas para monitorizarlas y registrarlas, para así aprender de ellos. A pesar de que en algunos países no están claramente definidos los aspectos legales de estos sistemas, lo cierto es que cada vez son más utilizados.

4.2.1 "Honeypot"

Durante los últimos años han ido adquiriendo creciente popularidad los denominados "honeypots" (sistemas trampa): "Recursos de sistema de información cuyo valor reside en el uso no autorizado o ilícito de dichos recursos". [6]

Un "honeypot" no es un sistema de detección de intrusiones, pero puede ayudar a mejorar sus métodos de detección y aportar nuevos patrones de ataque. Es un sistema diseñado para engañar a los intrusos, poder estudiar sus actividades, y así aprender de sus métodos. Se basa en la idea de "conocer al enemigo" para poder combatirlo. [7]

El concepto de *sistema trampa* no es nuevo, y ya fue introducido hace años por Cliff Stoll en su "Cuckoo's Egg" [8] y por Bill Cheswik [9]. En ambos trabajos se comentaba el uso de mecanismos similares a trampas, que servían para monitorizar la actividad de los intrusos permitiendo su posterior análisis. Más tarde apareció el término "honeypot", pero la idea era la misma.

Los sistemas trampa están diseñados para imitar el comportamiento de aquellos sistemas que puedan ser de interés para un intruso. Suelen contar con mecanismos de protección para que un atacante con éxito no pueda acceder a la totalidad de la red. Naturalmente, si un intruso consigue entrar en un sistema trampa, no debe percatarse de que está siendo monitorizado o engañado.

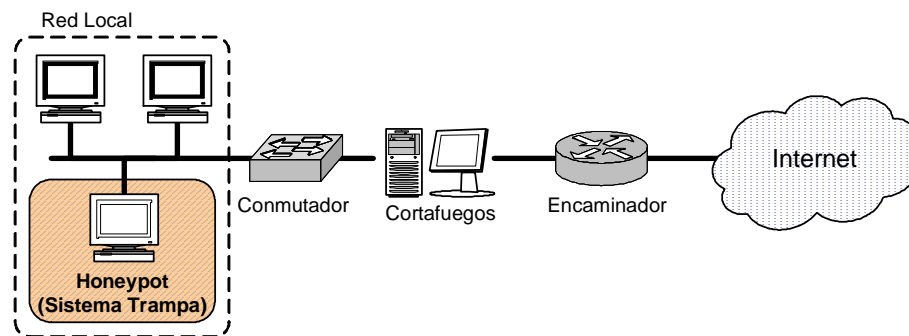


Figura 4-3 - Ejemplo de un "HoneyPot" (Sistema trampa)

La mayoría de los sistemas trampa están instalados detrás de un cortafuegos, aunque también es posible situarlos delante de ellos. El cortafuegos responsable del tráfico de un sistema trampa suele programarse para permitir las conexiones entrantes al sistema, y limitar las conexiones salientes.

4.2.1.1 Ventajas e inconvenientes

Los sistemas trampa poseen una serie de características que los distinguen claramente de otras soluciones de seguridad [10]. A continuación se describen algunas de las ventajas e inconvenientes asociados a los mismos.

4.2.1.1.1 Ventajas

- **Pocos y valiosos datos:** Los sistemas trampa registran poco volumen de datos, pero de mucho valor. Un sistema trampa no se utiliza como sistema de producción, sino únicamente como objeto de ataques. Por lo tanto, no suele registrar cantidades importantes de información. Sin embargo, toda esta información es útil, porque está relacionada con actividades hostiles. Esto hace, además, que los datos obtenidos sean claros y fácilmente analizables.
- **Falsas alarmas:** La filosofía de los sistemas trampa elimina la existencia de actividad normal o de producción en los mismos. Estas herramientas de seguridad sólo deben recibir únicamente actividades sospechosas. Esto reduce significativamente el número de falsos positivos (alarma cuando no existe ataque) y falsos negativos (omisión de alarma cuando hay verdaderamente un ataque).
- **Recursos:** Este tipo de herramientas no hace análisis de las actividades que registran. Por tanto, los recursos que consumen son reducidos, al contrario que muchos IDSs, que pueden llegar a descartar información por esta razón. Los sistemas trampa centran sus necesidades en la infraestructura necesaria para poder registrar toda la actividad que tenga lugar en ellos.
- **Simplicidad:** Uno de los puntos más importantes a favor de los sistemas trampa es su sencillez. No utilizan complicados algoritmos de análisis, ni rebuscados métodos para registrar la actividad de los intrusos. Por el contrario, sólo hay que instalarlos y esperar. Algunos sistemas trampa de desarrollo pueden poseer mayor nivel de complejidad, pero no

comparable a otros enfoques. Cuanto sencillo es un método, más posibilidades tiene de funcionar.

- **Encriptación:** Los problemas de monitorización relacionados con protocolos de encriptación (SSH, SSL, IPSec, etc.) aparecen cuando se intercepta una comunicación entre dos entidades, protegida mediante cifrado. Los sistemas trampa suelen ser uno de los extremos de la comunicación cifrada durante una intrusión o ataque. Además, registran en todo momento la actividad ocurrida.
- **Reutilización:** La mayoría de los productos de seguridad necesitan mantener al día sus mecanismos de detección y defensa para mantener su efectividad. Si no se renuevan, dejan de ser útiles. No obstante, los sistemas trampa, debido a su propia naturaleza, siempre serán de ayuda. Independientemente del tiempo que pase, siempre habrá atacantes dispuestos a comprometer estos sistemas de una u otra forma, mostrando el nivel de actividad de este sector y los métodos que utilizan.
- **IPv6:** Uno de los problemas que presentan algunas herramientas de seguridad es que no soportan el protocolo IPv6, sucesor del actual IPv4 ampliamente utilizado en Internet. Este protocolo está siendo principalmente utilizado en países asiáticos como Japón. Utilizar IPv6 utilizando túneles sobre IPv4, como hacen algunos atacantes, puede imposibilitar la detección por parte de muchos sistemas de detección. No obstante, los sistemas trampa registran toda la actividad ocurrida, por lo que se pueden identificar este tipo de ataques. Como ya se comentó, los sistemas trampa no registran grandes volúmenes de datos.

4.2.1.1.2 Desventajas

- **Punto de vista limitado:** Los sistemas trampa carecen de valor si no reciben ataques. Si un atacante logra identificar uno de estos sistemas, puede anular toda su efectividad evitándolos. Los sistemas trampa pueden no ser atacados, aún estando situados en la misma red que otros sistemas de producción que sí pueden ser objeto de ataques.
- **Riesgo:** Si un sistema trampa es atacado con éxito puede ser utilizado por el intruso para acceder al resto de sistemas de la red en que está instalado. El riesgo varía según el grado de complejidad del sistema trampa; cuanto más sencillo es, menores riesgos implica. Este aspecto es crítico a la hora de implementar este tipo de sistemas. Y siempre se toman medidas para minimizar este factor.
- **"Finger print" (Huella dactilar):** "Fingerprinting" consiste básicamente en la identificación, local o remota, de un sistema o servicio. Esto hace a través de diversos métodos, como por ejemplo realizando un escaneo de puertos, o enviando peticiones de solicitud de versión, u observando las respuestas del sistema ante determinados comandos. Es posible que la deficiente implementación de un sistema trampa lo delate, haciéndolo reconocible ante un intruso. Por ejemplo, si un sistema trampa que emula un servidor FTP, no implementa bien un comando como "prompt", y sí reconoce "propmt", esto se convierte en un patrón que lo hace identificable a los ojos de un atacante. Un sistema trampa, como ya se comentó, pierde eficacia cuando es reconocido por un atacante, convirtiéndose incluso en una herramienta que puede ser utilizada por éste para desviar la atención de un administrador de seguridad. No obstante, en ocasiones, un atacante puede cesar en sus actividades al percatarse de la existencia de un sistema de estas características.

4.2.2 "Honeynet"

La idea de "honeypot" es desarrollada con el término "Honeynet" (Red trampa). Esta expresión fue adoptada por "The Honeynet Project"; una organización no lucrativa, fundada por Lance Spitzner. Este grupo está compuesto por expertos en seguridad, cuyo objetivo es aprender las herramientas, tácticas y motivos de los atacantes. [11]

Una "Honeynet" es una herramienta de investigación. Es un tipo de "honeypot" que consiste en una red diseñada para ser comprometida por intrusos. Sirve para estudiar las técnicas utilizados por los intrusos que la han comprometido.

Una "Honeynet" no es lo mismo que un sistema trampa tradicional. A continuación se describen las diferencias más significativas:

- Una "Honeynet" no es un sistema en solitario, sino una red. Esta red puede estar compuesta por distintos sistemas trampa, tales como Linux, Windows, Solaris, "routers", conmutadores, etc. El hecho de proporcionar un entorno de red aporta un ambiente más "creíble" desde el punto de vista del atacante. Un entorno de sistemas heterogéneo permite además, captar la atención de más intrusos, algunos de los cuales están especializados en atacar determinados sistemas operativos o servicios. Por otra parte, permite aprender un mayor y variado número de tácticas de ataque.
- Los sistemas utilizados en una "Honeynet" son sistemas de producción. Es decir, son sistemas reales, aunque no se utilicen con otro propósito que el de monitorizar su actividad. Ningún sistema o servicio es emulado. No se hace intento alguno de disminuir su seguridad. Normalmente se instalan los sistemas trampa más conocidos, con la configuración que traen por defecto, como Linux Red Hat, servidores Windows o servidores Solaris.

Las "Honeynets" son herramientas de seguridad con un punto de vista diferente al tradicional *defensivo*, presente en cortafuegos, encriptación o sistemas de detección de intrusiones. Son herramientas diseñadas básicamente para aprender y adquirir experiencia en el área de seguridad.

El Proyecto Honeynet ha definido dos tipos de arquitecturas básicas para sus "Honeynets": GenI y GenII. Ambas arquitecturas son descritas a continuación, seguidas de varios métodos para implementar "Honeynets" virtuales. [12]

4.2.2.1 GenI

Esta arquitectura simple fue la primera en desarrollarse, en 1999. Una red es situada detrás de un dispositivo de control de acceso, generalmente un cortafuegos, como se muestra a continuación.

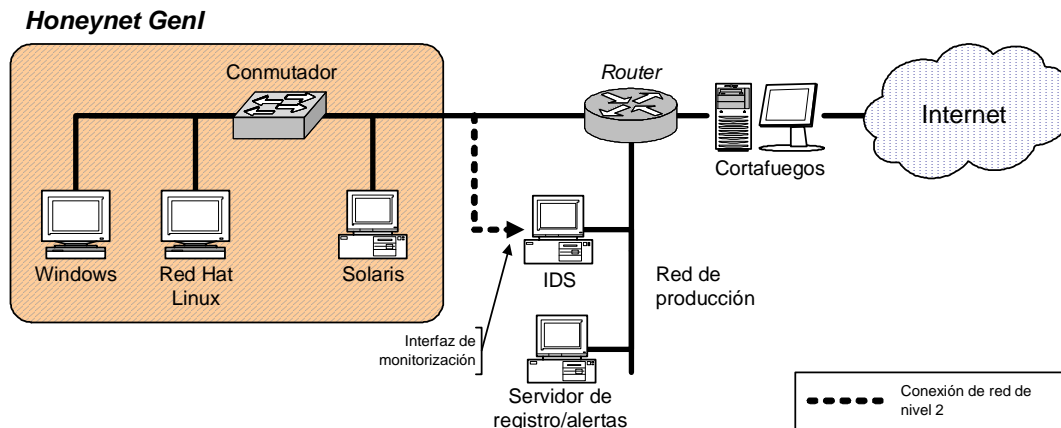


Figura 4-4 - Arquitectura "Honeynet" GenI

En el diagrama, se puede ver un cortafuegos de nivel tres separando la "Honeynet" en tres redes diferentes: la "Honeynet", Internet y la red de Producción. Cualquier paquete que entre o salga de la Honeynet tiene que pasar a través del cortafuegos y del "router". El cortafuegos filtra las conexiones entrantes y salientes. El "router" complementa este filtrado. El cortafuegos está diseñado para permitir cualquier conexión entrante, pero controla las conexiones salientes.

Este tipo de arquitectura es eficaz contra ataques automatizados, o contra atacantes de nivel básico. Pero no son de gran utilidad contra atacantes avanzados. El entorno proporcionado por las "Honeynets" GenI suele ser poco atractivo, consistiendo básicamente en instalaciones por defecto de sistemas operativos. Hay que destacar que este modelo apenas se implementa ya, siendo más utilizado su sucesor: GenII, comentado en el siguiente apartado.

4.2.2.2 GenII

Esta arquitectura de "Honeynets" fue desarrollada en 2002, y fue pensada para solventar muchos de los problemas existentes en el modelo anterior. Con respecto a las tecnologías GenI, esta arquitectura es más fácil de implementar, difícil de detectar y de mantenimiento más seguro.

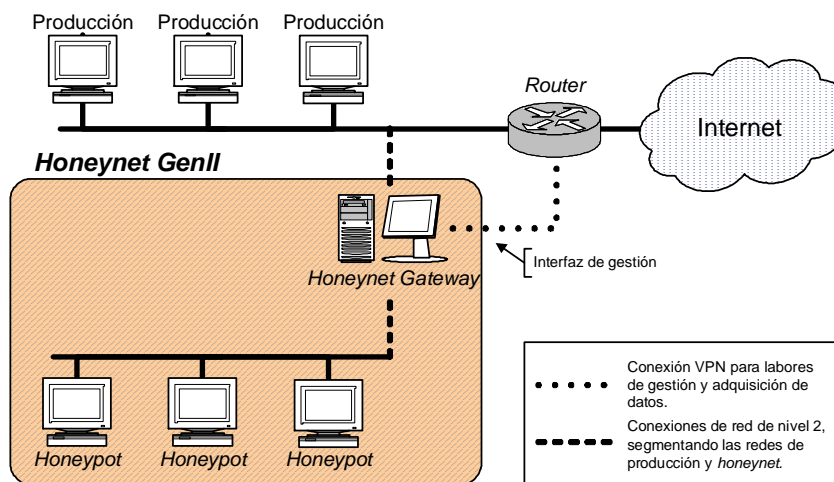


Figura 4-5 - Arquitectura "Honeynet" GenII

Como se puede ver en la Figura 4-5, la primera diferencia con respecto a la arquitectura GenI es que se utiliza un "Honeynet Gateway" (Puerta de enlace de la Red trampa) que combina los elementos de IDS y cortafuegos aparecidos por separado en el modelo GenI. Esto simplifica su gestión. La segunda diferencia radica en el propio "gateway", que trabaja a nivel 2, de forma similar a un puente. Este método, muy común en este tipo de mecanismos, permite prescindir de dirección IP, reduciendo las posibilidades de detección por parte de los atacantes.

Por otra parte, el "Gateway" no encamina paquetes. En vez de bloquear las conexiones de salida, se limita el ancho de banda del atacante, haciendo más realista y flexible el entorno.

Los sistemas trampas introducidos dentro de la "Honeynet" consisten normalmente en instalaciones básicas de los sistemas operativos más comunes, a veces con algunos servicios de red activados para hacerlos más atractivos como objetivo de ataque.

4.2.2.3 "Honeynet" virtual

La aparición de herramientas de emulación o soporte virtual, han hecho posible este modelo de implementación de "Honeynets". Este enfoque consiste en crear una "Honeynet" completa en un sólo equipo físico. Una "Honeynet" virtual no es una arquitectura, sino una forma de implementarlas: de esta manera, se puede utilizar para crear tanto arquitecturas tipo GenI como GenII.

Entre las opciones existentes para crear una "Honeynet" virtual destacan el producto comercial VMware [13], y User Mode Linux (UML), desarrollado por Jeff Dike [14]. Consiste en un módulo especial del núcleo de sistema que permite ejecutar muchas versiones virtuales de Linux en el mismo sistema simultáneamente. A continuación se describen brevemente las ventajas e inconvenientes de ambos productos:

- VMware es de pago y de código cerrado, mientras que UML es de libre distribución.
- VMware permite tres modos de instalación: "Workstation", GSX, o ESX. Cada uno con diferentes capacidades, según las necesidades del usuario.

- UML necesita significativamente menos recursos que VMware.
- VMware soporta más sistemas operativos que UML, el cual, está limitado a sistemas Linux; aunque se está desarrollando una aplicación para Windows.
- Una de las mejores características con las que cuenta VMware, es que tiene una consola de administración remota que presenta al sistema invitado como si se estuviera sentado delante, permitiendo su gestión remota sin generar tráfico de red. UML no posee interfaz gráfica, sino realiza la gestión a través de la línea de comandos.
- Al ser UML un producto de código abierto, no proporciona soporte oficial ni comercial.

Una "Honeynet" virtual puede ser *Auto-contenida* o *Híbrida*.

4.2.2.3.1 "Honeynet" virtual Auto-contenida

La "Honeynet" virtual Auto-contenida engloba una "Honeynet" en un sistema físico único.

- Ventajas:
 - Fácilmente transportable, especialmente si se instala en un portátil.
 - Rápida puesta en funcionamiento. Una vez instalada, sólo hay que conectarla a la red y configurarla en pocos minutos.
 - Es barata y ocupa poco espacio. Sólo hace falta un ordenador.
- Desventajas:
 - Si falla el hardware, la "Honeynet" entera podría dejar de funcionar.
 - Ordenador de altas prestaciones. Aunque sólo requiere un ordenador, tiene que tener suficiente memoria y procesador.
 - Seguridad. Como todos los sistemas comparten el mismo hardware, puede que un atacante acceda a otras partes del sistema. Mucho depende del software virtual.
 - Limitación por software. Como todo tiene que ejecutarse en una sola máquina, hay software que no se podrá utilizar por problemas de incompatibilidad. Por ejemplo, una IOS Cisco en un procesador Intel.

A continuación, la Figura 4-6 describe la arquitectura de una "Honeynet" virtual Auto-contenida.

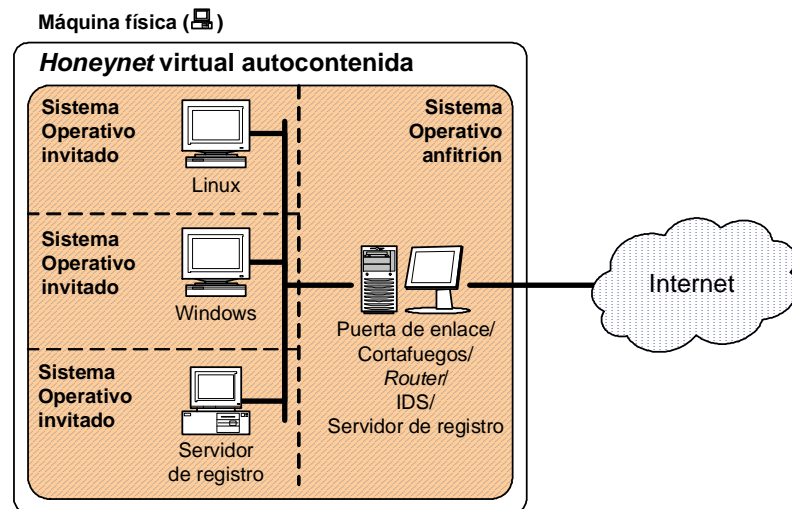


Figura 4-6 - "Honeynet" virtual auto-contenida

4.2.2.3.2 "Honeynet" virtual Híbrida

Una "Honeynet" virtual Híbrida es una combinación de una "Honeynet" y del software virtual. Es decir, los sensores de IDS y el almacenamiento de registros, están en un sistema separado y aislado, para reducir el riesgo de compromiso. Sin embargo, todos los "honeypots" son ejecutados virtualmente en una única máquina.

- Ventajas:
 - Seguridad. El único peligro sería que el atacante accediera a otros "honeypots".
 - Hay mayor flexibilidad a la hora de utilizar software para el control y captura de los datos de red.
- Desventajas:
 - Al implicar a más de una máquina, la movilidad es más reducida.
 - Es más cara y ocupa más espacio que la Auto-contenida.

La Figura 4-7 describe la arquitectura del modelo de "Honeynet" Híbrida.

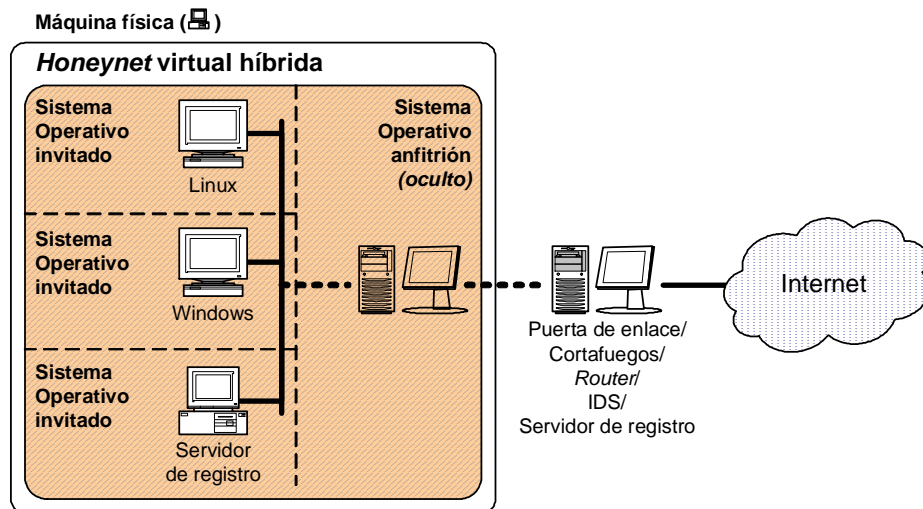


Figura 4-7 - "Honeynet" virtual híbrida

4.2.2.4 Ventajas e inconvenientes

Los sistemas comentados en este apartado permiten estudiar en detalle las tácticas, métodos y motivos de los intrusos, aspecto que los diferencia del resto de los productos y soluciones de seguridad. Su enfoque no está basado en la tradicional posición *defensiva* (cortafuegos, encriptación, etc.). Lejos de bloquear los ataques, su metodología se basa en el seguimiento en detalle de los procesos de intrusiones.

No obstante, hay que conocer los aspectos legales que puede suponer la implantación de uno de estos sistemas antes de decidir hacerlo.

4.2.2.4.1 Ventajas

- Ayudan a descubrir nuevos ataques, en ocasiones no publicados por las autoridades de seguridad. Esto permite mejorar los motores de detección de los sistemas de detección de intrusiones, así la creación de nuevos patrones de ataque.
- Los atacantes no dañan sistemas reales.
- Utilizar sistemas trampa similares a los de producción permite identificar fallos de seguridad existentes en el entorno real.
- Ayudan a perfeccionar los mecanismos de respuesta ante incidentes.
- Aportan mucha experiencia en el campo de la seguridad.

4.2.2.4.2 Inconvenientes

- Este tipo de sistemas siempre han presentado dudas en cuanto a su verdadera efectividad a la hora de mejorar la seguridad. No obstante, cada vez están recibiendo más aceptación entre los miembros de la comunidad de seguridad.

- Es necesario un alto nivel de conocimientos y experiencia en materia de redes y seguridad para poder instalar eficazmente un sistema de estas características.
- Las implicaciones legales que conlleva la instalación de uno de estos sistemas no está bien definida.

4.2.3 "Padded Cell"

Los sistemas basados en células de aislamiento ("padded cell") tienen una metodología que puede recordar a los sistemas trampa, pero no son exactamente lo mismo. Funcionan de forma conjunta con un dispositivo que cuenta con capacidades de enrutamiento y detección de intrusiones, que al detectar algún ataque, lo redirige hacia un "host" especial (denominado célula de aislamiento).

Al igual que los sistemas trampa, mencionados en apartados anteriores, una célula de aislamiento ofrece al atacante un entorno aparentemente idéntico a uno real. Sin embargo, al estar protegida del resto de la red, no causa daños. En muchas ocasiones estos "host" aislados consisten en espejos de sistemas de producción reales, para proporcionar un escenario más creíble. Como los sistemas trampa, las células de aislamiento pueden utilizarse para comprender mejor los métodos utilizados por los intrusos.

Este tipo de sistemas se lleva utilizando, junto con sistemas de detección de intrusiones, desde finales de la década de los 80.

Un producto enfocado para trabajar con células de aislamiento es el "Bait and Switch", desarrollado por J. Whitsitt y A. Gonzalez [15]. Esta herramienta, se instala en un sistema con tres interfaces de red y redirecciona el tráfico hostil hacia un sistema trampa especialmente diseñado para recibir ataques (la célula de aislamiento), que normalmente consiste en copia *parcial* (con datos ficticios) de un sistema real.

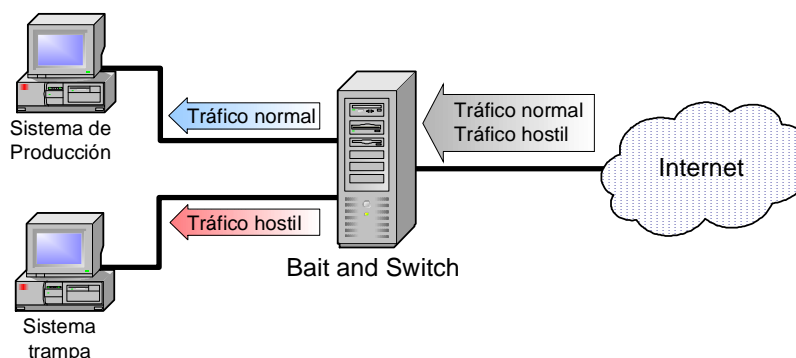


Figura 4-8 - Procedimiento general de "Bait and Switch"

Una vez redirigidas sus acciones, el atacante no se percató de que está atacando un sistema trampa en vez del real. Por otra parte, sus actividades pueden ser monitorizadas para ser estudiadas con el fin de mejorar la seguridad de la red.

"Bait and Switch" utiliza el sistema de detección de intrusiones Snort (para detectar los ataques) [16], así como `iproute2`¹ y `netfilter`².

4.2.3.1 Ventajas e inconvenientes

Dadas las similitudes entre estos sistemas y las "honeynets", la mayoría de las ventajas e inconvenientes descritos en el apartado anterior, pueden ser trasladados a este caso.

4.3 Verificador de integridad de ficheros

Un verificador de integridad de ficheros es una herramienta utilizada por sistemas de detección de intrusiones, normalmente los basados en máquina, para mejorar sus capacidades.

Aplican funciones resumen, u otros métodos de cifrado robustos, a ficheros críticos de sistema, comparando los resultados con una base de datos de referencia, y comunicando los posibles cambios o diferencias.

Hay varias razones por las que utilizar este tipo de herramientas para detectar intrusiones. Un atacante puede alterar o eliminar ficheros para no dejar evidencias de su actividad. Además, puede querer instalar algún troyano que le permita obtener el control de la máquina. O incluso puede dejar una puerta trasera que le deje volver a entrar en el sistema.

Los verificadores de integridad de ficheros no sólo pueden servir para detectar intrusiones a través de los cambios encontrados en ficheros. También son de gran ayuda durante un análisis forense, facilitando la identificación del ataque o método utilizado. Además, ayudan a devolver a la normalidad un sistema, optimizando el proceso de restauración.

Uno de los productos más conocidos entre los verificadores de integridad es Tripwire®. [17]

4.4 Cortafuegos: Prevención de Intrusiones

Un cortafuegos es un sistema diseñado para evitar el acceso no autorizado a una red privada. Pueden ser dispositivos hardware, software, o una combinación de ambos. Se suele utilizar para proteger el acceso a redes privadas desde otras redes, como por ejemplo Internet.

¹ `iproute2` es una herramienta que permite el manejo de interfaces de red bajo Linux.

² `netfilter/iptables` es un subsistema de cortafuegos para Linux que permite trabajar con tráfico de red, filtrando paquetes o realizando funciones de NAT (Traducción de Direcciones de Red). Está disponible en <http://www.netfilter.org/>.

Cuando se reúne la capacidad de bloqueo de un cortafuegos y la capacidad de análisis de un IDS en un sólo producto, se obtiene un Sistema de Prevención de Intrusiones ("Intrusion Prevention Systems") o IPS.

Los Sistemas de Detección de Intrusiones son sistemas de seguridad *reactivos*, más que *proactivos*. Es decir, esperan a que tenga lugar un ataque para emitir una alarma. Los IPSs, son dispositivos (de software o hardware) que detienen cualquier ataque antes de que pueda causar daños.

Los IPSs son considerados un caso especial de IDS porque ambos sistemas comparten la misma metodología básica. Como ya se comentó, los IPSs son IDS a los que se le ha añadido la capacidad de un cortafuegos (filtro de tráfico de red). De hecho, muchos expertos los consideran la siguiente generación de IDS.

Por otra parte, el comportamiento de un IPS es similar al de un IDS programado para responder ante ataques de forma activa, como ya se apuntó en el apartado 3.3.2.1 "Respuestas activas". Los sistemas de prevención descartan o bloquean los paquetes sospechosos tan pronto son identificados. Todos los paquetes pertenecientes a una misma sesión sospechosa pueden ser eliminados de la misma forma. Algunos IPSs también contemplan la posibilidad de detectar anomalías en el uso del protocolo, como paquetes manipulados intencionadamente.

Atendiendo a la fuente de datos que utilizan, los IPSs son de dos tipos: basados en máquina (HIPS) y basados en red (NIPS). Los HIPS utilizan agentes instalados directamente en la máquina a proteger, y están muy relacionados con el sistema operativo y sus servicios. Los NIPS, también conocidos como "Gateway IDS" (GIDS), son sistemas que contienen al menos dos interfaces de red (para monitorización interna y externa), e integran tanto características de cortafuegos como de IDS.

A continuación se describirán cinco modelos de IPS: IDS basado en red en modo "in-line", conmutador de nivel siete, cortafuegos basado en aplicación/IDS, conmutador híbrido y aplicación engañosa. [18]

4.4.1 IDS basado en red, en modo "in-line"

Una de las formas más comunes de implementar un NIDS (IDS basado en red) es utilizando dos dispositivos de red. Uno para interceptar el tráfico de red, y otro para efectuar las labores de gestión y administración, como en la Figura 4-9.

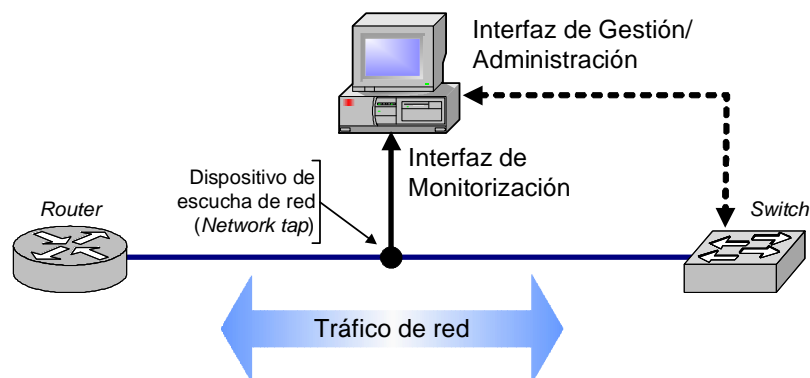


Figura 4-9 - NIDS en modo de escucha ("Tap mode")

La interfaz de red utilizada para la monitorización está conectada a un dispositivo de escucha ("network tap") que le permite *pinchar* el tráfico del segmento de red. Además, esta interfaz no suele tener asignada ninguna dirección IP, para disminuir en lo posible las posibilidades de ser detectado. Esto impide que algún elemento de red le envíe paquetes, o que el NIDS pueda responder a ellos.

En el modo "in-line" (en línea) el NIDS actúa a nivel 2, como un puente. Se sitúa entre la red que se desea proteger y el resto. Cuenta con una interfaz de red para recibir el tráfico del exterior, y otra para transmitirlo a la red a proteger. Además suele tener otra interfaz para las labores de administración y gestión.

Esta situación le permite el control total sobre el tráfico que pasa por su tramo de red. No sólo puede analizar todo el tráfico que recibe, antes de decidir qué hacer, sino que puede gestionar el ancho de banda. La siguiente figura describe este modo de funcionamiento.

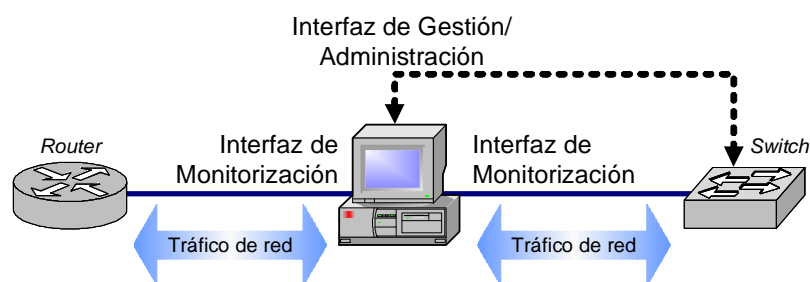


Figura 4-10 - NIDS en modo en línea ("In-line mode")

Uno de los sistemas que desarrollan esta idea es Hogwash; una herramienta que utiliza el motor de detección de Snort para anular los paquetes maliciosos antes de que lleguen a su objetivo [19]. Este IPS, además de contar con las labores de detección y bloqueo normales, tiene la opción de reescribir el tráfico de red. Así, si un atacante envía una petición maliciosa, Hogwash puede modificarla antes de dejarla pasar.

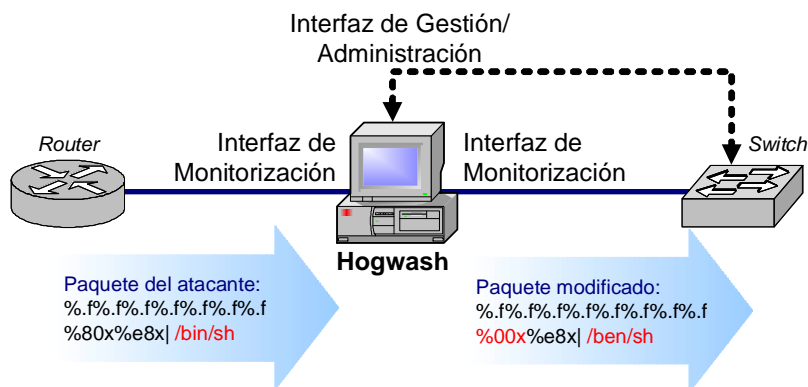


Figura 4-11 - Procedimiento general de "Hogwash"

Un NIDS instalado en modo "in-line", y con una serie de modificaciones para soportar el tratamiento de tráfico de red, ofrece las posibilidades de un NIDS normal, con la capacidad añadida de un cortafuegos.

La detección de ataques dependen directamente de los métodos utilizados por el NIDS. Desgraciadamente, la mayoría de los productos comerciales existentes están basados en la detección de usos indebidos, por lo que las posibilidades de reconocer nuevos ataques es limitada. Algunos productos intentan solventar esta carencia incorporando funciones de detección de anomalías de protocolo.

Aparte de Hogwash, mencionado antes, otros IPSs similares son "IntruShield", de IntruVert Networks [20], "ISS Guard", de Internet Security Systems[21], "UnityOne™", de NetScreen Technologies[22], o algunos productos de TippingPoint Technologies[23].

4.4.2 Conmutador de nivel siete

Un conmutador ("switch") ha sido tradicionalmente un dispositivo de nivel dos. La creciente necesidad de trabajar con grandes anchos de banda ha hecho que vayan ganando popularidad los conmutadores de nivel siete.

Estos dispositivos se suelen utilizar para balancear la carga de una aplicación entre varios servidores. Para ello, examinan la información de aplicación (por ejemplo HTTP, FTP, DNS, etc.) para tomar decisiones de encaminamiento.

Algunos fabricantes han empezado a añadir capacidades a estos conmutadores, para proporcionar protección ante ataques como DoS (Denegación de Servicio) o DDoS (DoS Distribuida).

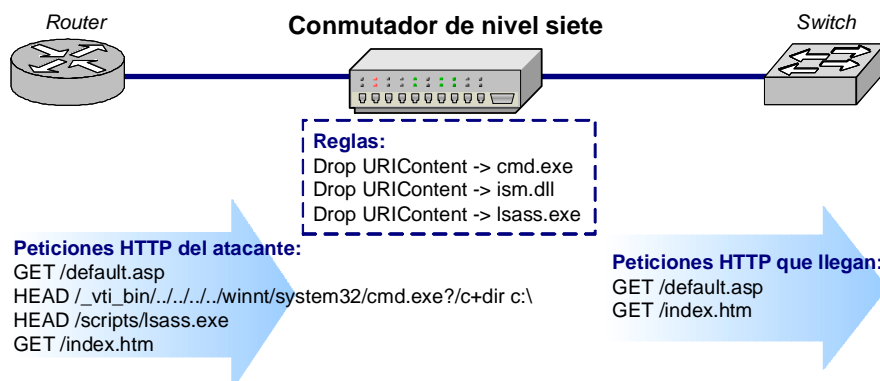


Figura 4-12 - Procedimiento general de un conmutador de nivel siete

Los conmutadores de nivel siete pueden trabajar fácilmente con redes de alta velocidad. Su método de detección es similar al utilizado por los NIDS, comentados anteriormente, por lo que presentan desventajas similares. Admiten la adición de nuevos patrones de ataque, como los NIDS. No obstante, los ataques que mejor reconocen y bloquean, frente a otras soluciones de seguridad, son los basados en DoS.

Otra de las ventajas de estos dispositivos, que no se encuentran en otros IPSs, es que admiten redundancia. Pueden ser implementados en modo "hot standby"¹ (espera en caliente) o en modo de balance de carga².

Algunas de las empresas que proporcionan este tipo de productos son Radware [24], Top Layer [25], o Foundry Networks [26].

4.4.3 Cortafuegos/IDS de aplicación

Los cortafuegos/IDS de aplicación trabajan en el nivel siete del modelo OSI, y se instalan en el sistema a proteger.

Estos IPSs poseen un alto grado de configuración para cada aplicación protegida. No analizan tráfico de red, como otros IPSs, sino elementos tales como la gestión de memoria, llamadas a sistema o intentos de conexión de la aplicación. Este enfoque permite reducir problemas de seguridad asociados a una programación deficiente, así como detectar ataques propios de este nivel como los de desbordamiento de búfer.

¹ Un dispositivo secundario es configurado para que se active en caso de que el primario falle.

² Distribución de tráfico entre dos o más dispositivos, utilizando preferencias descritas por políticas particulares.

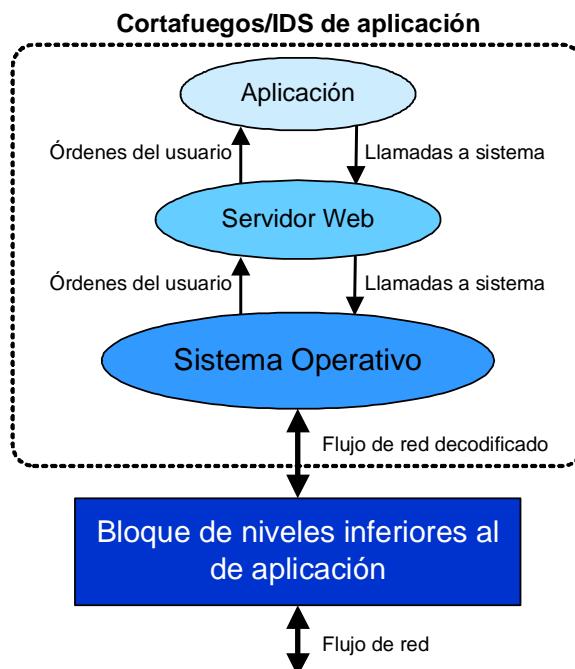


Figura 4-13 - Cortafuegos/IDS de aplicación

Este tipo de sistemas necesitan realizar una fase de creación de perfil de sistema, similar a la fase de entrenamiento acometida por los analizadores de detección de anomalías, descrita en el capítulo anterior. En ella, se procede a registrar la actividad de la aplicación para elaborar un modelo de comportamiento que sirva para detectar posibles intrusiones, junto con una serie de políticas.

Una vez en funcionamiento, si el IPS identifica alguna acción que no haya sido definida durante la creación del perfil, la bloquea por defecto. Este método, aunque cómodo de configurar, presenta evidentes problemas si no se definen todos los comportamientos normales de la aplicación. Por otra parte, si se actualiza la aplicación, es posible que se necesite repetir el proceso de creación de perfil.

No obstante, este modelo de IPS es de los más apropiados para proteger aplicaciones. De los comentados aquí, es el único que monitoriza la actividad y relación de la aplicación con el sistema operativo. Además, están instalados en cada máquina física a proteger, por lo que ofrecen un alto nivel de personalización.

Ejemplos de empresas que producen IPS basados en cortafuegos/IDS de aplicación son OKENA, con su "Storm watch" [27], o Enterecept [28].

4.4.4 Conmutador híbrido

Este tipo de dispositivos son una combinación de los dos anteriores: los conmutadores de nivel siete y cortafuegos/IDS de aplicación.

Son dispositivos hardware instalados de la misma forma que los conmutadores de nivel siete, pero no utilizan conjuntos de reglas como los NIDS, sino un método de detección basado en políticas similar al de los cortafuegos/IDS de aplicación. Analizan el tráfico de red en busca de información definida en las políticas aplicadas.

Una de las ventajas de estos productos es que se pueden configurar importando los resultados de un analizador de vulnerabilidades, comentado en apartados anteriores, utilizado contra el sistema a proteger. Esto permite implementar de forma rápida y efectiva este tipo de sistemas.

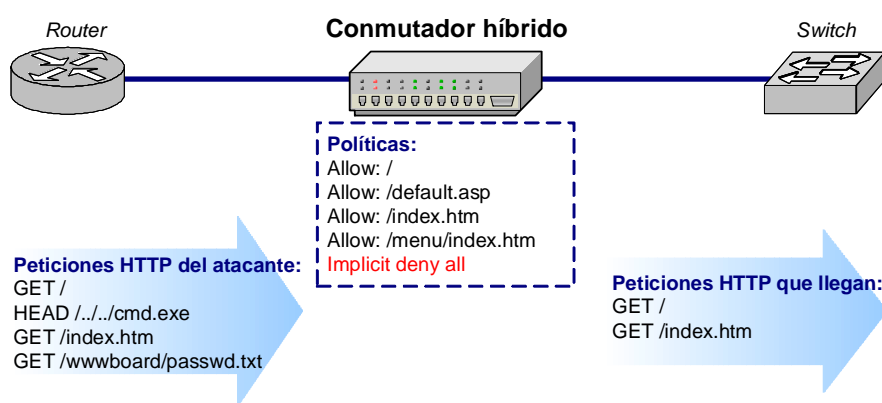


Figura 4-14 - Procedimiento general de un conmutador híbrido

Un conmutador híbrido tiene conocimientos sobre el servidor que protege (servidor FTP, Web, SMTP etc.), como un conmutador de nivel siete, pero también de las aplicaciones que hay sobre él. Además, al igual que los cortafuegos/IDS de aplicación, también bloquea todas las acciones no definidas como permitidas.

Si este tipo de IPS tiene demasiada carga de trabajo, puede ser utilizado en conjunción con un conmutador de nivel siete, que le redirija sólo aquellas peticiones que considere maliciosas.

Dos empresas que diseñan productos basados en esta tecnología son "F5 Networks" [29] y "KaVaDo, Inc." [30].

4.4.5 Aplicación engañosa

El particular enfoque de los IPSs basados en aplicación engañosa ("deceptive application"), comprende dos fases. La primera consiste en la monitorización del tráfico de red para crear un modelo de actividad normal, similar a la fase de creación de perfil de los cortafuegos/IDS de aplicación. Durante la segunda fase, si el IPS observa algún intento de conexión a algún servicio que no existe, devuelve una respuesta falsa hacia el atacante.

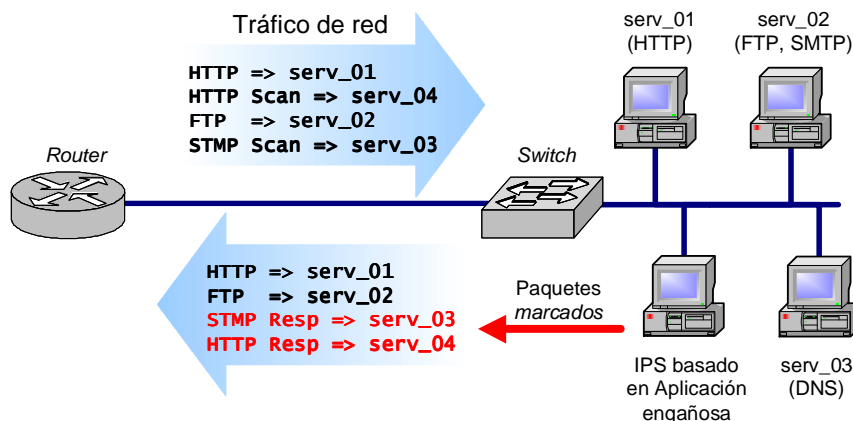


Figura 4-15 - IPS basado en aplicación engañosa ("deceptive application")

La respuesta está *marcada* por el IPS, de tal manera que cuando el posible atacante decida volver a intentar alguna conexión, el IPS reconocerá su *marca* y lo bloqueará. Como se puede observar, el ataque es detectado y anulado antes de que ocurra.

También se pueden introducir marcas en el campo de datos de los paquetes, permitiendo la detección de ataques contra servicios que existen.

Uno de los inconvenientes de este sistema es que el intruso identifique el método utilizado por el IPS para marcar los paquetes. Esto le permitiría *desmarcar* los paquetes antes de ejecutar su ataque, sorteando la protección.

El producto "ActiveScout", de la empresa ForeScout es un IPS que utiliza técnicas de aplicación engañosa. [31]

4.5 Referencias

- [1] Shostack, Adam and Scott Blake. *Towards a Taxonomy of Network Security Assessment Techniques*. Proceedings of 1999 Black Hat Briefings, Las Vegas, NV, July 1999.
- [2] Bace, Rebecca and Peter Mell. *Intrusion Detection Systems*. [en línea] [consultado en marzo, 2003]. Disponible en <<http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf>>.
- [3] Cerberus Information Security, Ltd. *Cerberus' Internet Scanner - CIS*. [en línea] 2000 [consultado en abril, 2003]. Disponible en <<http://www.cerberus-infosec.co.uk/cis.shtml>>.
- [4] Deraison, Renaud. *The Nessus Project*. [en línea] 1998- 2003 [consultado en abril, 2003]. Disponible en <<http://www.nessus.org>>.
- [5] Venema, Wietse y Dan Farmer. *SATAN*. [en línea] 1995 [consultado en abril, 2003] Disponible en <<http://www.fish.com/satan/>>.
- [6] Spitzner, Lance. *Honeypots: Definition and Value of Honeypots*. [en línea] Última modificación, 29 de mayo, 2003 [consultado en junio, 2003]. Disponible en <<http://www.spitzner.net/honeypots.html>>

- [7] The Honeynet Project. *Know your enemy*. Addison-Wesley, septiembre 2001. cap. 2 *What a Honeynet Is*.
- [8] Stoll, Clifford. *The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage*. New York, NY: Doubleday, 1989.
- [9] Cheswick, Bill. *An Evening with Berferd In Which a Cracker is Lured, Endured and Studied*. [en línea] 1991 [consultado en abril, 2003]. Disponible desde Internet en <<http://www.securityfocus.com/data/library/berferd.ps>>.
- [10] Spitzner, Lance. *The Value of Honeypots*. [en línea] 10 de octubre, 2001 [consultado en abril, 2003]. Disponible en <<http://www.securityfocus.com/infocus/1492>>
- [11] The Honeynet Project. [en línea] abril, 1999 [consultado en abril, 2003]. Disponible en <<http://www.honeynet.org>>.
- [12] Spitzner, Lance. *Honeypots, tracking Hackers*. Addison-Wesley, septiembre 2002. cap. 11 *Honeynets*, p. 242-262.
- [13] VMware, Inc. *VMware*. [en línea] 1998-2993 [consultado en abril, 2003]. <<http://www.vmware.com/>>.
- [14] Dike, Jeff. *User-mode Linux Kernel*. [en línea] actualizado con frecuencia [consultado en abril, 2003]. Disponible en <<http://user-mode-linux.sourceforge.net/>>.
- [15] Whitsitt Jr, John y Alberto Gonzalez. *The Bait and Switch Honeypot*. [en línea] 2002 [consultado en abril, 2003]. Disponible en <<http://baitswitch.sourceforge.net>>.
- [16] Roesch, Marty et al. *Snort.org*. [en línea]. Actualizado semanalmente [consultado en marzo de 2003]. Disponible en <<http://www.snort.org>>.
- [17] Kim Gene H. and E. H. Spafford. *Tripwire: A Case Study in Integrity Monitoring*. Internet Beseiged: Countering Cyberspace Scofflaws; edited by Dorothy and Peter Denning, Addison-Wesley, 1997.
- [18] Desai, Neil. *Intrusion Prevention Systems: the Next Step in the Evolution of IDS*. [en línea] 27 de febrero, 2003 [consultado en abril, 2003]. Disponible en <<http://www.securityfocus.com/infocus/1670>>
- [19] Larsen, Jason. *Hogwash*. [en línea] junio, 2001 [consultado en abril, 2003]. Disponible en <<http://hogwash.sourceforge.net/>>.
- [20] IntruVert. *IntruShield*. [en línea] 2002 [consultado en abril, 2003]. Disponible en <<http://www.intruvert.com/products/index.htm>>.
- [21] Internet Security Systems. *ISS Guard*. [en línea] 1994 [consultado en abril, 2003]. Disponible en <http://www.iss.net/products_services/enterprise_protection/rsnetwork/guard.php>.
- [22] NetScreen Technologies, Inc. [en línea] 1998 [consultado en abril, 2003]. Disponible en <<http://www.netscreen.com/products/idp/>>.
- [23] TippingPoint Technologies. *UnityOne™*. [en línea] 1998 [consultado en abril, 2003]. Disponible en <<http://www.tippingpoint.com/products/index.html>>.
- [24] Radware. [en línea] 1997 [consultado en abril, 2003]. Disponible en <<http://www.radware.com/>>
- [25] Top Layer. [en línea] 1997 [consultado en abril, 2003]. Disponible en <<http://www.toplayer.com/>>.
- [26] Foundry Networks. [en línea] 1999 [consultado en abril, 2003]. Disponible en <<http://www.foundrynet.com/>>.
- [27] OKENA. *Storm watch*. [en línea] 1999 [consultado en abril, 2003]. Disponible en <<http://www.okena.com/>>.

- [28] Entercept. *Entercept*. [en línea] 2003 [consultado en abril, 2003]. Disponible en <<http://www.entercept.com/>>.
- [29] F5 Networks. *Sanctum Appshield*. [en línea] 2003 [consultado en abril, 2003]. Disponible en <http://www.f5.com/solutions/applications/Firewalls/sanctum_sb.html>.
- [30] KaVaDo, Inc. *InterDo* [en línea] 2001[consultado en abril, 2003]. Disponible en <<http://www.kavado.com/>>.
- [31] Fore Scout. *ActiveScout*. [en línea] abril, 2000 [consultado en abril, 2003]. Disponible en <<http://www.forescout.com/>>.

Capacidades y limitaciones

*"La invencibilidad está en uno mismo,
la vulnerabilidad en el adversario"
El arte de la guerra, Sun Tzu*

La falta de conocimiento en torno a los sistemas de detección de intrusiones provoca a veces que sean malinterpretados. Esto crea ciertas confusiones con respecto a lo que realmente pueden hacer y lo que no. A pesar de que son de gran ayuda en materia de seguridad, no son la solución definitiva.

Por esta y otras razones, se resumirán a continuación algunos de los aspectos más relevantes de estos sistemas.

5.1 Capacidades

Estas son algunas de las funciones y ventajas asociadas a los IDSs [1]:

- **Monitorización:** Cuentan con métodos para monitorizar y analizar tanto los eventos de sistema, como el comportamiento de los usuarios.
- **Claridad:** Extraen los datos más relevantes de entre grandes cantidades de datos de eventos de auditoría, lo que facilita el trabajo del auditor de seguridad. Para ello suelen utilizar diversos métodos, como la reducción de auditoría, o técnicas de filtrado estadísticas.
- **Registro:** La mayoría de los IDSs proporciona no sólo métodos para registrar su propia actividad, sino que permiten emitir informes sobre los eventos más importantes ocurridos en un determinado período de tiempo.
- **Comprobación continua:** Crean un modelo sobre el estado del sistema, y comparar los cambios posteriores con respecto a ese modelo. En este aspecto se incluyen los casos en que se utilizan algoritmos de cifrado para determinar si ha habido cambios en el sistema de ficheros (verificadores de integridad).
- **Correlación:** Aunque de forma limitada, pueden establecer patrones de relación entre ataques o comportamientos similares, mostrados desde distintas máquinas para así determinar por ejemplo, si el atacante es la misma persona, o si se trata de un ataque coordinado.
- **Ataques conocidos:** Los IDSs basados en la detección de usos indebidos, pueden reconocer ataques que coincidan con los patrones que almacenan en su base de conocimiento.

- **Ataques no conocidos:** Normalmente utilizan técnicas estadísticas para elaborar patrones de actividad y contrastarlos con la actividad normal, detectando posibles anomalías. Aunque poco desarrollado en la práctica, este enfoque tiene ilimitadas posibilidades. Los métodos basados en redes neuronales, algoritmos genéticos, minería de datos o los relacionados con el sistema inmune biológico, son tan sólo algunos de los utilizados en la detección de anomalías. Todos ellos han dado resultados satisfactorios.
- **Fallos de seguridad:** Los analizadores de vulnerabilidades, considerados como un caso especial de IDS (enfoque estático), permiten comprobar la seguridad de la configuración de un sistema. En ocasiones esto se hace lanzando ataques conocidos contra el objetivo, para evaluar sus reacciones. Otra de las formas de descubrir vulnerabilidades consiste en repasar automáticamente la configuración del sistema, en busca de debilidades en las políticas de seguridad.
- **Tiempo real:** La mayoría de los productos de detección de intrusiones actuales utilizan mecanismos de análisis y registro en tiempo real.
- **Alarmas:** Comunican alarmas a los responsables cuando se produce una situación anormal, como una intrusión. Las opciones para hacer esto son bastante variadas, pudiéndose por ejemplo, registrar un evento de sistema, o enviar una notificación vía correo electrónico o mensajes SMS ("Short Message Service").
- **Sencillez de uso:** Las características de detección automática así como contar con una interfaz fácil de usar, hace que muchos IDSs permitan incluso a usuarios no expertos en seguridad, mejorar de forma sensible la seguridad de sus sistemas.
- **Seguridad básica:** Proporcionan información sobre las políticas de seguridad por defecto, así como métodos para corregir los posibles errores de configuración de forma automática.
- **Actualización:** La mayoría de los productos de detección de intrusiones basados en patrones de ataques contemplan la posibilidad de actualizar con frecuencia sus bases de conocimiento. Muchos de ellos permiten programar este proceso, que suele realizarse periódicamente mediante comunicación cifrada.

5.2 Limitaciones

Estas son algunas de las cosas que los IDSs no hacen, además de varios de sus inconvenientes:

- **Solución definitiva:** Los problemas de seguridad pueden originarse por múltiples motivos. No existe ninguna solución única que los resuelva todos. Los sistemas de detección de intrusiones no son una excepción. No obstante, aportan una serie de características únicas que los convierten en herramientas de gran ayuda en muchos entornos.
- **Falsos positivos:** Uno de los inconvenientes más populares entre la detección de intrusiones es el de las falsas alarmas; falsos positivos y falsos negativos. Los falsos positivos consisten en aquellas alarmas que tienen lugar cuando en realidad no se está

produciendo ninguna intrusión. Existen códigos, algunas de cuyas partes coinciden con patrones de ataque de desbordamiento de búfer, que son detectados como intrusiones, cuando en realidad no lo son. Por otra parte, los detectores de anomalías pueden reconocer como hostil la aparición de un nuevo tipo de tráfico, provocado por la reciente instalación de un nuevo servicio, cuando en realidad la situación es perfectamente normal. Lo más negativo de esta cuestión es que la continua aparición de falsos positivos puede hacer que un administrador acabe ignorando las alarmas, que es igual de negativo que no recibirlas.

- **Falsos negativos:** Son uno de los tipos de falsas alarmas, y se producen cuando no se emite el correspondiente aviso cuando sucede realmente un ataque o intrusión. Este tipo de situaciones, por razones obvias, también representa un problema. Cuando un atacante utiliza una técnica nueva, un ataque modificado basado en alguno ya existente, un ataque especializado contra este tipo de sistemas, o cuando un detector de anomalías es "entrenado" de forma progresiva por un intruso, para que interprete una acción hostil como normal, son sólo algunos ejemplos en los que pueden ocurrir falsos positivos.
- **Recursos:** El proceso de registro de datos y análisis (especialmente en tiempo real) hace que los sistemas de detección tengan importantes requerimientos de recursos de sistema, como tiempo de proceso o espacio de almacenamiento en bases de datos. Esto se hace especialmente necesario durante la monitorización de redes de alta velocidad.
- **Autosuficiencia:** No pueden compensar las debilidades o ausencia de otros sistemas de seguridad de la infraestructura, como contraseñas de baja calidad, cortafuegos, antivirus, etc.
- **Sobrecarga:** No pueden detectar, analizar y enviar alarmas frente a ataques de forma instantánea cuando hay demasiada carga de trabajo (excesivo tráfico de red, actividad de sistema) [2]. Estos sistemas llegan a descartar paquetes de red o segmentos de información de actividad de sistema, cuando la situación de sobrecarga es crítica.
- **Defensa ante nuevos ataques:** En la mayoría de los casos, no pueden detectar ataques de reciente aparición, o variantes de ataques existentes. Esto ocurre con mayoría de productos comerciales, que suelen utilizar detección de usos indebidos, basada en reglas o patrones de ataques. La detección de anomalías, dada la naturaleza de este tipo de análisis, permite ampliar el rango de detección de este tipo de ataques, pero no los reconoce todos.
- **Defensa ante ataques sofisticados:** Como ya se ha comentado, estos sistemas son de gran ayuda a la hora de simplificar las tareas de auditoría de seguridad. Pueden detectar con cierta eficacia ataques comunes, o de relativa simplicidad. Filtran grandes cantidades de información, destacando datos que pueden estar relacionados con posibles intrusiones. Sin embargo, no deben ser sobrevalorados. Aún no están preparados para identificar ataques demasiado sofisticados, realizados por atacantes expertos, que en algunas ocasiones utilizan técnicas de fragmentación de paquetes o incluso protocolos propios. En ese aspecto, sigue siendo necesaria la intervención humana.

- **Defensa ante ataques directos:** Al igual que ocurre con otros productos, como antivirus, o cortafuegos, no son capaces de bloquear ataques diseñados para evitar o inutilizar específicamente estos sistemas. Estas acciones son siempre realizadas por atacantes con amplios conocimientos sobre este tipo de sistemas.
- **Investigación automática:** Realizan tareas de análisis, y envían alarmas en caso de reconocer intrusiones o acciones hostiles. No obstante, la labor de investigación de cada ataque realizado la debe realizar un humano. Este tipo de acciones conlleva ciertas responsabilidades y habilidades de las que carecen estos sistemas.
- **Conocimiento de cada situación:** Estos sistemas no conocen de antemano las particularidades de cada entorno en que son implementados. Es el responsable de seguridad quien debe configurarlos, y adaptarlos a cada situación progresivamente.
- **Calidad de los datos:** No pueden compensar errores producidos por el uso de datos de mala calidad. Hay ataques que consisten en saturar a los IDSs con información redundante, o simplemente ruido. Cada fuente de datos adicional incrementa las posibilidades de obtener información contaminada por un atacante. Trabajar con datos carentes de utilidad, invalida los resultados obtenidos.
- **Calidad de los protocolos:** No compensan las debilidades asociadas al diseño de un protocolo. Por ejemplo, TCP/IP y muchos otros protocolos no fueron creados para realizar mecanismos robustos de autenticación. Cuando alguien realiza un ataque, la dirección origen de los paquetes involucrados no tiene por qué ser necesariamente la del atacante. Esto dificulta la identificación y persecución de los culpables mediante procesos legales y judiciales.
- **Entornos conmutados:** Los detectores de intrusiones basados en red no trabajan bien en entornos de red que utilicen conmutadores ("switched environments"). Estos dispositivos sólo les envían el tráfico de red que va destinado a ellos mismos, dificultando las tareas de monitorización de tráfico de red global.
- **Encriptación:** El uso de comunicaciones cifradas (como SSH, SSL, IPSec, etc.) puede inhabilitar la utilidad de un detector de intrusiones basado en red, ya que no puede interpretar lo que está monitorizando. Aún en el caso de que pudiera interpretar lo que percibe, tener que descifrar los datos supondría una carga adicional de proceso. Esto no sólo incrementaría sus requerimientos de recursos, sino que podría hacer esta labor casi imposible en entornos con grandes cargas de tráfico. Para evitar este problema, los detectores se suelen instalar en los puntos extremos de la comunicación, para examinar los datos descifrados por las máquinas ("hosts").
- **IPv6:** Muchos detectores de intrusiones comerciales son incapaces de interpretar el protocolo IPv6, sucesor del ampliamente utilizado en Internet: IPv4. El protocolo IPv6 no está siendo adoptado por igual en todo el mundo, teniendo mayor acogida en los países asiáticos. Sin embargo, incluso en entornos en los que se trabaja únicamente con IPv4, el protocolo IPv6 permite crear túneles sobre IPv4. Esto impide a los detectores de intrusiones reconocer aquellos ataques que lo utilizan. Esta situación se puede corregir añadiendo capacidades de análisis de este protocolo a los motores de detección.

5.3 Referencias

- [1] Bace, Rebecca and Peter Mell. *Intrusion Detection Systems*. [en línea] [consultado en marzo, 2003]. Disponible en <<http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf>>.
- [2] Spitzner, Lance. *The Value of Honeypots*. [en línea] 10 de octubre, 2001[consultado en abril 2003]. Disponible en <<http://www.securityfocus.com/infocus/1492>>

Capítulo 6

Implementación

*"- Buenos... Días... Doctor... Chandra... Aquí... Hal... Estoy...
Listo... Para... Mi... Primera... Lección... De... Hoy..."
2001 Una odisea espacial, Arthur C. Clarke*

El nivel de seguridad y la gestión ofrecida por los Sistemas de Detección de Intrusiones, los hace casi imprescindibles en muchas de las organizaciones que cuentan con importantes infraestructuras de red.

Como ya se apuntó en el capítulo anterior que estos sistemas, aunque tienen muchas ventajas, no son perfectos. Por otra parte, muchos administradores carecen de los conocimientos adecuados para aprovechar al máximo sus características. Esto hace que la implementación de estos sistemas requiera un proceso previo de planificación, preparación, fase de pruebas, y formación especializada. Cada solución debe adaptarse a las necesidades particulares de cada caso, analizando entre otras características, las políticas de la organización, el nivel de desarrollo y los recursos de red.

Los requisitos de sistema de los IDSs dependen de los objetivos y recursos de cada organización. Por lo general, suele ser recomendable alcanzar una solución basada en el uso conjunto de IDS de red (NIDS) y de "host" (HIDS).

Por otra parte, si se acomete la implementación de forma escalonada, añadiendo de forma progresiva cada IDS, los propios administradores irán adquiriendo más experiencia con cada adición. Normalmente se empieza con la instalación de IDS basados en red, más sencillos de instalar y gestionar. Posteriormente, se instalan IDS basados en "host" en las máquinas críticas.

Una planificación de estas características debe completarse con el uso regular de analizadores de vulnerabilidades sobre los IDSs y otros elementos de seguridad. De esta manera, se ayuda a mantener la estabilidad y confianza de estos mecanismos.

En caso de implementar sistemas o redes trampa, es necesario contar con personal especializado en temas de seguridad y redes, y ubicar dichos sistemas en entornos altamente protegidos. Además, hay que asesorarse previamente en temas legales relacionados con estos elementos.

A continuación se describirán las características más comunes del establecimiento de detectores de intrusiones basados en red y en máquina. [1]

6.1 Sistemas de Detección de Intrusiones de red

Los agentes de un IDS basado en red monitorizan el tráfico de red para enviar los datos al motor de análisis. Estos elementos de monitorización pueden colocarse en distintos puntos de la arquitectura.

Uno de los objetivos de los agentes es el de no ser reconocido por atacantes, así como no interferir en el rendimiento de la red. Para ello, se suelen conectar al medio utilizando dispositivos de escucha. La interfaz de red dedicada a la monitorización se configura de forma que no tenga dirección IP. En algunas ocasiones, estos dispositivos se conectan a la red mediante un cable de sólo recepción o un "network tap" (dispositivo de escucha de red).

El abanico de productos de detección de intrusiones basados en red es muy extenso. Entre ellos destaca por su popularidad Snort, un potente detector de intrusiones de red, escrito en código abierto, basado en reglas, que incorpora algunas funciones de detección de anomalías [2]. Algunos ejemplos más de productos de este tipo son: Bro, escrito por Vern Paxson [3]; Firestorm, de John Leach y Gianni Tedesco [4]; Tiny Personal Firewall (cortafuegos personal con capacidades de IDS), de Tiny Software Inc. [5]; o el IDS híbrido (basado en máquina y en red) Prelude, distribuido bajo licencia GPL y desarrollado por Yoann Vandoorselaere [6].

A continuación se describirán las localizaciones más comunes en las que se puede implementar un NIDS. Cada una tiene sus propias ventajas e inconvenientes.

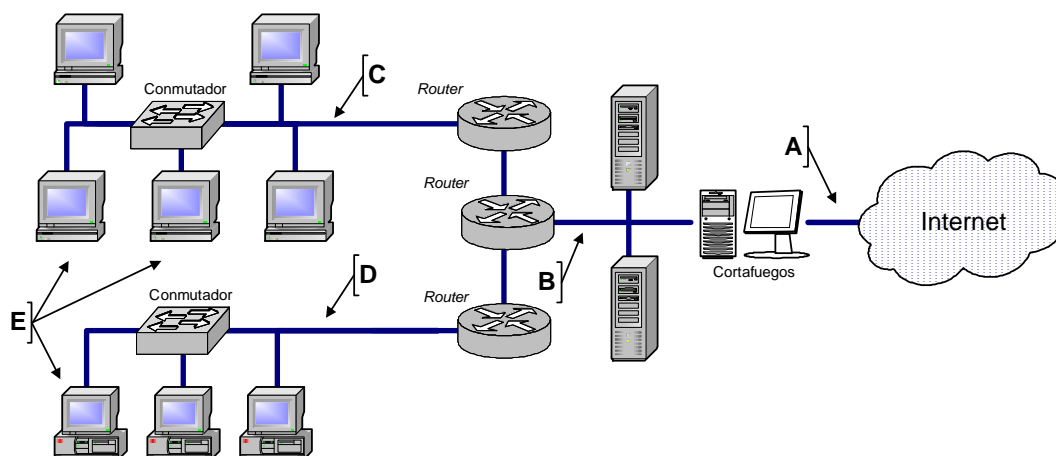


Figura 6-1 - Situaciones de implementación de un IDS

6.1.1 Delante del cortafuegos externo

Colocar los agentes delante del cortafuegos externo (A), permite:

- Monitorizar el número y tipo de ataques dirigidos contra la infraestructura de la organización.
- Detectar ataques cuyo objetivo es el cortafuegos principal.

Por otra parte, esta posición también presenta algunos inconvenientes:

- No permite detectar ataques que utilicen en sus comunicaciones algún método para ocultar la información, como algoritmos de encriptación, o *esteganografía*¹.
- En esta localización suele haber gran cantidad de tráfico de red. Un detector de intrusiones mal diseñado puede saturarse, descartando parte de la información que percibe, si no ha sido bien diseñado.
- Una situación como esta no ofrece ninguna protección. El NIDS puede convertirse en un blanco fácil si algún atacante logra identificarlo.

6.1.2 Detrás del cortafuegos externo

Esta localización (B), situada entre Internet y la red interna, se denomina DMZ (Zona Desmilitarizada). Se utiliza para proporcionar servicios públicos sin tener que permitir acceso a la red privada de la organización.

En esta subred se suelen ubicar los servicios principales de la infraestructura (servidores HTTP, FTP, SMTP, DNS, etc.). Normalmente está protegida por cortafuegos y otros elementos de seguridad.

Algunas de las ventajas de este caso son:

- Se monitorizan intrusiones que logran atravesar el cortafuegos principal.
- Se pueden detectar ataques dirigidos contra los servidores que ofrecen servicios públicos situados en esta subred.
- En caso de no detectar ataques con éxito, pueden reconocer algunas consecuencias de los mismos, como intentos de conexiones salientes, realizadas desde los servidores comprometidos.
- La identificación de los ataques y escaneos más comunes permite mejorar la configuración del cortafuegos principal.

A continuación se enumeran algunos de las desventajas de esta localización:

- Al igual que en el caso anterior, no permite identificar ataques que utilicen métodos para ocultar la información contenida en sus comunicaciones, como algoritmos de encriptación.
- La cantidad de tráfico existente normalmente en este segmento de red, puede hacer que el NIDS no pueda analizarla toda, descartando datos. Es importante diseñar un sistema capaz de responder ante situaciones críticas.

¹ Arte o ciencia de comunicar de manera oculta un mensaje, camuflando la información entre otro conjunto de datos para que pase desapercibida. Usualmente un texto, imagen, o archivo multimedia. Proviene de las palabras griegas *steganós* (cubierto) y *grapto* (escrito).

- La seguridad del NIDS mejora con la inclusión del cortafuegos que lo separa de la red exterior. Sin embargo, esto no excluye de tomar medidas adicionales para evitar que pueda ser comprometido por atacantes.

6.1.3 Redes principales

Cuando se monitoriza el tráfico de red en las redes con mayor actividad (C) se obtienen estas ventajas:

- Al haber más cantidad de tráfico, hay también más posibilidades de encontrar posibles ataques. Este hecho se cumple siempre que la cantidad de tráfico no supere la capacidad del NIDS.
- Se pueden detectar ataques producidos desde dentro de la propia red, como los realizados por personal interno.

Las desventajas relacionadas con esta posición son, entre otras:

- Al igual que en los casos anteriores, esta localización no permite detectar ataques que utilicen algoritmos de encriptación en sus comunicaciones.
- No pueden evitar problemas asociados al uso de conmutadores en la red. Las características de estos dispositivos podrían impedir la monitorización de los miembros de la red.
- Esta situación hace que estos sistemas sean especialmente vulnerables ante ataques provenientes, no ya del exterior, sino del interior de la propia infraestructura. Es vital tener este aspecto en cuenta a la hora de implementar un detector de intrusiones en esta localización.

6.1.4 Subredes de valor crítico

A veces, los servidores y recursos más importantes de una red son situados en una subred, separada de la red principal mediante dispositivos como cortafuegos (D). Para protegerlos debidamente, es necesario implementar detectores de intrusiones basados en red en estas subredes privadas.

Algunas de las ventajas de hacer esto son:

- Detectar ataques realizados contra elementos críticos de la red.
- Dedicar especial atención a los recursos más valiosos de la infraestructura.

A continuación se enumeran algunas desventajas del uso de esta opción:

- Como ya se comentó en las situaciones anteriores, este caso no permite detectar ataques que utilicen algoritmos de cifrado en sus comunicaciones.
- No evitan problemas de monitorización relacionados con el uso de conmutadores.

- No están estratégicamente bien situados ante ataques de origen interno.

6.1.5 Máquinas

Otra de las posibles formas de instalar este tipo de sistemas es en las propias máquinas (E), convirtiéndolas rastreadores de red. Los IDSs basados en red implementados de esta forma se denominan IDS de nodo de red (NNIDS) ("Network Node IDS").

La mayoría de los productos de detección basados en red nombrados antes se pueden implementar de esta forma. Cualquier detector basado en red, que permita la instalación de uno de sus agentes en una máquina, puede ser utilizado de esta forma.

Esta localización proporciona ventajas únicas:

- Se evitan los inconvenientes de la encriptación de las comunicaciones, presentes en las localizaciones anteriores. El NNIDS deja de recibir cifrado el tráfico originado o destinado a la máquina en la que está instalado. No obstante, seguirá percibiendo cifradas el resto de las comunicaciones.
- Es una forma de solventar problemas derivados del uso de conmutadores. Como ya se apuntó en el apartado 3.1.2 "Fuentes de información basadas en red", este tipo de dispositivos dificultan la monitorización del tráfico red, realizando tareas de encaminamiento, cosa que no hacen los concentradores. Situar un detector en una máquina permite al menos, examinar sus propias comunicaciones.

Por otra parte, este enfoque también tiene inconvenientes:

- La visión del sistema de detección está claramente limitada tanto por la situación de la máquina, como por la arquitectura de la red. Por ejemplo, si se utilizan conmutadores, sólo puede analizar el tráfico relacionado con la máquina anfitriona. No obstante, si se hace uso de concentradores, analizaría además el tráfico del resto de los miembros de la red, actuando como un rastreador.
- El NIDS está compartiendo los mismos recursos que la máquina que monitoriza. Esto reduce los recursos de la misma, afectando evidentemente a su rendimiento final.
- Que la máquina anfitriona sea comprometida puede tener graves consecuencias. El detector no sólo perdería toda eficacia, sino que además, podría ser controlado por el atacante para llevar a cabo sus fines. Obtener información sobre la infraestructura de la organización, o enviar falsas alarmas que distrajeran la atención del responsable de seguridad, son sólo algunos ejemplos de lo que un intruso podría hacer en dicha situación.

6.2 Sistemas de Detección de Intrusiones de máquina

Como se comentó al principio del capítulo, en una estrategia de implementación general, los detectores de intrusiones basados en máquina ("host") se suelen instalar después de los basados en red. Esto se hace así ya que, dadas sus características, son más complicados de instalar.

Este tipo de sistemas necesita ser configurado de forma individual en cada máquina, y utiliza como fuente de datos la información obtenida del sistema.

La mayoría de los detectores de intrusiones incluyen, entre otras funciones, mecanismos de verificación de integridad de archivos. Esto les permite, mediante el uso de algoritmos de cifrado como funciones resumen, reconocer cambios en los ficheros más importantes del sistema.

Aunque la situación ideal es la de contar con uno de estos sistemas *en cada una* de las máquinas de la red, lo cierto es que el procedimiento más extendido a seguir es el de instalarlos primero en los servidores más importantes. Una vez que los responsables se han acostumbrado a esta situación, se pueden ir implementando en el resto de los equipos.

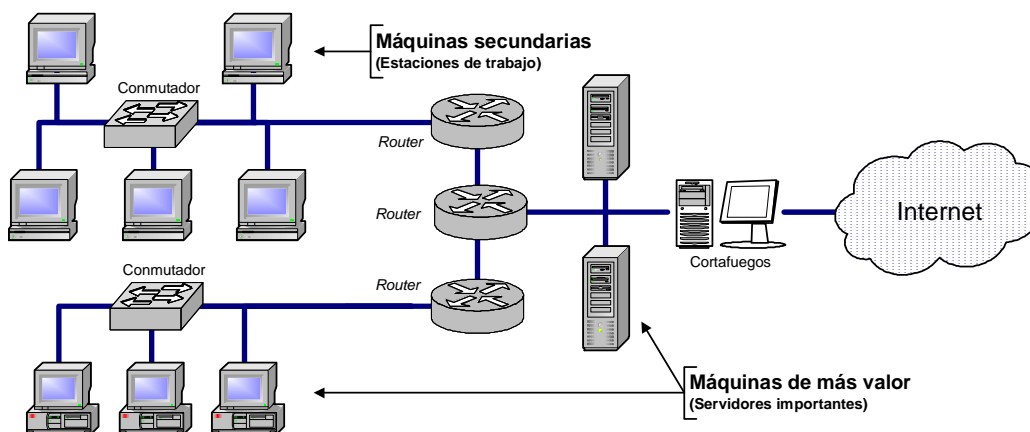


Figura 6-2 - Implementación progresiva de HIDS

Es muy importante que los administradores de estos sistemas se acostumbren a la forma de trabajar de estos sistemas, afinando la configuración para adaptarla a su situación particular, y aprendiendo a distinguir entre las falsas alarmas y los verdaderos problemas de seguridad.

Los informes emitidos por los detectores basados en máquina deben ser revisados de forma periódica. No siempre es posible ir examinando individualmente cada detector. Por ello, muchos productos facilitan mecanismos de centralización de registros, que permiten gestionar las alarmas de una forma más cómoda, rápida y eficiente.

Algunos de los productos más conocidos, que se utilizan como detectores de intrusiones basados en host son: GFI LANguard S.E.L.M, de GFI Software Ltd. [7]; Tripwire [8], LogCaster, de Enterprise International [9]; o el ya mencionado IDS híbrido, Prelude [6].

Algunas de las ventajas del uso de estos sistemas son:

- Trabajan con el sistema de ficheros, y con registros de sistema operativo locales, por lo que pueden detectar ataques que no identificados los detectores basados en red.
- Su especialización les otorga ventaja a la hora de detectar ataques específicos de los sistemas que monitorizan.
- Su posición privilegiada les permite identificar con precisión los elementos involucrados en un ataque, tales como procesos de sistema, ficheros o nombres de usuario.
- Dada su naturaleza, este tipo de sistemas no se ve afectado por un entorno de red con conmutadores.

6.3 Alarmas

Uno de los apartados más importantes de los IDSs es el relativo a las alarmas. Elaborar una arquitectura de los mecanismos de alarma y procedimientos de respuesta frente a ataques, antes de implementar estos sistemas, puede evitar muchos problemas en caso de ataque.

Estos sistemas permiten enviar una alarma de muchas formas: añadiendo un evento en los registros de auditoría o sistema, un mediante correo electrónico, utilizando protocolos de gestión de red (SNMP), a través de mensajes a móviles ("Short Message Service" (SMS)), etc.

Es recomendable dejar pasar unas semanas, antes de activar los mecanismos de alarma de un detector de intrusiones. Ese tiempo se debe dedicar para ajustar la configuración a cada escenario particular, reduciendo la aparición de falsas alarmas.

Por otra parte, si se configuran respuestas automáticas que permitan responder ante acciones hostiles, como por ejemplo, bloqueando la dirección origen del atacante, es preciso seguir de cerca su funcionamiento, para impedir que un intruso se aproveche de estas funciones. De lo contrario, un atacante podría "falsificar" su dirección origen, utilizando las de otras entidades, que podrían incluso pertenecer a clientes de la empresa atacada o de la propia red privada, provocando evidentes problemas.

Para asegurarse de que las alarmas llegan a su destino, se suelen utilizar técnicas de redundancia; se envía la misma alarma utilizando distintos métodos de comunicación. Este tipo de medidas se toma en casos de alarmas de nivel crítico.

6.4 Referencias

- [1] Bace, R. and Peter Mell. Intrusion Detection Systems. [en línea] [consultado en marzo, 2003]. Disponible en <<http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf>>.

- [2] Roesch, Marty et al. *Snort.org*. [en línea]. Actualizado semanalmente [consultado en marzo de 2003]. Disponible en <<http://www.snort.org>>.
- [3] Vern, Paxon. *Bro: A System for Detecting Network Intruders in Real-Time*. Lawrence Berkeley National Laboratory, Berkeley, CA and AT&T Center for Internet Research at ICSI, Berkeley, CA. [en línea]. 14 de diciembre de 1999 [consultado en marzo de 2003]. Disponible desde Internet en <<http://www.icir.org/vern/bro-info.html>>
- [4] Leach, John and Gianni Tedesco. *Firestorm*. [en línea]. 2002 [consultado en abril, 2003]. Disponible en <<http://www.scaramanga.co.uk/firestorm/index.html>>.
- [5] Tiny Software Inc. *Tiny Personal Firewall*. [en línea]. Fecha no disponible [consultado en abril, 2003]. Disponible en <<http://www.tinysoftware.com/>>.
- [6] Vandoorselaere, Yoann. *Prelude Hybrid IDS*. [en línea]. 1998 [consultado en abril, 2003]. Disponible en <<http://prelude-ids.org/>>.
- [7] GFI Software Ltd. *GFI Security Event Log Monitor*. [en línea]. Fecha no disponible [consultado en abril, 2003]. Disponible en <<http://www.gfi.com/>>.
- [8] Tripwire, Inc. *Tripwire*. [en línea]. Fecha no disponible [consultado en abril, 2003]. Disponible en <<http://www.tripwire.com/>>.
- [9] Enterprise International. *LogCaster*. [en línea] 1997 [consultado en abril, 2003]. Disponible en <<http://www.ei-europe.com/logcaster.htm>>.

Capítulo 7

Aspectos legales

Asegurar un sistema no sólo consiste en tomar las medidas necesarias para protegerlo ante ataques o u otros problemas, sino estar al tanto de los aspectos legales relacionados con el tema.

La implementación de un Sistema de Detección de Intrusiones en muchas ocasiones implica el cumplimiento de una serie de requisitos impuestos por la Ley. Cumplir con estas obligaciones permite perseguir a los culpables mediante procesos legales o judiciales. Los registros e informes proporcionados por un detector de intrusiones pueden ser requeridos como pruebas que ayuden a localizar y condenar a los responsables.

Desgraciadamente, los sistemas legales de muchos países no se han adaptado a la misma velocidad que el desarrollo de las tecnologías, dejando vacíos que permiten a los criminales delinquir con total impunidad.

En este capítulo se hará un breve repaso por el estado actual de los sistemas legales del mundo, para centrarse posteriormente en el marco legal de Europa y en el de España. Se comentarán los aspectos legales relacionados con delitos informáticos, y más concretamente, los relacionados con los Sistemas de Detección de Intrusiones

7.1 Sistemas legales en el mundo

Actualmente, los sistemas legales más utilizados en el mundo se pueden clasificar en seis tipos [1]: derecho civil, "common law", derecho consuetudinario, derecho musulmán, derecho talmúdico y derecho mixto. El "Derecho Mixto" es una combinación de dos o mas sistemas jurídicos y no a un tipo de sistema jurídico.

La Figura 7-1 ilustra la distribución de los sistemas legales en los diferentes países del mundo. A continuación se hará una descripción de cada uno de dichos sistemas.

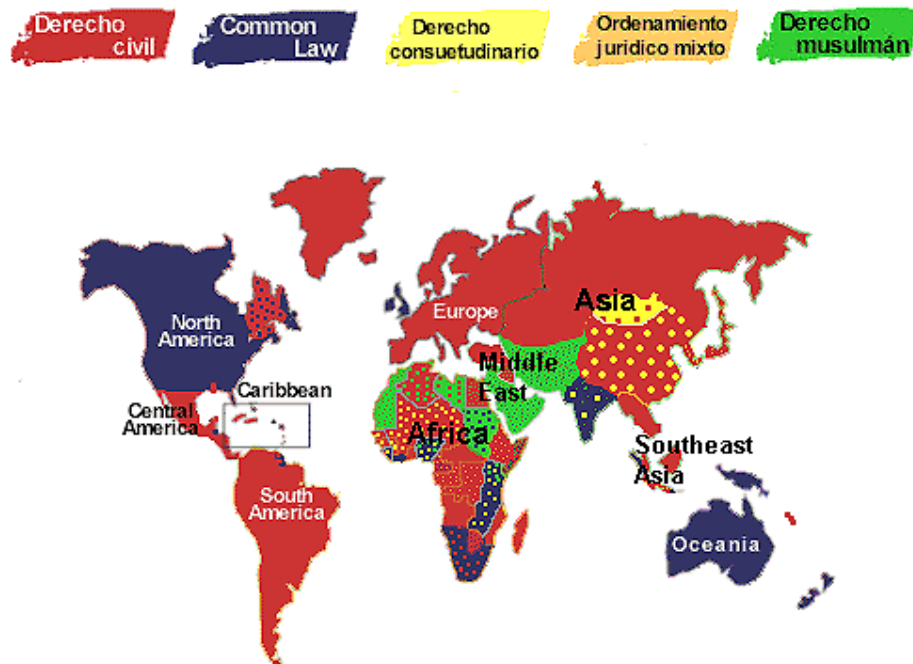


Figura 7-1 - Sistemas legales en el mundo

7.1.1 Derecho civil

Este sistema legal es el utilizado por aquellos países basados en el sistema legal romano. Profieren gran importancia al derecho escrito, y adoptan una codificación sistemática de su derecho común.

Por otro lado, también se encuentran en esta categoría aquellos países, generalmente de derecho mixto, que sin haber recurrido a la técnica de la ley codificada, poseen suficientes elementos de construcción jurídica romana, que permiten considerarlos como adscriptos a la tradición civilista.

También se incluyen en este tipo los países en los que, a pesar de que no contar con importante influencia romana, practican un derecho, codificado o no, que reposa en una concepción del rol de la ley similar a ésta de los países de tradición civilista "pura". Un ejemplo de este caso es el de los países de tradición escandinava.

Este es uno de los sistemas legales más practicados en todo el mundo. Entre los que lo han adoptado se encuentran muchos países europeos (España, Francia, Alemania, o Italia entre otros), así como gran parte de los países de Sudamérica (como Venezuela, Argentina, Brasil) y América Central (como México, El Salvador, Guatemala).

7.1.2 "Common law"

Otro de los sistemas legales más practicados en todo el mundo es el "Common law". En esta categoría entran aquellos países en los cuales el derecho reposa técnicamente, al menos en lo esencial, sobre los conceptos y los modos de organización jurídica del "common law" británico. Este modelo otorga gran importancia a la jurisprudencia, y no a la ley como medio ordinario de expresión del derecho común. En consecuencia, la mayoría de países adscritos a este sistema legal están más o menos relacionados con la tradición británica.

Entre los países que utilizan este sistema además de Inglaterra están Estados Unidos, Canadá, Irlanda, Australia o Nueva Zelanda.

7.1.3 Derecho consuetudinario

Aunque actualmente no existen países cuyos sistemas legales puedan ser enteramente consuetudinarios, el derecho consuetudinario juega un importante papel en gran número de países de derecho mixto.

Este sistema es practicado en algunos países africanos. También es aplicado, con diferentes condiciones, en países como China e India.

7.1.4 Derecho Musulmán, Derecho Talmúdico

Estos son sistemas autónomos de derecho religioso propiamente dicho. No hay separación entre el estado y la religión, al contrario que en el derecho canónico. Este último, aunque influenciado por dogmas religiosos, es producto de la elaboración humana es uno de los componentes de la tradición civilista.

Con excepción de Afganistán o las Islas Maldivas, los países que se rigen por este sistema lo utilizan en conjunción con algún otro (como "Common Law" o Derecho Civil). La mayoría de estos países se encuentran en oriente medio.

7.1.5 Derecho Mixto

Se engloban en esta categoría aquellos países donde dos o más sistemas se aplican de manera acumulativa o de interacción. También pertenecen a la misma aquellos países en los cuales hay una yuxtaposición de sistemas, dado que los mismos se aplican simultáneamente a áreas más o menos diferenciadas.

7.1.6 Territorios no independientes

Por último, cabe mencionar cierto número de sistemas aplicados en una serie de territorios no independientes, que por diversos motivos no están vinculados al sistema jurídico de la metrópolis. Estos sistemas, bien han adquirido o mantenido características distintas dentro del nombre federal o unidad política a la cual pertenece.

7.2 Otros sistemas

7.2.1 Derecho penal

Además de los sistemas antes mencionados, otra parcela importante en este caso es el Derecho penal, cuyo contenido es un catálogo de normas que protegen los valores que una sociedad considera más importantes, así como las sanciones que se pueden imponer para el caso de incumplimiento. Por tanto, el Derecho penal contiene una lista de delitos y sus penas.

Esta lista de delitos y penas puede provenir de un sistema de codificación, como en todo el mundo occidental, o de textos incluso religiosos (El Corán) pero es importante señalar que los derechos humanos y libertades fundamentales tienen gran trascendencia en este campo, por lo que rigen principios universales no aplicables a otras ramas del Derecho, como el principio de legalidad, *nulla poena sine praevia lege* que exige que para que se produzca un delito, debe, forzosamente, existir una Ley previa que defina dicho delito. Debido a esta razón, la costumbre no es fuente de este Derecho, ni siquiera en los países de Derecho consuetudinario o de Common Law.

7.2.2 Derecho procesal

Si los análisis de la detección de intrusiones tienen como destino servir de prueba en un procedimiento judicial, deberán asimismo tenerse en cuenta los diferentes requisitos legales para que sean válidas y eficaces la obtención, conservación y presentación de las pruebas en juicio. En este aspecto, cada país tiene sus reglas, no pudiéndose establecer en este caso categorías comunes.

7.3 Situación en Europa

En lo que respecta a la situación legal sobre delitos informáticos, los países pertenecientes al Consejo de Europa acordaron el 21 de noviembre de 2001 el "Convenio sobre la Ciberdelincuencia", en el que también participaron los Estados Unidos. Este documento fue firmado por los representantes de cada país, aunque su eficacia depende de su posterior refrendo por los órganos nacionales de cada país firmante.

Este documento sirvió para definir los delitos informáticos y algunos elementos relacionados con éstos, tales como "sistemas informáticos", "datos informáticos", o "proveedor de servicios". Los delitos informáticos fueron clasificados en cuatro grupos descritos a continuación:

- **Delitos contra la confidencialidad, la integridad y la disponibilidad de los datos y sistemas informáticos.**
 - Acceso ilícito a sistemas informáticos.
 - Interceptación ilícita de datos informáticos.
 - Interferencia en el sistema mediante la introducción, transmisión, provocación de daños, borrado, alteración o supresión de estos.
 - Abuso de dispositivos que faciliten la comisión de delitos.
- **Delitos informáticos.**

- Falsificación informática que produzca la alteración, borrado o supresión de datos informático que ocasionen datos no auténticos.
- Fraudes informáticos.
- **Delitos relacionados con el contenido.**
 - Delitos relacionados con la pornografía infantil.
- **Delitos relacionados con infracciones de la propiedad intelectual y derechos afines.**

Los delitos relacionados con los ataques detectados por los Sistemas de Detección de Intrusiones estarían principalmente contemplados en la primera categoría: "Delitos contra la confidencialidad, la integridad y la disponibilidad de los datos y sistemas informáticos".

En el "Convenio sobre la Ciberdelincuencia" se encomienda a cada Parte que tome las medidas necesarias para tipificar como delito en su derecho interno cada uno de los apartados descritos en cada categoría. A continuación se describirá cuál es la situación en España.

7.4 Situación en España

A pesar de que en España se están haciendo progresos en el plano legal para regular los delitos relacionados con las tecnologías de la información, lo cierto es que la situación aún es muy precaria. En este apartado se hará referencia a los principales elementos relacionados con el marco legal que tienen que ver con los delitos informáticos, intentando destacar aquellos temas relacionados con los Sistemas de Detección de Intrusiones.

7.4.1 Legislación

Se entiende por *delito informático* todo ilícito penal llevado a cabo a través de medios informáticos y que está íntimamente ligado a los bienes jurídicos relacionados con las tecnologías de la información o que tiene como fin estos bienes [2].

España cuenta con un Código Penal en el que no existe ningún título que se refiera específicamente a delitos informáticos. No obstante, se pueden encontrar algunos tipos penales adaptables a los definidos en el "Convenio sobre la Ciberdelincuencia".

Por otra parte, en España existe un conjunto de leyes, descrito más adelante, que complementa la labor de regulación de las Tecnologías de la Información.

7.4.1.1 Delitos informáticos y el Código Penal

Los tipos penales definidos en el Convenio del Consejo de Europa se pueden encontrar reflejados en el Código penal español de 1995 (Ley Orgánica 10/1995, de 23 de Noviembre). De esta forma se extraen las siguientes conductas delictivas, en las que los datos o sistemas informáticos son instrumentos de comisión del delito o el objeto del delito:

- **Delitos contra la confidencialidad, la integridad y la disponibilidad de los datos y sistemas informáticos.**

Artículo 197	Se tipifica en este artículo las conductas que llevan a apoderarse de mensajes de correo electrónico ajenos o accedan a documentos privados sin la autorización de sus titulares.
Artículo 264.2 Artículo 278.3	La destrucción, alteración o daño de programas o documentos contenidos en ordenadores
Artículo 278.1	Apoderarse o difundir documentos o datos electrónicos de empresas.

- **Delitos informáticos.**

Artículo 248.2	Estafas como consecuencia de alguna manipulación informática.
Artículo 256	Utilización no consentida de un ordenador sin la autorización de su dueño causándole un perjuicio económico superior a 300,50 €.

- **Delitos relacionados con el contenido.**

Artículo 186	La distribución entre menores de edad de material pornográfico.
Artículo 189	Distribución a través de Internet de material de pornografía infantil.

- **Delitos relacionados con infracciones de la propiedad intelectual y derechos afines.**

Artículo 270	La copia no autorizada de programas de ordenador o de música.
Artículo 270	Fabricación, distribución o tenencia de programas que vulneran las medidas de protección anti-piratería de los programas .
Artículo 273	Comercio a través de Internet de productos patentados sin autorización del titular de la patente

Una vez más, se debe mencionar que la mayoría de los actos delictivos relacionados con un detector de intrusiones están recogidos en el primer grupo.

A pesar de recoger muchos de los casos descritos por el Consejo de Europa, algunas conductas como el "Spam"¹, escanear puertos², la apología del terrorismo a través de Internet o el blanqueo de capitales no están contemplados entre los delitos tipificados en nuestro Código Penal. Debido a esta razón, su persecución penal se realiza conjuntamente con los delitos a los que los ordenadores o las redes sirven como la herramienta para su comisión, no siendo considerados delitos autónomos en sí mismos.

¹ Envío de publicidad no solicitada, generalmente a través del servicio de correo electrónico.

² Técnica común para obtener información de un sistema. Relativamente común entre las alarmas de un IDS.

7.4.1.2 Delitos informáticos y el C.N.P.

Los delitos informáticos contemplados en España, según la Brigada de Investigación Tecnológica del Cuerpo Nacional de Policía, se pueden clasificar en los siguientes, con su correspondencia en el Código Penal. El texto que acompaña a cada artículo es una explicación del delito, y no se corresponde con su contenido [3]:

- **Ataques que se producen contra el derecho a la intimidad**

**Artículos
del 197 al
201**

Delito de descubrimiento y revelación de secretos mediante el apoderamiento y difusión de datos reservados registrados en ficheros o soportes informáticos.

- **Infracciones a la Propiedad Intelectual a través de la protección de los derechos de autor**

**Artículos
270 y
otros**

Especialmente la copia y distribución no autorizada de programas de ordenador y tenencia de medios para suprimir los dispositivos utilizados para proteger dichos programas.

- **Falsedades**

**Artículos
386 y ss.**

Concepto de documento como todo soporte material que exprese o incorpore datos.

Extensión de la falsificación de moneda a las tarjetas de débito y crédito.

Fabricación o tenencia de programas de ordenador para la comisión de delitos de falsedad.

- **Sabotajes informáticos**

**Artículo
263 y
otros**

Delito de daños mediante la destrucción o alteración de datos, programas o documentos electrónicos contenidos en redes o sistemas informáticos.

- **Fraudes informáticos**

**Artículos
248 y ss.**

Delitos de estafa a través de la manipulación de datos o programas para la obtención de un lucro ilícito.

- **Amenazas**

**Artículos
169 y ss.**

Realizadas por cualquier medio de comunicación.

- **Calumnias e injurias**

**Artículos
205 y ss.**

Cuando se propaguen por cualquier medio de eficacia semejante a la imprenta o la radiodifusión.

- **Pornografía infantil**

**Artículo
189**

La producción, venta, distribución, exhibición, por cualquier medio, de material pornográfico en cuya elaboración hayan sido utilizados menores de edad o incapaces, aunque el material tuviere su origen en el extranjero o fuere desconocido.

El facilitamiento de las conductas anteriores (El que facilitare la producción, venta, distribución, exhibición...).

La posesión de dicho material para la realización de dichas conductas.

7.4.1.3 Legislación adicional

Existe un cuerpo legislativo, fuera del ámbito penal, que pretende regular el aspecto de la Sociedad de la Información. alguna de estas leyes ha sido especialmente formulada con ánimo de proteger la intimidad y privacidad de los ciudadanos y sus datos. Conocer y cumplir los requisitos descritos en las mismas hace posible la adopción de comportamientos útiles legalmente en caso de delito.

- **Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal (LOPD).** [4]
 - Supone una modificación importante del régimen sobre protección de datos de personas físicas contenido hasta entonces en la extinta LORTAD. Esta norma por introduce en el marco jurídico unos valores sobre la defensa de la intimidad y privacidad de los ciudadanos y consumidores, a los que reconoce un conjunto de derechos. No obstante, la ambigüedad y falta de precisión de ciertos términos y situaciones, dificulta su aplicación.
- **Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y Comercio Electrónico (LSSICE).** [5]
 - Esta ambiciosa Ley supone la primera regulación legal que con carácter general se dicta en España para el entorno de Internet. Aunque su principal objetivo consiste en aplicar la Directiva 200/31/CE (Directiva del Comercio Electrónico). También define otros factores relacionados con la "Sociedad de la Información", como las obligaciones de Servicio Universal, o la legalidad o ilegalidad de los actos que cualquier particular puede realizar en la Red.
- **Real Decreto Legislativo 1/1996, de 12 de abril (BOE 22-4-1996), por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual.** [6]
 - Esta Ley regula, aclara y armoniza las disposiciones legales vigentes sobre este tema. Constituye la referencia principal relativa a la regulación de la propiedad intelectual en España.
- **Real Decreto Legislativo 14/1999, de 17 de septiembre, sobre Firma Electrónica.** [7]
 - Reconoce la eficacia jurídica de la firma electrónica¹ y las condiciones para prestar servicios de certificación en España. Actualmente se está tramitando una nueva Ley de Firma Electrónica, que se halla en fase de Proyecto de Ley en esta fecha.

7.4.2 Intrusiones y la Legislación española

Las intrusiones, o accesos no consentidos, tienen cierta consideración por parte del Código penal español, el cual contempla dos casos concretos:

- "Hacking"² directo, mero acceso no consentido: Se define así a las intrusiones perpetradas con el único fin de vulnerar un mecanismo de seguridad que permita el

¹ Definida en este Real Decreto como: "Conjunto de datos, en forma electrónica, ajenos a otros datos electrónicos o asociados funcionalmente con ellos, utilizados como medio para identificar formalmente el autor o a los autores del documento que la recoge".

² Aquí se adopta la palabra "hacking" para describir aquellas acciones que implican una intrusión o violación de seguridad. A pesar de que la comunidad "hacker" suele definir este tipo de actividades con el término "cracking", más correcto, lo cierto es que nunca ha sido ampliamente aceptado.

acceso a sistemas informáticos o redes de comunicación electrónica de datos. El intruso sólo accede al sistema y sale, demostrando el fallo de seguridad del mismo, sin ánimo delictivo en esta conducta. El mero acceso y la mera permanencia no autorizada, actualmente no está castigada por el Código penal español, a diferencia de lo ocurrido en otros países, como Francia, que sí castiga y persigue este caso.

- "Hacking" indirecto: Consiste en el acceso no consentido a un sistema informático o redes de comunicación electrónica de datos con el fin de cometer un delito. En este caso la intrusión se concibe como un medio necesario para cometer el delito final cuyo móvil guía al sujeto desde el principio. En este caso el acceso queda subsumido en el delito finalmente cometido (descubrir secretos de empresa, vulnerar el *habeas data*¹, interceptar las comunicaciones, producir daños, etc.).

Por lo tanto, y según lo descrito, un mero acceso no consentido no constituye un delito en España.

7.4.3 Cuerpos especiales

En España se han creado organismos especiales de investigación tanto en el Cuerpo Nacional de Policía, como la Brigada de Investigación Tecnológica [3], como en la Guardia Civil, con el Grupo de Delitos Telemáticos [8]. Además, se les ha provisto de medios técnicos cada vez más avanzados para poder ejercer su labor.

Los mismos medios utilizados por los delincuentes para cometer sus delitos sirven también a los especialistas para establecer medidas de seguridad y obtener pruebas que los identifiquen e inculpen.

7.4.4 Necesidades y deficiencias

Como ya se comentó al principio del apartado sobre la situación en España, el marco legal español, a pesar del Código penal actual y las unidades especiales para el control de los delitos informáticos, presenta importantes limitaciones a la hora de perseguir a los delincuentes informáticos. La rapidez con que se desarrollan las nuevas tecnologías y la posibilidad de actuar desde cualquier parte del mundo, hace de los delitos informáticos uno de los retos más importantes a los que se enfrentan las autoridades legales y judiciales de los países más desarrollados. A continuación se muestran una serie de factores que complican la labor de estas entidades:

- Determinar la jurisdicción competente.
- Delitos cometidos desde fuera de España, provenientes de un país en el que no exista regulación sobre el tema.
- Dificultad para obtener pruebas fehacientes que inculpen al delincuente.

¹ Significa "que tengas los datos" o "que vengan los datos", es decir, tener conocimiento de la existencia de datos propios en poder de otro.

- Dificultad para identificar al autor del delito. No hay que olvidar que las técnicas de ocultación de dirección IP por parte de un intruso pueden hacer imposible su localización.
- Falta de adaptación de los organismos legislativos a los rápidos cambios y nuevas situaciones provocadas por la aparición de las nuevas tecnologías. Y necesidad de mayor cooperación entre distintos países para abordar el tema. Esto reduciría el ámbito de actuación de muchos delincuentes que se aprovechan de la situación actual.

7.5 Referencias

- [1] Universidad de Ottawa. Facultad de Derecho. *Los sistemas jurídicos del mundo*. [en línea] Actualizado el 23 de marzo, 2003 [consultado en mayo, 2003]. Disponible en <<http://www.droitcivil.uottawa.ca/world-legal-systems/esp-monde.html>>.
- [2] García Noguera, Noelia. *Delitos informáticos en el Código penal español*. [en línea] 15 de Julio de 2002. [consultado en mayo 2003]. Disponible en <<http://www.portaley.com/delitos-informaticos/codigo-penal.shtml>>.
- [3] Cuerpo Nacional de Policía de España. Brigada de Investigación Tecnológica. [en línea]. Fecha no disponible [consultado en mayo 2003]. <<http://www.mir.es/policia/bit/legisla.htm>>.
- [4] Boletín Oficial del Estado (BOE). *Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal*. [en línea] 13 de diciembre de 1999 [consultado en mayo 2003]. Disponible en <<http://www.boe.es/boe/dias/1999-12-14/pdfs/A43088-43099.pdf>>.
- [5] Boletín Oficial del Estado (BOE). *Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico*. [en línea] 12 de julio de 2002 [consultado en mayo 2003]. Disponible en <<http://www.boe.es/boe/dias/2002-07-12/pdfs/A25388-25403.pdf>>.
- [6] Ministerio de Educación Cultura y Deporte. *Propiedad Intelectual*. [en línea]. Fecha no disponible [consultado en mayo 2003]. <http://www.mcu.es/Propiedad_Intelectual/indice.htm>.
- [7] Boletín Oficial del Estado (BOE). *Real Decreto-Ley 14/1999, de 17 de septiembre, sobre firma electrónica*. [en línea] 18 de septiembre de 1999 [consultado en mayo 2003]. Disponible en <<http://www.boe.es/boe/dias/1999-09-18/pdfs/A33593-33601.pdf>>.
- [8] Guardia Civil. *Grupo de Delitos Telemáticos*. [en línea]. Fecha no disponible. [consultado en mayo 2003]. Disponible en <<http://www.guardiacivil.org/00telematicos/legislacion.htm>>.

Necesidades, líneas de trabajo

Aunque han aparecido hace ya varios años, los Sistemas de Detección de Intrusiones aún poseen muchos aspectos que necesitan mejorar. Existen grupos de desarrollo cuya labor se centra en algunas de estas características.

En este capítulo se abordan las principales deficiencias de estos sistemas y las líneas de trabajo que se llevan a cabo para solventarlas. [1]

8.1 Normalización

Salvo raras excepciones, ningún sistema suele llegar a ser utilizado ampliamente si no se normaliza o regulariza de alguna forma.

El concepto de normalización está presente en la vida cotidiana. Se pueden adquirir dos paquetes de hojas de diferentes fabricantes que cumplan la misma norma respecto al formato y tamaño. Los aparatos eléctricos de distintas empresas tienen enchufes que se pueden conectar a la red del hogar. Un teléfono móvil comprado en Europa, puede ser utilizado en Estados Unidos, si contempla los requisitos para realizar llamadas en esa región.

Todo esto se debe a la presencia de normas y estándares. Los organismos encargados de crear normas permiten que diferentes empresas adopten la misma solución para un determinado problema.

En lo que respecta a las tecnologías de la información, existen normas que definen normas y estándares para interfaces de comunicación y programación. Los protocolos de comunicación, como IP, TCP, o ICMP son tan sólo algunos ejemplos de este tipo de normas.

Por otra parte, también existen soluciones basadas en traductores, o vehículos de comunicación, con independencia de la plataforma o arquitectura con que se trabaja. Este enfoque también permite el acercamiento y trabajo conjunto de distintas plataformas. El sistema XDR (External Data Representation), por ejemplo, permite la transferencia de datos entre entidades de arquitecturas diferentes, tales como Estaciones SUN, VAX, IBM-PC, o Cray.

La detección de intrusiones no es una excepción al tema de la regularización. Como cualquier otro reto de gran envergadura, la colaboración y acuerdo entre diferentes organismos y empresas se hace cada vez se hace más necesaria. Muchas empresas, ante la ausencia de una solución óptima, han propuesto sus propios productos. Esto provoca situaciones en las que se puede llegar a utilizar cuatro productos diferentes para aprovechar las peculiaridades de cada uno. Afortunadamente existe cierto número de propuestas por parte de diferentes entidades, para evitar este tipo de situaciones.

8.1.1 CIDF

El "Common Intrusion Detection Framework" (CIDF) es uno de los proyectos más ambiciosos en el ámbito de la normalización de la detección de intrusiones. Fue fundado por Teresa Lunt, para la "Defense Advanced Research Projects Agency" (DARPA). [2]

Es un grupo encargado de crear interfaces que permitan a los desarrolladores de detección de intrusiones compartir sus conocimientos y así poder reutilizar los componentes obtenidos en otros sistemas.

Uno de los resultados de la creación del CIDF, fue la aparición de un grupo de trabajo especializado en la detección de intrusiones por parte de la "Internet Engineering Task Force" (IETF), comentado más adelante.

8.1.2 CRISIS

El "Critical Resource Allocation and Intrusion Response for Survivable Information Systems" (CRISIS), es un grupo de trabajo de similares objetivos a CIDF. Tienen como objetivo unificar los mecanismos de detección, respuesta y control de los sistemas de detección de intrusiones [3]. En concreto, trabajan en dos aspectos:

- Diseñar una arquitectura común para el desarrollo de la detección de intrusiones: En el marco actual, existen numerosas herramientas de detección de intrusiones y de respuesta ante intrusiones. CRISIS intenta proporcionar una plataforma que permita la cooperación entre ellas.
- Proveer mecanismos para la asignación de recursos críticos: Algunos ataques pueden llegar a bloquear al sistema comprometido. Este aspecto se centra en buscar soluciones que permitan a los sistemas de detección utilizar sus recursos para llevar a cabo sin problemas sus tareas de respuesta ante posibles ataques.

8.1.3 Formato de los datos de auditoría

Así como la gran mayoría de servidores Web utilizan un formato de registro ya definido en un estándar, los detectores de intrusiones también se pueden beneficiar de este aspecto. No obstante, esta no es una tarea trivial. El formato de los registros de auditoría está muy ligado al sistema operativo y, por desgracia, existen diferencias sustanciales entre muchos ellos. Este tema es uno de los principales obstáculos con los que se enfrenta la detección de intrusiones en su camino hacia la estandarización.

8.1.3.1 Libro Naranja, Libro Marrón

Una de las propuestas relativas al registro de auditoría está definida en el libro naranja ("Trusted Computer System Evaluation Criteria") [4] y en el libro Marrón ("Tan Book") [5].

En estos documentos se indican con detalle los requisitos que deben cumplir los mecanismos de auditoría.

No obstante, estas especificaciones han provocado numerosas confusiones debido a la falta de exactitud en algunos detalles. Por ejemplo, especifican el contenido con que deben contar los registros, pero no su formato. Por otra parte, indican aquellos eventos que pueden ser auditados, pero no cuáles deben ser auditados.

Esto ha provocado la aparición de sistemas operativos que, a pesar de cumplir con los requisitos descritos por estos documentos, no tienen nada que ver en cuanto a sus mecanismos y registros de auditoría.

8.1.3.2 IDES de Denning

Dorothy Denning propone en su documento sobre detección de intrusiones, un formato de registro de auditoría independiente de la plataforma. Desgraciadamente, este formato no aporta suficientes datos sobre la actividad del sistema para proporcionar detecciones de usos indebidos eficaces. [6]

8.1.3.3 SVR 4++ de Smaha

Steve Smaha propuso un formato de registro para sistemas UNIX denominado SVR++ 4, utilizado por las herramientas de detección de los Laboratorios Haystack, como STALKER. Este formato ha sido utilizado por varias empresas para sus productos de detección de intrusiones. [7]

8.1.3.4 Bishop

Matt Bishop, de la Universidad de California, Davis, desarrolló un formato de registro de auditoría estándar. Este formato está diseñado para ser independiente del sistema operativo o la plataforma de aplicación. [8]

8.1.3.5 IETF/IDWG

El "Internet Engineering Task Force" (IETF) creó un grupo de trabajo relativo al área de seguridad, centrado en la detección de intrusiones, denominado "Intrusion Detection Working Group" (IDWG). Algunos de los objetivos de este grupo son:

- Redactar un documento que especifique los requisitos para la comunicación entre IDS, y entre IDS y sistemas de gestión.
- Desarrollar una especificación de un lenguaje de intrusiones común, que describa un formato de datos que cumpla los requisitos.
- Elaborar un documento que describa los mejores protocolos para la comunicación entre IDS, y que indique qué relación tienen con los formatos de los datos.

Han desarrollado un sistema de intercambio de datos, y mecanismos de transporte, que permiten a los detectores de intrusiones contar con un sistema de mensajería específico, que les permite compartir los datos obtenidos entre sí. [9]

8.1.3.6 Mecanismos de auditoría

Hay que decir que algunos organismos como "Portable Operating System Interface" (POSIX) o X/OPEN han propuesto estándares de auditoría. No obstante, sus propuestas han

incluido Interfaces de Programación de Aplicaciones (APIs), o mecanismos de auditoría y no formatos de auditoría.

8.2 Integración

La integración está muy relacionada con los objetivos del apartado anterior. Consiste en desarrollar un sistema de tal forma que pueda entenderse con el resto de elementos de su entorno.

La integración, en lo que respecta a la detección de intrusiones, implica por ejemplo, poder interpretar correctamente el formato de las fuentes de datos (registros de auditoría del sistema operativo, tráfico heterogéneo de red), así como de contar con métodos que permitan interactuar y compartir recursos con otros mecanismos del sistema.

Este aspecto permite a diferentes sistemas de seguridad coordinar sus esfuerzos y resultados para encontrar y aportar pruebas sobre un determinado ataque o intrusión.

8.3 Escalabilidad

La gestión de la seguridad en redes se puede convertir en una tarea difícil cuando alcanzan gran tamaño. Este es uno de los retos a los que se enfrentan los detectores de intrusiones. La capacidad de proceso y otros recursos de sistema pueden ser insuficientes en determinadas ocasiones. A continuación se describen los dos tipos de elementos más comunes asociados a problemas de escalabilidad: el tiempo y el espacio.

8.3.1 Tiempo

Los problemas de escalabilidad asociados al tiempo se producen cuando un ataque se realiza de forma extremadamente larga. Esto puede provocar que un detector de intrusiones no tenga recursos para almacenar y relacionar todos los indicios de una intrusión. Por ejemplo, si un atacante decidiera escanear un servidor de forma intencionadamente lenta, abriendo un puerto cada hora, o cada día, tardaría mucho en completar su objetivo, pero también podría conseguir que el detector no reconociera el progreso de el ataque.

Estas situaciones se pueden solventar ampliando la franja de tiempo en que el detector debe asociar los eventos sucedidos. Pero esto tiene un límite físico, asociado a los recursos de sistema, como la cantidad de memoria o disco duro disponible.

No obstante, no se pueden añadir sin más recursos del sistema para extender el intervalo de tiempo en que un detector puede relacionar los eventos ocurridos. Un intervalo demasiado amplio podría hacer que el sistema asociara eventos que, por su lejanía en el tiempo, pueden no tener nada que ver, aumentando el número de falsos positivos.

8.3.2 Espacio

El otro elemento que puede intervenir en la escalabilidad de una infraestructura es el espacio. Esta relacionado con el número de miembros de una red. Cuando dicho número empieza a

ser del orden de varios miles, un diseño escalable del sistema de detección de intrusiones puede ayudar mucho en adaptarse a la nueva situación.

Cuando se gestiona una red de gran tamaño, no sólo puede tener muchas máquinas, sino que además, pueden estar repartidas por diferentes zonas geográficas, y tener distintas velocidades de conexión. Esto provoca la aparición de problemas asociados a las diferencias de reloj de cada miembro. Por otra parte, esto también puede dificultar la forma de representar los datos obtenidos.

El sistema de intrusiones utilizado, debe ofrecer facilidades para implementarse de forma estratificada, con una estructura jerárquica en forma de árbol. Los detectores más elementales, pueden dedicarse a monitorizar grupos de máquinas de forma local, y ser controlados y coordinados por una serie de detectores de niveles superiores.

8.3.2.1 GrIDS

"Graph-Based Intrusion Detection System" (GrIDS) es un proyecto apoyado por "Defense Advanced Research Projects Agency" (DARPA), desarrollado por la Universidad de California. Es un ejemplo de sistema de detección cuyo diseño permite adaptarse a los problemas de escalabilidad de espacio.

Elabora grafos de actividad de las máquinas de una red, para poder identificar ataques a gran escala. Por esta razón, se suele utilizar especialmente para la detección de ataques en los que está involucrado un importante número de máquinas, como los DDoS (ataques de denegación de servicio distribuida), o ataques provocados por gusanos. También es de gran ayuda en ataques realizados por un escáner de vulnerabilidades, como Nessus, o SATAN.

La estructura de GrIDS permite construir grafos de actividad utilizando los datos obtenidos de múltiples detectores de intrusiones. A nivel general, los resultados del motor de grafos son comparados con una base de patrones de grafos, configurada por el administrador. Esto se hace para distinguir entre aquellos grafos que indican patrones de actividad poco habitual o sospechosa, y grafos de actividad normal.

8.4 Administración y gestión

Los factores descritos en el apartado anterior pueden afectar a la administración de un sistema de detección de intrusiones. Un sistema que no esté bien diseñado, puede ver limitadas sus capacidades de control en una red de importante tamaño.

El número de sensores implicados en la gestión de una gran infraestructura implica contar con métodos especiales para coordinar los datos recibidos de ellos. Carecer de las funciones apropiadas puede hacer imposible la tarea de detectar intrusiones.

Para la gestión de la red, se puede hacer uso de mensajes del Protocolo Simple de Gestión de Red (SNMP). Las últimas versiones de dicho protocolo, ofrecen mecanismos de cifrado para proteger las comunicaciones. No obstante, aún no hay muchos productos de detección de intrusiones que soporten esta opción. Además, el uso de este protocolo puede poner sobre aviso a los posibles atacantes.

Otro de los aspectos clave para la administración de un sistema de detecciones distribuido es el relativo a la centralización los diversos elementos que lo forman, tales como sensores, o

motores de análisis. Por un lado, la distribución de estos elementos es vital cuando se trata de monitorizar redes grandes. Hay muchas ventajas de hacer esto, entre las cuales está conseguir más estabilidad, la posibilidad de repartir el trabajo entre varios elementos, o vigilar puntos claves de la red. Sin embargo, esto también implica aspectos negativos como mayor complejidad en la gestión del sistema, o un aumento del tráfico de red originado por las comunicaciones entre los elementos del detector.

Muchos detectores de intrusiones carecen de las suficientes funciones para poder investigar tanto los ataques a un sistema como sus resultados. Este hecho puede dificultar procedimientos como el análisis forense, la recuperación ante una intrusión, o la evaluación de daños.

Un detector de intrusiones debería facilitar mecanismos para ayudar a reconocer la incidencia, aislar los puntos de entrada del intruso, identificar los métodos utilizados para comprometer el sistema, y determinar los efectos derivados de la intrusión en el sistema. Esto no sólo permitiría a un administrador identificar las características del ataque, sino que le serviría para reparar y prevenir el sistema ante futuros ataques de similares características.

El aumento de los delitos informáticos ha provocado la aparición de nuevas leyes que contemplen este tipo de actos. Cada vez son más los sistemas de detección que incluyen soporte para llevar a cabo procedimientos legales.

Algunos ataques, como los de denegación de servicio, tienen como objetivo interrumpir la continuidad de un servicio determinado. Esto puede provocar la caída de un sistema, anulando su gestión. Muchos sistemas de detección de intrusiones intentan solventar este problema, y cuentan con funciones como balanceo de carga, o ajustes de tiempo de ejecución.

8.5 Análisis

Hay una serie de aspectos relacionados con el análisis en los detectores de intrusiones que necesitan especial atención.

8.5.1 Detectores basados en inteligencia artificial

La detección de intrusiones mediante técnicas de inteligencia artificial o técnicas no paramétricas es una de las áreas con más posibilidades dentro de estos sistemas de seguridad. Los detectores basados en las técnicas mencionadas, utilizan grandes cantidades de datos de entrenamiento para determinar qué actividad es normal y cuál es anómala. No obstante, el conjunto de datos de entrenamiento debe cumplir una serie de características que no siempre son fáciles de conseguir:

- Debe ser lo suficientemente completo como para reflejar todas las actividades de comportamiento *normal* del sistema.
- Debe estar libre de ataques. De no ser así, naturalmente, el detector podría etiquetar dichos ataques como comportamientos normales.

- Debe ser lo menos local posible. Esto es, no debe centrarse en aspectos o localizaciones particulares de una infraestructura. De lo contrario, el conjunto de datos podría no ser aplicable a otras zonas de la organización.

8.5.2 Falsas alarmas

Uno de los problemas que han impedido que la detección de anomalías no haya sido ampliamente aceptada por los usuarios, es el relativo a las falsas alarmas. Este problema sigue siendo uno de los retos más importantes para los desarrolladores. Ajustar los motores de detección de anomalías requiere mucho tiempo y un amplio conocimiento sobre el entorno administrado.

Los falsos positivos (errores de Tipo I) tienen lugar cuando el detector cree reconocer una intrusión cuando realmente no lo es. Si se establece una sensibilidad alta para un detector de anomalías, probablemente el número de este tipo de alarmas será importante. Esto puede hacer que un administrador acabe ignorándolas.

Si el número de falsos negativos (errores de Tipo II) es notable, el administrador dejará de confiar en el detector, ante su incapacidad para reconocer ataques reales.

8.5.3 Políticas de sistema

La traducción de las políticas administrativas en políticas de monitorización y de detección es uno de los aspectos que más tiempo suele llevar a la hora de implementar un detector de intrusiones. Esto se debe a las diferencias existentes entre la formulación de ambos grupos de políticas.

Las políticas administrativas indican comportamientos de usuarios, adecuados o no. Normalmente son independientes de la plataforma a implantarse, y están expresadas en términos de objetivos, direcciones, o intenciones de los usuarios, pero no de las máquinas que utilizan.

Las políticas de monitorización y detección descritas para los detectores de intrusiones deben, sin embargo, estar expresadas en términos de eventos que ocurren en un determinado sistema. Por lo tanto, son dependientes de la plataforma.

Algunas herramientas permiten elaborar políticas mediante simples cuestionarios, o aprovechando los resultados emitidos por un escáner de vulnerabilidades. Esta última forma de trabajo es utilizada por ciertos Sistemas de Prevención de Intrusiones, como los Conmutadores Híbridos, mencionados en el capítulo 4. Este tipo de medidas ayudan a integrar las herramientas de seguridad con las funciones de gestión de sistema y de red.

La definición de políticas en el nivel de aplicación mejora la adaptación de interfaces de construcción de políticas a un entorno determinado.

8.6 Fiabilidad

Para que un sistema de detección de intrusiones sea eficaz, configurarlos correctamente no es una medida suficiente. Los mecanismos involucrados en los procesos de detección deben estar provistos de elementos que los protejan en caso de caídas o errores del sistema o ataques.

Cada componente perteneciente al proceso de detección puede dejar de ser fiable. El objetivo es reducir al mínimo las posibilidades de que esto ocurra.

8.6.1 Fuentes de información

Hay gran variedad de agentes que pueden intervenir en la fase de recogida de información. Esta etapa cuenta por ejemplo con registros de auditoría, agentes, o registros de eventos de sistema.

Algunas situaciones pueden dificultar la obtención de datos, como por ejemplo el uso de comunicaciones cifradas, o un entorno conmutado.

Utilizar protocolos de cifrado (como SSL, SSH o IPSec) anula las posibilidades de interpretación de los datos pertenecientes una comunicación.

Los conmutadores, con sus capacidades de enrutamiento, son otro de los factores que también reducen las posibilidades de monitorización. Esto se puede solventar en parte utilizando conmutadores que tengan unos puertos especiales denominados "spanning ports"¹ (puertos de extensión o abarcadores). Sin embargo la arquitectura de los conmutadores modernos también impide hacer uso de esta opción, ya la suma del ancho de banda producida por varios puertos puede sobrepasar las capacidades de uno solo. Esto una limitación física impuesta por los propios conmutadores.

Otra de las formas de solventar este problema es mediante el uso de "network taps" o cables de red de sólo recepción. En ambos casos, el monitor de red se acopla a un tramo de red, y sólo intercepta el tráfico que pasa por su sección.

Cuando la fuente de datos consiste en algún elemento de un sistema, como sus registros de auditoría, hay que tener en cuenta que se pueden alcanzar posibles situaciones sobrecarga de proceso, ataques, o incluso la recogida de datos falsos, que pueden hacer inservibles los resultados.

8.6.2 Análisis

La estabilidad y fiabilidad de los elementos que toman parte en la fase de análisis están íntimamente relacionadas con el número de recursos del sistema.

Un motor de análisis que reciba grandes cantidades de información deberá contar con suficiente capacidad de proceso para llevar a cabo su tarea. Este problema se agudiza en los detectores de anomalías. Por otra parte, los detectores de usos indebidos tampoco se escapan a este tipo de problemas. Si el espacio asignado para la base de datos de ataques no es lo suficientemente

¹ Este tipo de puertos se programan para recibir una copia del tráfico dirigido a otros puertos.

grande no se podrán estudiar todos los tipos de intrusiones. Además, la aparición de nuevas formas de ataque hace que las bases de datos sean cada vez mayores, así como las necesidades de almacenamiento y tiempo de proceso.

Los detectores de intrusiones de red pueden llegar a descartar paquetes, si no pueden examinar todo el tráfico. Para evitar esta situación, en redes de alta velocidad hay que asegurarse de que los recursos de que dispone el analizador son suficientes.

Otro de los elementos que podrían afectar a la fiabilidad del analizador es que se convirtiera en el objetivo de algún intruso. Existen además, ataques diseñados para sortear las barreras de detección, como los basados en "polymorphic shells" (interfaces de comandos polimórficas). Esta idea, tomada de los programadores de virus, permite cambiar el aspecto del código involucrado en el ataque mediante técnicas de cifrado.

Una de las formas de solventar problemas de sobrecarga y seguridad implica instalar el detector de intrusiones en una máquina separada del objetivo a monitorizar, en casos de detección de "host". Por otra parte, en las situaciones en las que hay mucho tráfico de red, se podría instalar más de un detector basado en red. Dividiendo las tareas de monitorización se reduciría la carga de cada analizador.

8.6.3 Respuesta

La pérdida de estabilidad de los mecanismos de respuesta también puede tener consecuencias graves. A continuación se describen algunas situaciones que lo demuestran.

Si un atacante lograra interceptar o bloquear las alarmas o informes, el sistema de detección entero carecería de valor. Interceptar los mensajes, puede poner al atacante sobre aviso e incluso permitir la modificación del contenido de los mismos. Impedir que las alarmas lleguen al responsable de seguridad anularía toda la efectividad del detector.

Otro escenario distinto es el relativo a las respuestas automáticas. Este tipo de respuestas, comentado en el capítulo 3 "Modelo de funcionamiento", hace que el sistema reaccione de forma activa modificando el entorno. Naturalmente, si este método no cuenta con las apropiadas medidas de seguridad, puede convertirse en un instrumento peligroso en manos de un intruso. Por ejemplo, en el caso de que el detector bloquee automáticamente aquellas direcciones IP relacionadas con un ataque. Sin un intruso logra determinar esto, puede alterar su dirección de origen, y utilizar direcciones que pueden ser fundamentales para el buen funcionamiento de la organización; desde direcciones de clientes, hasta direcciones de la propia red local privada de la infraestructura.

Si los mecanismos de respuesta no proporcionan los requisitos de seguridad necesarios, llegando a ser comprometidos, sus resultados no podrán utilizarse como pruebas en procesos legales.

El uso de mecanismos de encriptación por parte de los sistemas de respuesta es uno de los elementos clave para mejorar significativamente de la fiabilidad de estos sistemas.

8.6.4 Comunicaciones

La fiabilidad de los elementos nombrados anteriormente depende también de los medios que utilicen para comunicarse.

Los mecanismos que participan en el proceso de detección necesitan comunicarse entre ellos para poder trabajar. Por ejemplo, un detector de máquina puede obtener los registros de sistema a través de una conexión de red mediante el Protocolo Syslog. Otro escenario típico es aquel en que un detector de red debe recibir datos procedentes de los diversos sensores de la infraestructura.

Dado su valor, los enlaces de comunicación deben contar con suficientes medidas de protección para evitar ser vulnerados. Algunos intrusos pueden inhabilitar las líneas de comunicación, o interceptar las comunicaciones.

Una de las formas de mejorar la seguridad y fiabilidad de las comunicaciones consiste en utilizar protocolos de cifrado (SSL, IPSec,...), especialmente cuando se trata de comunicaciones inalámbricas. De esta forma, aunque un atacante intercepte la comunicación, no será capaz de interpretarla, ni podrá alterar su contenido sin que las entidades conectadas se percaten de ello. En estos casos, un atacante puede percatarse de que sus actividades está generando mensajes (aunque estén cifrados) y podría alarmarse. Hay formas de evitar esto, como enviar mensajes en intervalos aleatorios de tiempo.

Otro de los métodos practicados para incrementar las posibilidades de recibir una alarma es el uso de la redundancia. Esta es una solución especialmente recomendable cuando el detector emite alarmas importantes. A veces se establecen varios canales de comunicación redundantes, para ir cambiando de uno a otro de forma aleatoria. En otras ocasiones, la alarma se envía a través de diferentes medios de comunicación, como por ejemplo: un correo electrónico, un mensaje a un teléfono móvil, y la adición de la alarma en el registro de eventos.

8.7 Interfaz de usuario

Las características de la interfaz ofrecida por el detector de intrusiones, para que el usuario interactúe con él, son otro apartado no menos importante que los ya mencionados.

La interfaz de usuario puede hacer que la tarea de administrar un sistema de detección resulte una sencilla y cómoda, o casi imposible. Esta es una de las razones para no escoger un determinado producto.

Hay muchas formas de representar los datos obtenidos por un detector de intrusiones, y tener deficiencias en este aspecto puede tener graves consecuencias en la detección de posibles ataques. El tipo de detector utilizado es uno de los factores que influyen en la forma de gestionar y visualizar los datos. Un detector de anomalías, por ejemplo, representa sus resultados en forma estadística, de forma distinta que un detector de usos indebidos, que puede mostrar una lista de los ataques identificados.

No todos los usuarios tienen las mismas necesidades de monitorización. Además, una interfaz debería ser lo suficientemente flexible para adaptarse a las necesidades tanto de usuarios noveles como expertos, y aportar información que indique cómo reaccionar ante determinados

ataques. Por lo tanto, el hecho de proporcionar amplias capacidades de personalización, mejora las posibilidades de gestión e identificación de ataques.

8.8 Referencias

- [1] Bace, R. *Intrusion Detection*. Macmillan Technical Publishing, 2000.
- [2] Common Intrusion Detection Framework (CIDF). [en línea]. Actualizado en septiembre 1999 [consultado en mayo, 2003]. Disponible en <<http://www.isi.edu/gost/cidf/>>.
- [3] Critical Resource Allocation and Intrusion Response for Survivable Information Systems (CRISIS). [en línea] 1997 [consultado en mayo, 2003]. Disponible en <<http://www.isi.edu/~brian/crisis/>>.
- [4] National Computer Security Center. *Department of Defense Trusted Computer System Evaluation Criteria*. Orange Book, DOD 5200.28-std, December 1985.
- [5] National Computer Security Center. *A Guide to Understanding audit in Trusted Systems*. Versión 2, June 1988.
- [6] Denning, Dorothy E. *An Intrusion Detection Model*. Proceedings of the 1986 IEEE Symposium on Security and Privacy, Oakland, CA, April 1986.
- [7] Smaha, Stephen E. *A Common Audit Trail Interchange Format for UNIX*. Technical report, Haystack Laboratories, Inc. Austin, Tx, October 1994.
- [8] Bishop, Matt. *A Standard Audit Log Format*. Proceedings of the 1995 National Information Systems Security Conference, Baltimore, MD, October 1995.
- [9] Wood, M. *Intrusion Detection Message Exchange Requirements*. Internet draft, Internet Engineering Task Force, April, 2003.

Capítulo 9

Futuro

*"Me interesa el futuro porque es el sitio donde
voy a pasar el resto de mi vida."
Woody Allen*

Especular sobre el futuro de una tecnología relativamente joven y dinámica como es el caso de los Sistemas de Detección de Intrusiones es una tarea arriesgada. En este capítulo se reflexionará sobre el posible camino que seguirán estos sistemas, teniendo en cuenta los avances tecnológicos realizados hasta ahora, las necesidades más importantes, y las principales líneas de desarrollo existentes.

9.1 Trayectoria recorrida

Las primeras investigaciones relativas a la detección de intrusiones se remontan a principios de los años ochenta. Por aquella época, los únicos estudios realizados sobre el tema se referían a la detección basada en máquina. A partir de los años noventa, y tras el creciente uso de las redes informáticas, se desarrollaron los primeros detectores basados en red, que eran poco más que rastreadores de tráfico de red. Con el paso del tiempo, los detectores de red han ido adquiriendo más importancia, hasta el punto de ser más utilizados que los de "host".

El desarrollo de la tecnología ha ayudado en cierta medida a estos sistemas, haciendo posible la aparición de detectores de intrusiones en tiempo real, y capacidad de actualizar las bases de datos de ataques nada más publicarse nuevos patrones.

Actualmente existe la posibilidad de implementar un IDS distribuido, con un servidor central que recoge información de varios agentes situados en puntos estratégicos de la infraestructura de red.

Por otra parte, se están empezando a combinar técnicas de cortafuegos e IDS, dando lugar a productos como los Sistemas de Prevención de Intrusiones (IPS), capaces de impedir la evolución de un ataque antes de que ocurra. Estos sistemas son considerados por muchos expertos en seguridad como la siguiente generación de los IDSs.

Todos estos acontecimientos no hacen más que demostrar que los IDSs juegan un papel cada vez más relevante en la defensa de una infraestructura informática.

9.2 Perspectivas de futuro

A pesar de todos los avances realizados, estos sistemas no tienen el nivel de madurez deseado. La mayoría comparte una serie de aspectos pendientes de ser mejorados.

9.2.1 Deficiencias y necesidades

Los aspectos de la industria de los IDSs que más trabajo necesitan pueden arrojar algo de luz sobre el probable futuro de estos sistemas. A continuación se reflexiona sobre el futuro de los IDSs haciendo un recorrido a través de sus principales puntos débiles y necesidades.

9.2.1.1 Falsos positivos

Es imposible tratar sobre las deficiencias de los IDSs y no mencionar el problema relativo a los falsos positivos. Los falsos positivos (errores de Tipo I) son aquellas alarmas que emite el detector cuando identifica erróneamente un ataque o intrusión. El problema llega cuando el número de este tipo de alarmas es inaceptablemente alto. Este inconveniente se agrava con los falsos negativos (errores de Tipo II), en los que el detector no reconoce un ataque cuando ha ocurrido.

Aunque existen numerosas soluciones que abordan el problema, aún queda mucho trabajo por hacer en este aspecto.

Es muy probable que este problema se vaya solucionando en el futuro de forma similar a como han hecho los desarrolladores de antivirus. Durante sus comienzos, los detectores de virus también tenían un importante número de falsas alarmas, y no identificaban todos los virus conocidos. En los últimos años, los productos de antivirus han mejorado sensiblemente, trabajando de forma desatendida, y actualizando automáticamente sus bases datos de virus.

9.2.1.1.1 Heurística

Actualmente es relativamente fácil sortear un detector basado en usos indebidos. Este tipo de detectores contiene una base de datos con patrones de ataques que comparan con la actividad que analizan. Un ataque conocido, modificado ligeramente, puede ser totalmente indetectable por un detector de este tipo.

La adopción de técnicas heurísticas, tal como han hecho los fabricantes de antivirus, es sólo una de las medidas que utilizarán los IDSs en el futuro para corregir esta situación.

9.2.1.2 Estandarización

Los productos de detección de intrusiones actuales no utilizan ningún estándar en diversos aspectos de su metodología. La adopción de estándares no sólo solucionaría problemas de compatibilidad y entendimiento entre los productos existentes; sino que permitiría a las empresas que se dedican al desarrollo de estos productos afrontar de forma conjunta un problema común.

Algunos de los aspectos en los que sería aplicable un estándar son: un protocolo de comunicación, un lenguaje genérico para la detección de alarmas, o una definición de un conjunto de reglas para personalizar el motor de detección.

Hay algunos organismos que han dedicado importantes esfuerzos en el ámbito de la estandarización y normalización de los IDSs. El proyecto CIDF [1] y el grupo IDWG de IETF [2] son tan sólo dos ejemplos de esto. Desgraciadamente, muchos de los fabricantes de detectores de intrusiones prefieren patentar sus propias soluciones antes que adoptar un estándar.

No cabe duda de que el uso de estándares es un paso fundamental en la mejora global de la detección de intrusiones, aunque el éxito de este factor dependa de su adopción por parte de los desarrolladores de estos sistemas.

9.2.1.3 Encriptación

Actualmente es difícil imaginarse un entorno seguro sin el uso de comunicaciones cifradas. Compartir el mismo medio físico para establecer diferentes comunicaciones obliga a utilizar algún tipo de algoritmo de cifrado para proteger la información. Esto es lo que hace en las VPNs o Redes Privadas Virtuales.

Aunque la encriptación en las comunicaciones es un obstáculo para el trabajo de los NIDS (IDS de red), hay formas de solventarlo. Una de las más recurridas consiste en la implantación de agentes en las máquinas que participan en la comunicación.

Por otra parte, será imprescindible el uso de técnicas de encriptación para establecer enlaces seguros entre los elementos de un IDS, como por ejemplo, entre los sensores y el sistema central. El avance y popularidad de los mecanismos de cifrado es imparable, y los IDSs que vendrán deberán estar preparados para esta situación.

9.2.1.4 Nuevos protocolos

Un aspecto a tener en cuenta por parte de los IDSs consiste en el reconocimiento y soporte de protocolos de nueva aparición. Uno de los ejemplos más claros de esta situación es el Protocolo Internet versión 6 (IPv6), sucesor del actual IPv4 utilizado ampliamente en Internet.

Si un IDS no es capaz de reconocer un nuevo protocolo, sus capacidades se verían anuladas ante un atacante que lo utilizara en sus actividades. En el caso de IPv6 esto es especialmente sensible, ya que sus características le permiten construir túneles sobre redes IPv4.

Es muy probable que los fabricantes de detectores de intrusiones tengan en cuenta esta característica en sus futuros productos.

9.2.1.5 Escalabilidad

Los primeros productos de detección sólo tenían que monitorizar los datos originados por una máquina, y posteriormente, el tráfico generado en una pequeña red local. Con el paso del tiempo las organizaciones se han ido haciendo cada vez mayores, así como su tráfico de red. Esto ha hecho que los desarrolladores tengan cada vez más en cuenta factores como la escalabilidad a la hora de diseñar sus detectores de intrusiones.

El número de alarmas generadas, así como el aumento de las necesidades de recursos son dos problemas derivados del crecimiento de los IDSs, que deberán ser tenidos en cuenta por los IDSs en un futuro inmediato.

9.2.1.5.1 Sobrecarga de alarmas

Cuando se despliega un IDS en una gran corporación puede llegar a generar miles de alarmas al día. Gestionarlas puede convertirse en una labor imposible si no se disponen de los medios adecuados. Aunque hay algunas soluciones al respecto, este aspecto aún necesita mejorarse.

9.2.1.5.2 Recursos

Con el aumento del tráfico de red también crecen las necesidades de recursos de sistema por parte de los detectores. Una de las soluciones que se están empezando a aplicar consiste en la fabricación de soluciones específicas basadas en hardware, que alivien en cierto modo la carga de proceso a la que están sometidos los sistemas de detección. Probablemente esta opción será bastante común en muchos de los productos que están por venir.

9.2.1.6 Detección de anomalías

La gran mayoría de los productos de detección de intrusiones han utilizado la detección de usos indebidos (comparación de patrones de ataques). Sin embargo, la detección de anomalías (mediante la creación y comparación de perfiles estadísticos de actividad), siempre ha recibido más aceptación en círculos académicos que prácticos o comerciales, salvo raras excepciones.

9.2.1.6.1 "Data mining"

Una de las técnicas más prometedoras en el ámbito de la detección de anomalías se denomina "data mining" (minería de datos) [3], descrita por muchos expertos como la sucesora de la estadística clásica.

Esta técnica permite elaborar sus propios modelos estadísticos de forma automática a partir de los datos analizados. Por el contrario, la estadística clásica hace uso de modelos conocidos, a veces no muy adecuados para aplicarlos a algunos comportamientos de sistema. La minería de datos identifica patrones de actividad sistema que puedan describir pautas de conducta. Estos patrones son luego aplicados por los motores de detección para encontrar signos de intrusiones.

Otra de las características de este método es que no sólo es aplicable a la detección de anomalías sino también a la de usos indebidos.

9.2.1.6.2 Otros algoritmos

Numerosas técnicas han sido aplicadas ya con espectaculares resultados en entornos de desarrollo. Algoritmos de inteligencia artificial, neuronales, genéticos, o los basados en el sistema inmune biológico, son tan sólo algunos de los métodos practicados.

Las ilimitadas posibilidades que ofrece la detección de anomalías la sitúan, sin lugar a dudas en un lugar muy importante en el prometedor porvenir de la industria de la detección de intrusiones.

9.2.2 Correlación, composición

Estos aspectos merecen un trato especial, de forma separada a los ya mencionados. A medida que pasa el tiempo, la *correlación* de los datos obtenidos de las fuentes de información para *componer* un nivel superior de abstracción, se confirma como uno de los elementos más relevantes en el futuro de los IDSs. [4]

El hecho de que los IDSs recojan datos de diversas fuentes es un concepto que se va a desarrollar hasta el punto de que en el futuro no se hablará de HIDS o NIDS. Ni siquiera será significativa la aparición de productos híbridos que combinen ambos tipos, como algunos ya existentes, tales como *Prelude*, de Yoann Vandoorselaere [5] o *Dragon Intrusion Detection System*, de Enterasys [6].

En la infraestructura típica de una organización coexisten diversos elementos que pueden aportar información sobre la actividad de un atacante o intruso. Algunos de estos dispositivos pueden ser: cortafuegos, "routers", servidores de correo [7], HTTP, DNS, agentes instalados en estaciones de trabajo, etc.

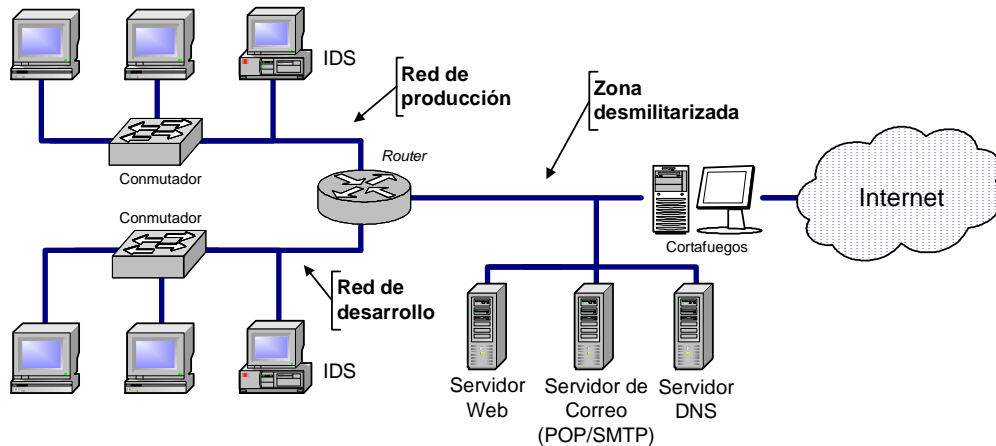


Figura 9-1 - Fuentes de información comunes

En el futuro los IDSs serán capaces de relacionar la información obtenida de gran parte de los elementos de una infraestructura para obtener una visión global de las actividades que se llevan a cabo.

El crecimiento de las comunicaciones cifradas y los entornos conmutados, hacen cada vez es más común la implementación de agentes, o sistemas de defensa en las máquinas. Aunque hay soluciones parciales a ambos problemas, los IDSs pretenden llegar más lejos con la correlación.

La instalación de dispositivos de monitorización y análisis en cada máquina permite concebir la idea de un sistema *central*, encargado básicamente de tareas de gestión y correlación de los datos recibidos a través de sus agentes. Esta forma de trabajo implica un enfoque totalmente distribuido en el diseño de los detectores.

Huelga decir que el esfuerzo requerido para combinar y obtener resultados eficaces de diversas fuentes de información es enorme. Hay que tener en cuenta numerosos factores como formatos de registro, sincronización, modelos de conducta, etc. Muchos de los algoritmos utilizados en la detección de anomalías tendrán un papel importante en el desarrollo de este aspecto. La adopción de algún estándar relativo a los protocolos utilizados, o el lenguaje de alarmas ayudará en gran medida a alcanzar este objetivo.

Aunque existen numerosas y variadas soluciones en la industria de la detección de intrusiones, lo más probable es que en un futuro a largo plazo estos sistemas reúnan en un sólo dispositivo todas las características de los elementos de seguridad actuales. Este elemento podría incluso coordinar la labor de diferentes IDSs. Desde dicho dispositivo, el administrador será gestionar de forma global y precisa la seguridad de su red. Por otra parte, una serie de elementos

distribuidos colaborarán con el sistema central, monitorizando la actividad, realizando tareas específicas, y enviando sus resultados.

9.3 Referencias

- [1] Common Intrusion Detection Framework (CIDF). [en línea]. Actualizado en septiembre 1999 [consultado en mayo, 2003]. Disponible en <<http://www.isi.edu/gost/cidf/>>.
- [2] Intrusion Detection Exchange Format (idwg). Internet Engineering Task Force. [en línea]. Julio, 1990 [consultado en abril, 2003]. Disponible en <<http://www.ietf.org/html.charters/idwg-charter.html>>.
- [3] Goeldenitz, Thomas. SANS. *IDS – Today and Tomorrow*. [en línea] 22 de enero de 2002 [consultado en abril, 2003]. Disponible en <<https://www.sans.org/rr/intrusion/today.php>>.
- [4] Tanase, Matthew. SecurityFocus. *The Future of IDS*. [en línea]. Última actualización el 4 de diciembre de 2001 [consultado en abril, 2003]. Disponible en <<http://www.securityfocus.com/infocus/1518>>.
- [5] Vandoorselaere, Yoann. *Prelude Hybrid IDS*. [en línea]. 1998 [consultado en abril, 2003]. Disponible en <<http://prelude-ids.org/>>.
- [6] Enterasys. *Dragon Intrusion Detection System*. [en línea]. Fecha no disponible [consultado en abril, 2003]. Disponible en <<http://www.enterasys.com/products/ids/>>.
- [7] McAnderson, Brenda y Paul Ramstedt. *Intrusion Detection Technology: Today and Tomorrow*. [en línea] 18 de noviembre de 1999 [consultado en abril, 2003]. Disponible en <<http://www.first.org/events/progconf/2000/D3-03.pdf>>.

Conclusiones

Las posibilidades y capacidades de las relativamente jóvenes tecnologías de detección de intrusiones no han pasado desapercibidas, motivando a numerosos profesionales y empresas de desarrollo. Se han podido observar las condiciones en que nacieron los Sistemas de Detección de Intrusiones, y las sorprendentes mejoras que han experimentado desde entonces.

La enorme variedad de grupos y propuestas de trabajo sobre las tecnologías de detección de intrusiones no hacen más que confirmar la importancia de las mismas. Muchas empresas de seguridad han ratificado este hecho, a través de importantes inversiones realizadas para desarrollar sus propias soluciones basadas en "software" o "hardware".

Uno de los aspectos que más pueden cambiar el estado de la industria de la detección de intrusiones será probablemente el relativo a la formulación y aplicación de estándares. No se puede saber con certeza cuándo se llegará a un consenso en el uso de un protocolo de comunicaciones, o lenguaje de alarmas comunes. No cabe duda de que esto dependerá de las decisiones tomadas por las organizaciones responsables de los productos más importantes.

Las nuevas tecnologías se desarrollan a un ritmo vertiginoso, y los retos de seguridad se suceden a la misma velocidad. Los detectores de uso indebido pueden reconocer ataques conocidos, siendo posible la actualización de sus bases de ataques periódicamente de forma similar a como ya se hace con los antivirus. Los detectores de anomalía son una de las herramientas de seguridad más prometedoras, pudiendo detectar ataques no conocidos, valiéndose de gran variedad de métodos de análisis.

Por otra parte, es importante recalcar que los IDSs no han sido desarrollados para sustituir a ninguna de las soluciones de seguridad existentes; no son la panacea. Como ya se ha apuntado, su labor consiste principalmente en reducir la carga de trabajo de los responsables de seguridad, realizando diversos análisis de los datos disponibles. Estos sistemas complementan a los elementos actuales y fortalecen la seguridad global de cualquier infraestructura.

Con el avance de las tecnologías informáticas, el apartado de la seguridad ha ido cobrando cada vez más valor, hasta convertirse en uno de los aspectos más relevantes de esta era. La detección de intrusiones, en concreto, reúne muchos de los elementos necesarios para convertirse en el pilar fundamental del futuro de la seguridad. Su capacidad de análisis y correlación de los datos obtenidos de múltiples fuentes, posiblemente hará de esta tecnología una de las más importantes del sector.

Apéndice A - Glosario de términos anglosajones

Access control: Control de acceso.

Access Control List: Lista de Control de Acceso (ACL).

Accountability: Capacidad de ser registrado.

Accounting: Contabilidad, auditoría.

ACL: Véase ("Access Control List").

Active response: Respuesta activa.

Add-on: Añadido.

Address: Dirección.

Agent: Agente. Véase también ("sensor").

AI: Véase ("Artificial Intelligence").

Anomaly: Anomalía.

Anomaly detection: Detección de anomalías.

Anonymity: Anonimato, anonimía.

API: Véase ("Application Program Interface").

Application based: Basado en aplicación.

Application log: Registro de aplicación.

Application Program Interface: Interfaz de programación de aplicaciones (API).

Archive: Archivo, (al ser muy usado en la traducción de "file", puede ser conveniente aclarar el tipo de archivo referido).

Array: Formación, estructura, matriz, vector, ("arreglo" en América Latina).

Artificial Intelligence: Inteligencia artificial (AI).

Assesment system: Véase ("Vulnerability Scanner").

Assessment: Evaluación, estimación.

Attribute: Atributo.

Audit: Auditoría.

Audit class: Clases de auditoría.

Audit flag: Indicador de auditoría.

Audit reduction: Reducción de auditoría.

Audit trail: Rastro o registro de auditoría.

Authentication: Autenticación, autentificación.

Authorization: Autorización.

Availability: Disponibilidad.

Back door: Puerta trasera. Véase también ("vulnerabilidad").

Backbone: Eje principal, red troncal, estructura principal.

Backup: Copia de seguridad, copia de respaldo.

Bandwidth: Amplitud de banda, ancho de banda.

Basic Security Module: Módulo de Seguridad Básico (BSM).

Batch: Lote.

Batch mode analysis: Análisis en modo por lotes.

Batch processing: Procesamiento por lotes, procesamiento en lotes.

Berkeley Packet Filter: Filtro de paquetes Berkeley (BPF).

Binary Log Format: Formato de registro binario.

BPF: Véase ("Berkeley Packet Filter").

Bridge: Puente. Véase también ("conmutador").

BSM: Véase ("Basic Security Module").

Buffer: Búfer, memoria tampón, memoria intermedia.

Buffer Overflow: Desbordamiento de búfer, desbordamiento de la pila.

Bug: Error, fallo, gazapo.

Byte: Byte, octeto.

Cache: Almacén, ante-memoria, depósito.

CGI: Véase ("Common Gateway Interface").

Checksum: Suma de control, suma de verificación, suma de comprobación.

CIDF: Véase ("Common Intrusion Detection Framework").

Cluster: Grupo, cúmulo.

Clusterig analysis: Análisis por grupos, agrupado.

CMU/Stanford Packet Filter: Filtro de paquetes CMU/Stanford.

Colored Petri Net: Petri Net Coloreada (CP-net).

Common Gateway Interface: Interfaz Común de Acceso (CGI).

Common Intrusion Detection Framework: Marco de Detección de Intrusiones Común. (CIDF).

Composition: Composición.

Compromised: Comprometido, violentado.

Confidentiality: Confidencialidad.

Correlation: Correlación.

Coverage: Cobertura, alcance.

CP-net: Véase ("Colored Petri Net").

Crack (v.): Invadir, penetrar.

Credentialed analysis: Análisis con acreditaciones.

DAC: Véase ("Discretionary Access Control").

Data Mining: Minería de datos.

Data reduction: Reducción de datos. Véase ("reducción de auditoría").

Datagram: Datagrama.

DDoS: Véase ("Distributed Denial of Service").

Deceptive application: Aplicación engañosa.

Decoy server: Servidor señuelo, o servidor trampa. Véase ("honeypot").

De-Militarized Zone: Zona desmilitarizada, red perimétrica (DMZ).

Denial of Service: Denegación de servicio (DoS). Véase también ("Denegación de servicio distribuida").

Deterministic: Determinista.

Discretionary Access Control: Control de acceso discrecional (DAC). Véase también ("Control de accesos obligatorio").

Distributed Denial of Service: Denegación de servicio distribuida (DDoS). Véase también. ("Denegación de servicio").

DMZ: Véase ("De-Militarized Zone").

DNS: Véase ("Domain Name System").

Domain Name System: Sistema de Nombres de Dominio (DNS).

DoS: Véase ("Denial of Service").

Dynamic analysis: Análisis dinámico, o en tiempo real.

EDP audit: Véase ("Electronic Data Process" audit).

Electronic Data Process audit: Auditoría mediante proceso electrónico de datos.

Encryption: Cifrado.

Event: Evento, suceso.

Event horizon: Horizonte de evento.

Event log: Registro de eventos.

Expert system: Sistema experto.

Exploit: Ardid, artificio.

External penetrators: Intrusos desde el exterior.

False negative: Falso negativo.

False positive: Falso positivo.

File: Fichero, archivo.

File Transfer Protocol: Protocolo de Transferencia de Ficheros (FTP).

Fingerprint: Huella dactilar, huella digital.

Firewall: Cortafuegos.

Flag: Indicador.

Free Software: Software libre. Véase también ("código abierto").

FTP: Véase ("File Transfer Protocol").

Fuzzy data mining: Minería de datos difusa.

Fuzzy logic: Lógica difusa.

Gateway: Pasarela, puerta de enlace.

Group Identifier: Identificador de grupo (GID).

Guards: Guardias.

Hash Function: Función resumen.

HID: Véase ("Host based Intrusion Detection").

Honeynet: Red trampa, red de miel.

Honeypot: Sistema trampa, tarro de miel.

Host: Anfitrión, máquina anfitriona, puesto.

Host authentication: Autenticación por máquina.

Host based: Basado en máquina.

HTTP: Véase ("Hypertext Transfer Protocol").

Hub: Concentrador. Véase también ("repetidor").

Hypertext Transfer Protocol: Protocolo de Transferencia de Hipertexto (HTTP). Véase. también ("WWW").

Identification and authentication: Identificación y autenticación (I&A).

IDES: Véase ("Intrusion Detection Expert System").

IDS: Véase ("Intrusion Detection System").

IETF: Véase ("Internet Engineering Task Force").

Information Retrieval: Recuperación de información.

In-line mode: Modo en línea.

Inode: Inodo, nodo i.

Integration: Integración.

Integrity: Integridad.

Integrity checker: Comprobador de integridad.

Interface: Interfaz (fem.).

Internal penetrators: Intrusos desde el interior.

Internet: Internet.

Internet Engineering Task Force: Grupo de Trabajo de Ingeniería de Internet (IETF).

Internet Protocol Security: Seguridad de Protocolo Internet (IPSec).

Internet Protocol version 6: Protocolo Internet versión 6.

Interoperability: Interoperabilidad.

Interval based analysis: Análisis basado en intervalo.

Intrusion: Intrusión.

Intrusion detection: Detección de intrusiones.

Intrusion Detection Expert System: Sistema Experto de Detección de Intrusiones (IDES).. Véase también "Sistema Experto de detección de intrusiones de siguiente generación (NIDES)".

Intrusion Detection System: Sistema de detección de intrusiones (IDS).

Intrusion Prevention System: Sistema de prevención de intrusiones (IPS).

Intrusive monitoring: Monitorización intrusa.

IPS: Véase ("Intrusion Prevention System").

IPsec: Véase ("Internet Protocol Security").

IPv6: Véase ("Internet Protocol version 6").

Isolation: Aislamiento.

Log: Registro, historial.

Logger: Gestor de registro de actividades.

MAC: Véase ("Mandatory Access Control").

Mainframe: Gran ordenador, servidor corporativo, ordenador central, macrocomputadora.

Mandatory Access Control: Control de accesos obligatorio (MAC). Véase también ("Control de acceso discrecional").

Man-in-the-middle attack: Ataque por interceptación.

Masquerade: Enmascaramiento, falseamiento de identidad.

Masquerader: Enmascarado.

Masquerading: Enmascaramiento, mimetización.

Message digest: Resumen de mensaje. Véase ("función resumen").

Misuse: Uso indebido.

Misuse Detection: Detección de usos indebidos.

Monitor: Monitor, monitorizar.

Monitoring policy: Política de monitorización.

Multihost: Multi-máquina.

Multi-host based: Basado en multi-máquina. Véase también ("basado en máquina").

National Center for Supercomputing Alliances: Centro Nacional de Alianzas de Superordenadores (NCSA).

NCSA: Véase ("National Center for Supercomputing Alliances").

Network based: Basado en red.

Network hop: Salto de red.

Network management: Gestión de redes.

Network Node Intrusion Detector: Detector de Intrusiones de Nodo de Red.

Network Tap: Dispositivo de escucha de red.

Network Time Protocol: Protocolo de Tiempo de Red (NTP).

Next-Generation Intrusion Detection Expert System : Sistema Experto de detección de intrusiones de siguiente generación (NIDES). Véase también "Sistema Experto de Detección de Intrusiones (IDES)".

NID: Véase ("Network Intrusion Detection").

NIDES: Véase ("Next-Generation Intrusion Detection Expert System").

NNID: Véase ("Network Node Intrusion Detection").

Noncredentialed analysis: Análisis sin acreditaciones.

Nonintrusive monitoring: Monitorización no intrusa.

Nonparametric: No paramétrico.

NTP: Véase ("Network Time Protocol").

Open Source: Código abierto. Véase también ("software libre").

Open Systems Interconnection: Interconexión de Sistemas Abiertos (OSI).

Operating system audit trails: Registros de auditoría de sistema operativo.

Orange Book: Libro Naranja. Véase ("Trusted Computer System Evaluation Criteria (TCSEC)").

OS fingerprinting: Identificación de Sistema Operativo.

OSI: Véase ("Open Systems Interconnection").

Packet: Paquete.

Padded cell: Célula de aislamiento, célula acolchada.

Passive response: Respuesta pasiva.

Password: Clave, contraseña.

Password cracker: Rompedor de contraseñas.

Patch: Parche.

Performance management: Gestión de rendimiento.

Plugin: Accesorio, añadido, módulo.

Polymorphic shell: Interfaz de comandos polimórfica.

Polymorphic virus: Virus polimórfico.

Portscan: Sondeo de puertos, escaneo de puertos. Véase también ("escaneo sigiloso de puertos").

Predictive Pattern Generation: Generación de Patrones Probables.

Privacy: Privacidad.

Privilege: Privilegio.

Promiscuous mode: Modo promiscuo.

Protocol anomaly filter: Filtro de anomalías de protocolo. Véase también ("filtro de anomalías estadísticas").

Protocol stack: Pila de protocolos.

Race condition: Condición de carrera.

Rainbow serie: Serie Arco Iris. Véase también ("Libro Naranja" y "Libro Marrón").

Real-time analysis: Análisis en tiempo real.

Record: Informe, historial.

Repeater: Repetidor. Véase también ("concentrador").

Router: Encaminador, enrutador.

Rule based: Basado en reglas.

Rulebase: Base de reglas.

Scalability: Escalabilidad.

Script: Guión.

Secure Shell: Interfaz de comandos segura (SSH).

Secure Socket Layer: Capa de Conexión Segura (SSL).

Security: Seguridad.

Security log: Registro de seguridad.

Security management: Gestión de seguridad.

Security policy: Política de seguridad.

Segment: Segmento.

Self-contained: Auto contenido.

Self-healing: Auto curación.

Sensor: Sensor. Véase también ("agente").

Server Message Block: Bloque de mensajes de servidor (SMB).

Session creep: Deslizamiento sigiloso de sesión.

Shadow Password: Contraseña oculta.

Shell: Interfaz de comandos.

Shunning: Rechazo, esquivamiento. Véase también ("respuesta activa").

Signature: Firma, patrón.

Simple Mail Transfer Protocol: Protocolo Simple de Transferencia de Correo.

SMB: Véase ("Server Message Block").

SMS: Véase ("Systems Management Server").

SMTP: Véase ("Simple Mail Transfer Protocol").

Sniffer: Rastreador.

Sniffing cable: Cable de rastreo, cable de sólo recepción.

Spanning port: Puerto de extensión, puerto abarcador.

Spoofing: Falseamiento, enmascaramiento.

SSH: Véase ("Secure Shell").

SSL: Véase ("Secure Socket Layer").

Stack: Pila.

Stack smashing: Desbordamiento de la pila. Véase también ("pila", "desbordamiento de búfer").

State transition: Transición de estado.

Static analysis: Análisis estático.

Statistic Anomaly Filter: Filtro de anomalías estadísticas.

Stealth port scan: Escaneo sigiloso de puertos. Véase también ("escaneo de puertos").

Steganography: Esteganografía.

Stream: Corriente, flujo.

STREAMs: STREAMs.

Subnet: Subred.

Switch: Conmutador. Véase también ("puente").

Switched enviroment: Entorno conmutado.

System log: Registro de sistema.

Systems Management Server: Servidor de Gestión de Sistemas (SMS).

Tan Book ("A Guide to Understanding Audit in Trusted Systems"): Libro Marrón ("Guía. para la Comprensión de la Auditoría en Sistemas de Confianza").

Tap mode: Modo de escucha.

Target based: Basado en objetivo.

TCP/IP: Véase ("Transmission Control Protocol / Internet Protocol").

TCSEC: Véase ("Trusted Computer System Evaluation Criteria").

Testing by exploit: Probar mediante explotación. Véase también ("analizador de. vulnerabilidades").

Thread: Hebra, hilo (de mensajes, o de ejecución), flujo de control o flujo de ejecución.

Threat: Amenaza.

Threshold: Umbral.

Token: Testigo.

Token authentication: Autenticación por testigo.

Token based: Basado en testigo.

Trail: Rastro, registro.

Training: Entrenamiento.

Transceiver: Transmisor-receptor.

Transmission Control Protocol / Internet Protocol: Protocolo de Control de Transmisión /. Protocolo Internet (TCP/IP).

Trigger: Disparador.

Trojan Horse: Caballo de Troya, troyano. Véase también ("puerta trasera").

Trust: Confianza.

Trusted Computer System Evaluation Criteria: Criterio de Evaluación de Sistemas. Informáticos Fiables (TCSEC).

Trusted processes: Procesos de confianza.

Trusted systems: Sistemas de confianza.

Type I error: Error de Tipo I. Véase también ("falso positivo").

Type II error: Error de Tipo II. Véase también ("falso negativo").

User-agent: Agente de usuario.

Virtual Private Network: Red Privada Virtual (VPN).

VPN: Véase ("Virtual Private Network").

Vulnerabilities: Vulnerabilidades.

Vulnerability analysis: Análisis de vulnerabilidades.

Vulnerability scanner: Escáner o analizador de vulnerabilidades.

Web: Malla, telaraña. Véase también ("WWW").

Wireless: Inalámbrico.

World Wide Web: Malla mundial, telaraña mundial. Véase también ("web").

Worm: Gusano.

Wrapper: Envoltura, forro, empacador.

WWW: Véase ("World Wide Web").

Apéndice B - Glosario

Agente. Véase también ("sensor"): En detección de intrusión, una entidad independiente que realiza labores de monitorización y análisis de bajo nivel y envía sus resultados a un coordinador o un transmisor-receptor. También conocido como sensor.

Almacén, ante-memoria, depósito: Mecanismo especial de almacenamiento de alta velocidad. Puede ser una zona reservada de la memoria principal, o un dispositivo independiente de almacenamiento de alta velocidad.

Amenaza: Situación o evento con que puede provocar daños en un sistema.

Amplitud de banda, ancho de banda: 1. Diferencia en hertzios (Hz) entre la frecuencia más alta y la más baja de un canal de transmisión. 2. Datos que puede ser enviados en un periodo de tiempo determinado a través de un circuito de comunicación. Se mide en bits por segundo (bps).

Anomalía: No usual o estadísticamente raro.

Anonimato, anonimia: Carácter o condición de anónimo (desconocimiento del nombre o identidad).

Análisis basado en intervalo: Análisis desarrollado de forma discontinua. Se aplica en casos de recopilación no continua de datos, y en casos de recopilación continua pero análisis discontinuo. También se denomina análisis en modo por lotes, o estático.

Análisis con acreditaciones: En análisis de vulnerabilidades, enfoque de monitorización pasiva en los que son necesarias contraseñas u otro tipo de credenciales. Normalmente implica el acceso a los datos de un objeto de sistema.

Análisis de vulnerabilidades: Análisis del estado de la seguridad de un sistema o sus componentes mediante el envío de pruebas y recogida de resultados en intervalos.

Análisis dinámico, o en tiempo real: Análisis desarrollado en tiempo real, o de forma continua.

Análisis en modo por lotes: Véase "análisis basado en intervalo".

Análisis en tiempo real: Análisis realizado de forma continua, con resultados obtenidos en un tiempo en que permita alterar el estado actual sistema.

Análisis estático: Análisis de información desarrollado de forma discontinua. También conocido como análisis basado en intervalo o en modo por lotes.

Análisis sin acreditaciones: En análisis de vulnerabilidades, enfoque de monitorización pasiva en los que las contraseñas u otro tipo de credenciales no son necesarias. Normalmente implica el lanzamiento de ataques contra el sistema, provocando algún tipo de reacción.

Aplicación engañosa: Aplicación cuya apariencia y comportamiento emulan a una aplicación real. Normalmente se utiliza para monitorizar acciones realizadas por atacantes o intrusos.

Ardid, artificio: Implementación de un fallo de seguridad, utilizado bien para comprobar y demostrar la existencia del fallo, o bien para comprometer el sistema de forma ilícita.

Ataque por interceptación: Estrategia de ataque en la que el atacante intercepta una comunicación entre dos partes, substituyendo el tráfico entre ambas a voluntad y controlando la comunicación.

Auditoría mediante proceso electrónico de datos: Evolución de los tradicionales procesos y prácticas de auditoría, utilizando sistemas de proceso de datos.

Auditoría: Proceso de examinar y revisar un informe cronológico de los eventos de sistema para determinar su significado y valor.

Autenticación, autentificación: Proceso de confirmar la identidad de una entidad de sistema (un usuario, un proceso, etc.).

Auto contenido: Historiales de eventos de sistema que no necesitan de otros historiales para su interpretación.

Autorización: Acción de otorgar el acceso a usuarios, objetos o procesos.

Basado en aplicación: Se utiliza para describir monitores que recogen datos a partir de aplicaciones. Las fuentes de datos pueden ser registros de eventos u otro tipo de información perteneciente a aplicaciones.

Basado en multi-máquina. Véase también ("basado en máquina"): Que monitoriza información de fuentes internas a múltiples máquinas.

Basado en máquina: Que monitoriza información de fuentes internas a una máquina.

Basado en objetivo: Que monitoriza información de determinados objetos, generalmente utilizando métodos de cifrado como funciones resumen para permitir la detección de cambios.

Basado en red: Que monitoriza información de fuentes de red, generalmente captura de paquetes.

Basado en reglas: En detección de intrusión, que utiliza patrones de actividad (generalmente ataques conocidos) para reconocer una intrusión.

Basado en testigo: Sistemas que emplean elementos especiales como tarjetas inteligentes, llaves, o discos para la autenticación de usuario.

Base de reglas: Conjunto de reglas utilizadas para analizar los registros de datos.

Bit. Véase también "byte": Abreviación de "binary digit". Unidad elemental de información en un sistema informático. Tiene un único valor en formato binario: "0" ó "1".

Bloque de mensajes de servidor (SMB): También conocido como "Session Message Block", NetBIOS y LanManager. Es un protocolo utilizado por sistemas Windows para compartir ficheros, impresoras, puertos serie y otras entidades de comunicación entre ordenadores.

Byte, octeto: Unidad de información compuesta por ocho bits. Modificando los diferentes bits de un byte se pueden obtener hasta 256 combinaciones diferentes.

Búfer, memoria tampón, memoria intermedia: Área de memoria de un sistema reservada para almacenar información de forma temporal. Generalmente se utiliza para compensar las diferencias de velocidad surgidas entre varias señales o procesos.

Caballo de Troya, troyano. Véase también ("puerta trasera"): Programa informático de aspecto inofensivo que oculta en su interior un código que permite abrir una "puerta trasera" en el sistema en que se ejecuta.

Cable de rastreo, cable de sólo recepción: Cable de red modificado para imposibilitar el envío de datos, permitiendo exclusivamente su recepción.

Capa de Conexión Segura (SSL): Protocolo creado por Netscape para permitir la transmisión cifrada y segura de información a través de la red.

Capacidad de ser registrado: Habilidad de relacionar una determinada actividad o evento con la parte responsable.

Cifrado: Proceso mediante el cual se toma un mensaje en claro, se le aplica una función matemática, y se obtiene un mensaje codificado.

Cobertura, alcance: Proporción de ataques conocidos que un detector de intrusiones es capaz de detectar.

Composición: 1. En detección de intrusiones, proceso de combinar información procedente de distintas fuentes en un flujo de datos coherente. 2. En seguridad informática, combinar un conjunto de componentes en un sistema para obtener los atributos de seguridad del sistema, según las propiedades de los componentes.

Comprobador de integridad: Herramienta de seguridad que utiliza funciones resumen basadas en algoritmos de cifrado para detectar alteraciones en objetos de sistema.

Comprometido, violentado: Estado de un equipo/sistema cuando un intruso ha entrado.

Concentrador. Véase también ("repetidor"): Dispositivo que permite la interconexión de las estaciones de trabajo entre sí. No realiza funciones de encaminamiento; lo que recibe por un puerto lo reenvía a través del resto.

Condición de carrera: Comportamiento anómalo provocado por una dependencia excesiva del tiempo relativo transcurrido entre diferentes eventos.

Confianza: Esperanza firme de que un sistema se comporte como corresponde.

Confidencialidad: Requisito de seguridad que indica que el acceso a los recursos de sistema debe estar limitado exclusivamente a los usuarios con acceso autorizado.

Conmutador. Véase también ("puente"): Elemento utilizado para interconectar máquinas a una red. Tiene funciones de encaminamiento básico de tráfico de red, y permite subdividir las redes en segmentos, de forma similar a un puente.

Control de acceso discrecional (DAC). Véase también ("Control de accesos obligatorio"): Política de acceso a los datos en la que el propietario del objeto, de forma voluntaria (discrecional), concede o deniega el acceso a éste a otros sujetos.

Control de acceso: Limitar el acceso a objetos de acuerdo a los permisos de acceso del sujeto. El control de acceso puede ser definido por el sistema (Control de accesos obligatorio, MAC) o por el propietario del objeto (Control de accesos discrecional, DAC).

Control de accesos obligatorio (MAC). Véase también ("Control de acceso discrecional"): Política de acceso a los datos en la que el sistema comparte de forma obligatoria tanto los objetos como los sujetos. A partir de dicha forma de compartir los elementos, se establecen unas reglas de acceso.

Correlación: En detección de intrusiones, relación que se establece entre diferentes fuentes de información.

Cortafuegos: Herramienta de seguridad que proporciona un límite entre redes de distinta confianza o nivel de seguridad mediante el uso de políticas de control de acceso de nivel de red.

Criterio de Evaluación de Sistemas Informáticos Fiables (TCSEC): Conocido comúnmente como Libro Naranja, describe las propiedades que deben cumplir los sistemas para contener información sensible o clasificada. Este criterio fue desarrollado por el Centro de Seguridad Informática Nacional (NCSC).

Célula de aislamiento, célula acolchada: Sistema o red consistente en una copia parcial de un sistema real, al que un dispositivo con capacidades de enrutamiento y detección de intrusiones redirige el tráfico hostil.

Código abierto. Véase también ("software libre"): Software que cumple los criterios descritos por la iniciativa "Open Source". Este término no implica el acceso al código fuente.

Datagrama: Mensaje que se envía en una red de comunicaciones de ordenadores por intercambio de paquetes.

Denegación de servicio (DoS). Véase también ("Denegación de servicio distribuida"): Estrategia de ataque que consiste en saturar de información a la víctima con información inútil para detener los servicios que ofrece.

Denegación de servicio distribuida (DDoS). Véase también ("Denegación de servicio"): Estrategia de ataque que coordina la acción de múltiples sistemas para saturar a la víctima con información inútil para detener los servicios que ofrece. Los sistemas utilizados para el ataque suelen haber sido previamente comprometidos, pasando a ser controlados por el atacante mediante un cliente DDoS.

Desbordamiento de búfer, desbordamiento de la pila: Técnica que consiste en almacenar más datos en un búfer de los que puede contener. Los datos que no caben pueden invadir zonas adyacentes a la del búfer, corrompiéndolas o sobrescribiéndolas. Este método es ampliamente utilizado para realizar ataques que abren interfaces de comando remotas.

Desbordamiento de la pila. Véase también ("pila", "desbordamiento de búfer"): Caso especial del desbordamiento de búfer, en el que el objetivo es la pila del sistema.

Deslizamiento sigiloso de sesión: Técnica utilizada por un usuario que consiste en modificar gradualmente su comportamiento para entrenar al detector de anomalías. De esta forma, se consigue que el detector diagnostique como actividad normal un posible ataque.

Detección de anomalías: Detección basada en la actividad de sistema que coincide con la definida como anormal.

Detección de intrusiones: Proceso de monitorizar los eventos de un sistema o red en busca de signos que indiquen problemas de seguridad.

Detección de usos indebidos: Detección basada en la actividad de sistema que coincide con la definida como mala.

Detector de Intrusiones de Nodo de Red: Detector de intrusiones basado en red que se instala en una máquina. Esta medida ayuda a solventar problemas como los asociados a entornos conmutados, o cifrado en las comunicaciones.

Determinista: Propiedad de los procesos que permite recorrer un proceso hacia delante o hacia atrás, desde cualquier punto del proceso.

Disponibilidad: Requisito de seguridad que implica que la información y los servicios del sistema continúen en funcionamiento y que los usuarios autorizados puedan acceder a los recursos cuando lo necesiten, dónde lo necesiten, y en la forma en que lo necesiten.

Dispositivo de escucha de red: Dispositivo, de aspecto externo similar a un concentrador o un conmutador, que permite a un rastreador interceptar el tráfico de red entre dos segmentos sin ser detectado. Además, apenas afecta al rendimiento de la red.

Encaminador, enrutador: Dispositivo que reenvía paquetes de datos entre redes. Permite conectar al menos dos redes. Los puntos de conexión con el "encaminador" son las puertas de enlace de cada red.

Enmascarado: Atacante que accede a un sistema utilizando identificadores de usuario y contraseñas de usuarios legítimos.

Entorno conmutado: Entorno de red en el que predomina el uso de conmutadores.

Envoltura, forro, empacador: Software que complementa las características de otro software para mejorar determinados aspectos como compatibilidad, o seguridad.

Error de Tipo I. Véase también ("falso positivo"): En detección de intrusiones, error producido cuando el sistema diagnostica como ataque una actividad normal. También conocido como falso positivo.

Error de Tipo II. Véase también ("falso negativo"): En detección de intrusiones, error producido cuando el sistema diagnostica como actividad normal un ataque. También conocido como falso negativo.

Escalabilidad: Forma en que la solución a un determinado problema se comporta cuando el tamaño del problema crece.

Escaneo sigiloso de puertos. Véase también ("escaneo de puertos"): Barrido de puertos mediante diversas técnicas con el fin de evadir los métodos de detección comunes. Algunas de estas técnicas implican un escaneo intencionadamente lento, o el envío de paquetes especiales aprovechando particularidades del protocolo.

Escáner o analizador de vulnerabilidades: Herramienta diseñada para llevar a cabo análisis de vulnerabilidades.

Esteganografía: Arte de transmitir información de modo que la presencia de la misma pase inadvertida. Se suele hacer camuflando los datos en el interior un texto, imagen, o fichero multimedia. Proviene de las palabras griegas *steganós* (cubierto) y *grapto* (escrito).

Ethernet: Sistema de red de área local de alta velocidad.

Falseamiento, enmascaramiento: Modificación de la identidad de origen real durante una comunicación. El método más común consiste en alterar directamente la dirección origen de cada paquete de la comunicación.

Falso negativo: En detección de intrusiones, error producido cuando el sistema diagnostica como ataque una actividad normal. También conocido como error de tipo I.

Falso positivo: En detección de intrusiones, error producido cuando el sistema diagnostica como actividad normal un ataque. También conocido como error de tipo II.

Filtro de anomalías de protocolo. Véase también ("filtro de anomalías estadísticas"): Tipo de filtro de anomalías estadísticas al que se le han añadido conocimientos sobre un protocolo determinado, para poder detectar usos poco comunes del mismo.

Filtro de anomalías estadísticas: Filtro que permite la detección de actividades y comportamientos poco usuales o comunes.

Filtro de paquetes Berkeley (BPF): Una arquitectura diseñada para la captura de paquetes, desarrollada en el "Lawrence Berkeley National Laboratory".

Firma, patrón: En detección de intrusión, patrones que indican los usos indebidos de un sistema.

Formato de registro binario: Formato de registro utilizado por herramientas basadas en las librerías "libpcap", como por ejemplo "tcpdump". Se aplica para registrar el tráfico de red. Algunas de las ventajas del formato binario sobre el formato ASCII son que ocupa menos, y la información que contiene puede ser accedida en menor tiempo.

Función resumen: Función de cifrado que permite detectar cambios en objetos.

Generación de Patrones Probables: Técnica que permite, mediante métodos estadísticos, predecir eventos futuros basándose en los que ya han tenido lugar. En determinadas circunstancias, si no se cumplen los eventos esperados, hay posibilidades de que se trate de un ataque.

Gestión de redes: Controlar diversos aspectos de una red para optimizar su eficiencia. Las cinco categorías de gestión de red son: seguridad, fallo, auditoría, configuración y gestión de rendimiento.

Gestión de rendimiento: En gestión de redes, medición de los diferentes elementos de la red. Los resultados de estas mediciones se utilizan para optimizar su funcionamiento.

Gestión de seguridad: 1. Proceso de establecer y mantener la seguridad en un sistema o red de sistemas informáticos. Las etapas de este proceso incluyen la prevención de problemas de seguridad, detección de intrusiones, investigación de intrusiones, y resolución. 2. En gestión de redes, controlar (permitir, limitar, restringir, o denegar) acceso a la red y recursos, buscar intrusiones, identificar puntos de entrada de intrusiones, y reparar o cerrar estas posibles vías de acceso.

Gestor de registro de actividades: Componente de sistema encargado de las labores de registro de actividad.

Grupo de Trabajo de Ingeniería de Internet (IETF): Una de las principales organizaciones encargadas de la formulación de estándares en Internet.

Gusano: Programa informático que se auto-duplica y auto-propaga. A diferencia que los virus, suelen estar diseñados para redes.

Horizonte de evento: Límite de tiempo aplicable a una característica de sistema determinada, como por ejemplo la diferencia de tiempo entre dos entradas a un sistema.

Identificación de Sistema Operativo: Conjunto de técnicas utilizadas para determinar la identidad del sistema operativo de un sistema remoto. Generalmente se logra mediante el envío de determinados datos de red y el posterior análisis de las respuestas recibidas.

Identificación y autenticación (I&A): Mecanismo de seguridad que asigna una identidad única a cada usuario (identificación) y la comprueba (autenticación).

Inodo, nodo i: Estructura de datos que contiene información sobre cada archivo en sistemas UNIX. Cada archivo tiene un nodo-i asociado.

Integración: En ingeniería de sistemas, combinación de componentes en una entidad coherente.

Integridad: Requisito de seguridad que indica que la información deberá ser protegida ante alteraciones no autorizadas.

Inteligencia artificial (AI): Ciencia que busca la comprensión de entidades inteligentes.

Interconexión de Sistemas Abiertos (OSI): Estructura de protocolos en siete niveles propuesta por ISO ("International Standardisation Organisation") e ITU-T ("International Telecommunication Union Telecommunication Standardization Sector").

Interfaz Común de Acceso (CGI): Especificación para la transmisión de datos entre programas residentes en servidores Web y navegadores.

Interfaz de comandos polimórfica: Interfaz de comandos cuyo código cambia con cada ejecución. Esto se hace para evadir los detectores de intrusión basados en reglas. En la mayoría de los casos se utilizan durante ataques de desbordamiento de búfer.

Interfaz de comandos segura (SSH): También conocida como "Secure Socket Shell", es una interfaz de comandos basada en UNIX y un protocolo para acceder de forma segura a una máquina

remota. Es ampliamente utilizada por administradores de red para realizar tareas de gestión y control. SSH es un conjunto de tres utilidades: `slogin`, `ssh` y `scp`; versiones seguras de las anteriores utilidades de UNIX: `rlogin`, `rsh` y `rcp`.

Interfaz de programación de aplicaciones (API): Conjunto de rutinas, protocolos, y herramientas para la construcción de aplicaciones software.

Interoperabilidad: Capacidad de un sistema para trabajar con otros sin que sean necesarios grandes esfuerzos por parte del usuario.

Intrusión: Violación intencionada de las políticas de seguridad de un sistema.

Intrusos desde el exterior: Usuarios no autorizados de un sistema.

Libpcap: Interfaz independiente del sistema, para la captura de paquetes de nivel de usuario, escrito en el "Lawrence Berkeley National Laboratory".

Libro Marrón ("Guía para la Comprensión de la Auditoría en Sistemas de Confianza"): Uno de los volúmenes de la Serie Arco Iris que explica los criterios del sistema de auditoría de Sistemas de Confianza, mencionando aspectos relevantes para los sistemas de detección de intrusiones.

Lista de Control de Acceso (ACL): Conjunto de datos que indican al sistema operativo qué permisos tiene un usuario o grupo sobre un determinado objeto de sistema. Cada objeto tiene atributos de seguridad únicos que indican qué usuarios pueden accederlo, y la Lista de Control de Acceso contiene una descripción de los privilegios de acceso de cada objeto y usuario.

Lote: En informática, programa asignado a un sistema para ser ejecutado de forma desatendida. Los trabajos por lotes suelen ejecutarse en un plano secundario, mientras que los interactivos se ejecutan en primer plano.

Lógica difusa: Forma de razonamiento que incorpora criterios múltiples para tomar decisiones y valores múltiples para evaluar posibilidades. Permite formalizar operaciones del razonamiento impreciso sobre conceptos imprecisos, comunes en el razonamiento humano.

Malla mundial, telaraña mundial. Véase también ("web"): Sistema de información distribuido, basado en hipertexto. La información puede ser de diferente naturaleza, como por ejemplo texto, gráfico, audio, o vídeo.

Malla, telaraña. Véase también ("WWW"): Servidor de información WWW. Se utiliza también para definir el universo WWW en su totalidad.

Marco de Detección de Intrusiones Común (CIDF): Grupo de trabajo encargado de crear interfaces que permitan a los desarrolladores de detección de intrusiones compartir sus conocimientos y poder reutilizar los resultados en otros sistemas. Fue fundado por Teresa Lunt.

Minería de datos: Arte y ciencia de descubrir y explotar relaciones nuevas, útiles, y provechosas en grandes cantidades de información.

Modo en línea: Método de interceptación del tráfico de red que consiste en hacer pasar todo el tráfico a través de un monitor o rastreador, generalmente configurado como puente para minimizar el impacto sobre el rendimiento de la red y dificultar su detección.

Modo promiscuo: Respecto a una interfaz de red, el modo de operación que genera una interrupción por cada actividad de red detectada. Esto permite a la interfaz recoger todo el tráfico de red de su segmento y entregárselo al detector de intrusiones.

Monitor, monitorizar: Cualquier mecanismo o método utilizado por un sistema de detección de intrusiones para obtener información.

Monitorización intrusa: En análisis de vulnerabilidades, obtener información mediante la realización de comprobaciones que afectan al estado del sistema, llegando en algunos casos a provocar su caída.

Monitorización no intrusa: En análisis de vulnerabilidades, obtener información mediante la ejecución de una lista de comprobaciones de los atributos del sistema.

Módulo de Seguridad Básico (BSM): Paquete de seguridad de "Sun Microsystems" proporcionado por los sistemas operativos de Sun para cumplir con los requisitos del documento TCSEC (la clase C2).

No paramétrico: Técnicas estadísticas que no hacen suposiciones sobre la distribución subyacente de los datos.

Paquete: Estructura de datos con una cabecera que puede estar o no lógicamente completa. Más a menudo, se refiere a un empaquetamiento físico de datos que lógico. Se utiliza para enviar datos a través de una red conmutada de paquetes.

Parche: En seguridad informática, código que corrige un fallo (agujero) de seguridad.

Petri Net Coloreada (CP-net): Lenguaje orientado a objetos para el diseño, especificación y verificación de sistemas. Está especialmente indicado en sistemas compuestos por gran variedad de procesos que necesitan estar comunicados y sincronizados.

Pila de protocolos: Conjunto de protocolos que se implementan en un determinado sistema.

Pila: Área de datos o búfer utilizada para almacenar peticiones que deben ser atendidas. Tiene una estructura FILO (primero en entrar, último en salir) o LIFO (último en entrar, primero en salir).

Política de monitorización: Conjunto de reglas que definen la forma en que se debe capturar e interpretar la información.

Política de seguridad: 1. Conjunto de estatutos que describen la filosofía de una organización respecto a la protección de su información y sistemas informáticos. 2. Conjunto de reglas que ponen en práctica los requisitos de seguridad del sistema.

Privacidad: Estar libre de accesos no autorizados.

Privilegio: Nivel de confianza perteneciente a un objeto de sistema.

Procesamiento por lotes, procesamiento en lotes: Procesamiento realizado en intervalos de tiempo, de forma discontinua.

Procesos de confianza: Procesos que sirven para cumplir un objetivo de seguridad.

Protocolo Internet versión 6: Revisión del Protocolo Internet que viene a sustituir a la tradicional versión 4. Cuenta con nuevas características, como mejoras en las direcciones, simplificación de la cabecera, nuevo soporte de extensiones y opciones, etiquetado de tráfico, y capacidades de autenticación y privacidad.

Protocolo Simple de Transferencia de Correo: Protocolo de comunicaciones para la transmisión de correo electrónico entre ordenadores.

Protocolo de Control de Transmisión / Protocolo Internet (TCP/IP): Conjunto de protocolos básico sobre los que se fundamenta Internet. Se sitúan en torno al nivel tres y cuatro del modelo OSI.

Protocolo de Tiempo de Red (NTP): Protocolo situado sobre TCP/IP diseñado para permitir la sincronización de los relojes de las máquinas conectadas a través de una red.

Protocolo de Transferencia de Ficheros (FTP): Protocolo que permite a un usuario de un sistema acceder a otro sistema de una red, e intercambiar información con el mismo.

Protocolo de Transferencia de Hipertexto (HTTP). Véase también ("WWW"): Protocolo usado para la transferencia de documentos WWW.

Puente. Véase también ("conmutador"): Dispositivo que permite la interconexión de dos redes con igual o distintos interfaces o pila de protocolos. Realiza funciones de encaminamiento de paquetes a nivel de enlace. Un puente multi-puerto es prácticamente un conmutador.

Puerta trasera. Véase también ("vulnerabilidad"): Mecanismo que permite a un atacante entrar y controlar un sistema de forma oculta. Suelen instalarse justo después de comprometer un sistema.

Puerto de extensión, puerto abarcador: Puerto especial con el que cuentan algunos conmutadores avanzados. Está programado para poder recibir una copia del tráfico destinado a uno o varios puertos del conmutador.

Rastreador: Dispositivo capaz de capturar todos los paquetes de datos que viajan por el segmento de red al que está conectado. Cuenta con una interfaz de red en modo promiscuo.

Rechazo, esquivamiento. Véase también ("respuesta activa"): Respuesta ante un ataque en la que el sistema termina y rechaza las subsiguientes conexiones con dirección del atacante.

Red Privada Virtual (VPN): Red generalmente construida sobre infraestructura pública, que utiliza métodos de cifrado y otros mecanismos de seguridad para proteger el acceso y la privacidad de sus comunicaciones.

Red trampa, red de miel: Es un tipo de sistema trampa. Es una red de sistemas reales diseñada para ser comprometida.

Reducción de auditoría: Método utilizado para eliminar información redundante o no necesaria de los registros de auditoría.

Registro de aplicación: En sistemas Windows, es uno de los tres tipos de registros de eventos. Este registro contiene los eventos generados por las aplicaciones.

Registro de eventos: Mecanismo de auditoría utilizado por sistemas Windows.

Registro de seguridad: En sistemas Windows, es uno de los tres tipos de registros de eventos. Este registro contiene los eventos considerados como relevantes en materia de seguridad.

Registro de sistema: 1. En sistemas Windows, es uno de los tres tipos de registros de eventos. Este registro contiene los eventos generados por los componentes de sistema. 2. en sistemas UNIX, ficheros de eventos de sistema y aplicaciones, que suelen consistir en ficheros de texto consistentes en una línea por cada evento.

Registros de auditoría de sistema operativo: Historiales de eventos de sistema generados por el mecanismo especializado de un sistema operativo.

Repetidor. Véase también ("concentrador"): Dispositivo que regenera la señal que pasa a través de la red, permitiendo extender la distancia de transmisión de dicha señal. Un repetidor multi-puerto se conoce como un concentrador.

Respuesta activa: Respuesta en la que el sistema (automáticamente, o junto con el usuario) modifica el curso del ataque. Hay tres formas básicas de respuestas activas: ejecutar acciones contra el intruso, corregir el entorno, y recopilar más información.

Respuesta pasiva: Respuesta en la que el sistema simplemente registra e informa de la intrusión o ataque, delegando en el usuario las acciones subsecuentes.

Rompedor de contraseñas: Herramienta de seguridad diseñada para descubrir las contraseñas de los usuarios. En la mayoría de los casos se utilizan diferentes aproximaciones para obtenerlas.

STREAMs: Modelo de programación de sistema para el desarrollo de controladores de dispositivos.

SVR4++: Una propuesta de estándar para el formato de los registros de auditoría de los sistemas operativos. Fue publicada por Stephen Smaha.

Salto de red: Estrategia de ataque en la que el atacante intenta ocultar su identidad realizando sus actividades desde otros sistemas comprometidos.

Segmento: Unidad lógica de datos, en particular un segmento de TCP es la unidad de datos transferida entre dos módulos de TCP.

Seguridad de Protocolo Internet (IPSec): Conjunto de protocolos desarrollados por IETF para soportar el intercambio seguro de paquetes en el nivel IP. IPsec se utiliza ampliamente para implementar Redes Virtuales Privadas (VPNs).

Seguridad: 1. Según un enfoque práctico, la seguridad implica que un sistema se comporte de la manera esperada. Esta definición depende de los niveles de confianza 2. Según un enfoque formal, consiste en el cumplimiento de la "tríada de conceptos": confidencialidad, integridad y disponibilidad.

Sensor. Véase también ("agente"): En detección de intrusión, una entidad que realiza labores de monitorización y obtención de datos de las fuentes de información. También conocido como agente. En muchos IDS, el sensor y el analizador forman parte del mismo componente.

Serie Arco Iris. Véase también ("Libro Naranja" y "Libro Marrón"): Conjunto de documento que definen la Iniciativa de Sistemas Confiables, un programa del gobierno de EE.UU. para resolver problemas relacionados con la seguridad informática. Los nombres de los documentos se corresponden con los colores de sus cubiertas.

Servidor de Gestión de Sistemas (SMS): Sistema desarrollado por Microsoft que permite gestionar la configuración de estaciones y servidores Windows.

Sistema Experto de Detección de Intrusiones (IDES). Véase también "Sistema Experto de detección de intrusiones de siguiente generación (NIDES)": Sistema de Detección de Intrusiones basado en reglas diseñado para detectar ataques conocidos. Fue el precursor de NIDES.

Sistema Experto de detección de intrusiones de siguiente generación (NIDES). Véase también "Sistema Experto de Detección de Intrusiones (IDES)": Detector de intrusiones que realiza funciones de monitorización en tiempo real de actividades de usuario a través de múltiples sistemas conectados vía *Ethernet*.

Sistema de Nombres de Dominio (DNS): Servicio distribuido de búsqueda de datos que realiza traducciones entre direcciones IP y nombres de máquinas. La estructura de los nombres de máquina (nombres de dominio), que son más fáciles de recordar que las direcciones IP, sigue una estructura jerárquica.

Sistema de detección de intrusiones (IDS): Sistema que monitoriza redes de ordenadores y sistemas en busca de violaciones de políticas de seguridad. Está compuesto por tres elementos fundamentales: fuentes de información, motor de análisis y mecanismos de respuesta.

Sistema de prevención de intrusiones (IPS): Sistema que combina las capacidades de bloqueo de un cortafuegos y las de análisis de un IDS. Está diseñado para detener ataques antes de que tengan éxito.

Sistema experto: Aplicación informática que realiza una tarea que podría realizar un humano experto, como por ejemplo, diagnóstico de enfermedades, ataques, o búsqueda de rutas de encaminamiento óptimas. Los sistemas expertos pertenecen a una categoría general de aplicaciones que utilizan técnicas de inteligencia artificial.

Sistema trampa, tarro de miel: Recurso de sistema de información cuyo valor reside en el uso no autorizado o ilícito de dicho recurso.

Sistemas de confianza: Sistemas que emplean las suficientes medidas para cumplir los requisitos necesarios para su uso en el proceso de información sensible o clasificada.

Software libre. Véase también ("código abierto"): Código que otorga libertad a los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el mismo.

Sondeo de puertos, escaneo de puertos. Véase también ("escaneo sigiloso de puertos"): Barrido de puertos generalmente para determinar qué servicios ofrece un sistema. Es uno de los métodos más comunes entre los atacantes para obtener información de sus objetivos.

Suma de control, suma de verificación, suma de comprobación: Algoritmo matemático que genera un número único a partir de un conjunto de datos, utilizada para comprobar la integridad de los mismos.

Tcpdump: Herramienta de monitorización y adquisición de datos que realiza labores de filtrado, recopilación, y visualización de paquetes.

Uso indebido: Actividad o comportamiento conocida como mala o inapropiada.

Virus polimórfico: Virus informático que cambia de aspecto con cada ejecución. Esta característica tiene el objeto de evitar los detectores de virus.

Vulnerabilidades: Debilidades en un sistema que pueden ser utilizadas para violar las políticas de seguridad.

Zona desmilitarizada, red perimétrica (DMZ): Máquina o pequeña subred situada entre una red interna de confianza (como una red local privada) y una red externa no confiable (como Internet). Normalmente en esta zona se sitúan los dispositivos accesibles desde Internet, como servidores Web, FTP, SMTP o DNS, evitando la necesidad de acceso desde el exterior a la red privada. Este término es de origen militar, y se utiliza para definir un área situada entre dos enemigos.

Apéndice C - Bibliografía

Esta bibliografía está basada en una lista iniciada y mantenida por Steve Smaha y su equipo de *Haystack Labs* junto con algunas modificaciones y actualizaciones hechas por Rebecca Bace. Yo he aprovechado para añadir algunas entradas más de reciente aparición, además de comprobar y actualizar las referencias existentes a documentos en Internet. Se ha procurado incluir en la siguiente lista exclusivamente aquellas referencias bibliográficas relacionadas de una u otra forma con la detección de intrusiones.

- Abbott, Robert P., J.S. Chin, J.E. Donnelley, W.L. Konigsford, S. Tokubo, and D.A. Webb. *Security Analysis and Enhancements of Computer Operating Systems*. Technical report NBSIR 76 - 1041, Institute for Computer Science and Technology, National Bureau of Standards, 1976.
- Anderson, James P. *Computer Security Threat Monitoring and Surveillance*. Fort Washington, PA: James P. Anderson Co., 1980.
- _____. *Computer Security Technology Planning Study*. ESD-TR-73-51, v II. Electronic Systems Division, Air Force Systems Command, Hanscom Field, Bedford, MA, October 1972.
- Anderson, Ross. *Liability and Computer Security: Nine Principles*. Third European Symposium on Research in Computer Security (ESORICS), Brighton, U.K., November 1994.
- Anderson, Ross, and A. Khattak. *The Use of Information Retrieval Techniques for Intrusion Detection*. Presentation, First International Workshop on the Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium, September 1998.
- Anderson, Ross, and R. Needham. *Programming Satan's Computer*. Computer Science Today, Computer Science Today, Lecture Notes in Computer Science, Springer-Verlag, Heidelberg, Germany, v 1000: 426—441. Springer LNCS v 1000: 426—441.
- Axelsson, Stefan. *On a Difficulty of Intrusion Detection*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Axelsson, Stefan, U. Lindqvist, U. Gustafson, and E. Jonsson. *An Approach to UNIX Security Logging*. Proceedings of the Twenty-First National Information System Security Conference, Crystal City, VA, October 1998.
- Bace, Rebecca. *A New Look at Perpetrators of Computer Crime*. Proceedings of the Sixteenth Department of Energy Computer Security Group Conference, Denver, CO, May 1994.
- _____. *Intrusion Detection*. Macmillan Technical Publishing, 2000.
- _____ and Peter Mell. *Intrusion Detection Systems*. [en línea]. [consultado en marzo, 2003]. Disponible en <<http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf>>.
- Balasubramaniyan, J. S., J. o. García-Fernández, D. Isacoff, E. H. Spafford, and D. Zamboni. *An Architecture for Intrusion Detection Using Autonomous Agents*. COAST technical report 98/05, Purdue University, W. Lafayette, IN, June 1998.
- Balasubramaniyan, Jai S., J. o. Garcia-Fernandez, D. Isacoff, E. H. Spafford, and D. Zamboni. *An Architecture for Intrusion Detection Using Autonomous Agents*. Proceedings of the Fourteenth IEEE Computer Security Applications Conference, Tucson, AZ: 13 - 24, December 1998.

- Baldwin, Robert W. *Kuang: Rule-Based Security Checking*. MIT, Lab for Computer Science Programming Systems Research Group, May 1994.
- Baldwin, Robert W. *Rule-Based Analysis of Computer Security*. Massachusetts Institute of Technology, June 1987.
- Banning, Debra, G. Ellingwood, C. Franklin, C. Muckinhirn, and D. Price. *Auditing of Distributed Systems*. Proceedings of the Fourteenth National Computer Security Conference, Washington, DC, October 1991.
- Bauer, David S. and M. E. Koblenz. *NIDX—An Expert System for Real-Time Network Intrusion Detection*. Proceedings of the IEEE Computer Networking Symposium, New York, NY, pp. 98 - 106, April 1988.
- Bishop, Matt. *A Model of Security Monitoring*. Proceedings of the Fifth Annual Computer Security Applications Conference, Tucson, AZ, December 1989.
- _____. *A Standard Audit Log Format*. Proceedings of the 1995 National Information Systems Security Conference, Baltimore, MD, October 1995.
- _____. *Vulnerabilities Analysis: Extended Abstract*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Bishop, Matt, S. Cheung, C. Wee, J. Frank, J. Hoagland, and S. Samorodin. *The Threat from the Net*. IEEE Spectrum 34, no. 8(1997): 56 - 63.
- Bishop, Matt and Michael Dilger. *Checking for Race Conditions in File Access*. Computing Systems 9, no. 2 (Spring 1996): 131 - 152.
- Blain, Laurent and Yves Deswarte. *An Intrusion-Tolerant Security Server for an Open Distributed System*. Proceedings of the European Symposium on Research in Computer Security, Toulouse, France, October 1990.
- Bradley, Kirk, S. Cheung, N. Puketza, B. Mukhejee, and B.. A. Olsson. *Detecting Disruptive Routers: A Distributed Network Monitoring Approach*. Proceedings of the Nineteenth IEEE Symposium on Security and Privacy, Oakland, CA, May 1998.
- Brentano, James. *An Expert System for Detecting Attacks on Distributed Computer Systems*. Master thesis, Division of Computer Science, University of California, Davis, CA, March 1991.
- Brentano, James, S. R. Snapp, G. V. Dias, T. L. Goan, L. T. Heberlein, C.-L. Ho, K. N. Leavitt, B. Mukherjee, and S. E. Smaha. *An Architecture for a Distributed Intrusion System*. DOE Computer Security Conference, Las Vegas, NV, March 1991.
- Bridges, Susan M. and Rayford B. Vaughn. *Fuzzy Data Mining and Genetic Algorithms applied to Intrusion Detection*. Mississippi State University.
- Carrettoni, F., S. Castano, G. Martella, and P. Samarati. *RETISS: A Real Time Security System for Threat Detection Using Fuzzy Logic*. Proceedings of the Twenty-Fifth Annual IEEE International Carnahan Conference on Security Technology, Taipei, Taiwan, October 1991.
- Cheswick, William. *An Evening with Berferd in Which a Cracker Is Lured, Endured, and Studied*. Proceedings of USENIX Security Conference, San Francisco, CA, Winter 1992.
- Cheung, Steven, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, J. Rowe, S. Staniford-Chen, B.. Yip, and D. Zerkle. *The Design of GrIDS: A Graph-Based Intrusion Detection System*. University of California, Davis, Computer Science Department technical report CSE-99 - 2 1999.

- Cheung, Steven and K. N. Levitt. *Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection*. Proceedings New Security Paradigms Workshop 1997, Cumbria, U.K., September 1997.
- Chung, Christina, M. Gertz, and K. Levitt. *Misuse Detection in Database Systems Through User Profiling*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Chung, Mandy, N. Puketza, R. A. Olsson, and B. Mukherjee. *Simulating Concurrent Intrusions for Testing Intrusion Detection Systems: Parallelizing Intrusions*. Proceedings of the 1995 National Information Systems Security Conference, Baltimore, MD, October 1995.
- Christoph, Gary G., K. A. Jackson, M. C. Neumann, C. L. B. Siciliano, D. D. Simmonds, C. A. Stallings, and J. L. Thompson. *UNICORN: Misuse Detection for UNICOS*. Proceedings of Supercomputing '95, San Diego, CA, December 1995.
- Clyde, Allen R. *Insider Threat Identification Systems*. Proceedings of the Tenth National Computer Security Conference, Washington, DC, September 1987.
- _____. *A Surveillance-Gate Model for Automated Information Security and Insider Threat Identification on Sensitive Computer Systems*. Proceedings of the Second Insider Threat Identification Systems Conference, Rockville, MD, April 1987.
- _____. *Suspicious Event Testing and Weighted Scoring for the Analysis of a Surveillance Data Set*. Proceedings of the Third Insider Threat Identification Systems Conference, Rockville, MD, April 1987.
- Crosbie, Mark. *Applying Genetic Programming to Intrusion Detection*. Proceedings of 1995 AAAI Fall Symposium on Genetic Programming, San Jose, CA, November 1995.
- Crosbie, Mark, B. Dole, T. Ellis, I. Krsul, and E. H. Spafford. *IDIOT - Users Guide*. Technical report TR-96 - 050, Purdue University, COAST Laboratory, W. Lafayette, IN, September 1996.
- Crosbie, Mark, and E. H. Spafford. *Defending a Computer System Using Autonomous Agents*. Proceedings of the Eighteenth National Information Systems Security Conference, Baltimore, MD, October 1995.
- D.C.I. Intelligence Information Handling Committee. Proceedings of the 1987 Intrusion Detection Expert System Conference, Vienna, VA, November 1987.
- Debar, Herve, M. Becker, and D. Siboni. *A Neural Network Component for an Intrusion Detection System*. Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1992.
- _____. and B. Dorizzi. *An Application of a Recurrent Network to an Intrusion Detection System*. Proceedings of the International Joint Conference on Neural Networks, Baltimore, MD, June 1992.
- Denning, Dorothy E. *An Intrusion Detection Model*. Proceedings of the 1986 IEEE Symposium on Security and Privacy, Oakland, CA, April 1986.
- _____. D. Edwards, R. Jagannathan, T. Lunt, and P. G. Neumann. *A Prototype IDES - A Real-Time Intrusion Detection Expert System*. Final report, Computer Science Lab, SRI International, Menlo Park, CA, August 1987.
- _____. and P. G. Neumann. *Requirements and Model for IDES - A Real-Time Intrusion Expert System*. Technical report, Computer Science Lab, SRI International, Menlo Park, CA, August 1985.

- de Queiroz, Jose Duarte, L. F. Rust da Costa Carmo, L. Pirmez. *Micael: An Autonomous Mobile Agent System to Protect Networked Applications of New Generation*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Dias, Gihan, K. N. Levitt, and B. Mukherjee. *Modeling Attacks on Computer Systems: Evaluating Vulnerabilities and Forming a Basis for Attack Detection*. SRI Intrusion Detection Workshop 5, Menlo Park, CA, May 1990.
- Dickerson, John E., Jukka Juslin, Ourania Koukousoula, Julie A. Dickerson. *Fuzzy Intrusion Detection*. Electrical and Computer Engineering Department. Iowa State University.
- Doak, Justin. *Intrusion Detection: The Application of Feature Selection, a Comparison of Algorithms, and the Application of a Network Analyzer*. Master thesis, University of California, Davis, CA, September 1992.
- Dowell, Cheri and P. Ramstedt. *The Computerwatch Data Reduction Tool*. Proceedings of the Thirteenth National Computer Security Conference, Washington, DC, October 1990.
- Farmer, D. and E. H. Spafford. *The Cops Security Checker System*. In the Proceedings of the Summer of 1990 Usenix Conference, Anaheim, CA: 165 - 170, June 1990.
- Farmer, Dan, and W. Venema. *Improving the Security of Your Site by Breaking into It*. Internet [en línea]. 1993 [consultado en junio, 2003] Disponible desde <<http://www.fish.com>>.
- Farmer, D. and W. Venema. *Security Administrator's Tool for Analyzing Networks (SATAN)*. [en línea]. [consultado en junio, 2003] Disponible desde <<http://www.fish.com/zen/satan/satan.html>>.
- Feiertag, Richard, L. Benzinger, S. Rho, and S. Wu. *Intrusion Detection Intercomponent Adaptive Negotiation*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Fox K. L., R. Henning, J. Reed. A Neural Network Approach Towards Intrusion Detection'. Proceedings of the 13th National Computer Security Conference. Pp 125-134 Washintong, DC, October 1990.
- Frank, Jeremy. *Machine Learning and Intrusion Detection: Current and Future Directions*. Proceedings of the Seventeenth National Computer Security Conference, Baltimore, MD, October 1994.
- Frincke, Deborah, D. Tobin, and Y. Ho. *Planning, Petri Nets, and Intrusion Detection*. Proceedings of Twenty-First National Information System Security Conference, Crystal City, VA, October 1998.
- Garvey, Thomas D. and T. Lunt. *Model-Based Intrusion Detection*. Proceedings of the Fourteenth National Computer Security Conference, Washington, DC, October 1991.
- Gates, James D. *Tools for Identifying the Source of Security Breaches*. Proceedings of the Third Insider Threat Identification Systems Conference, Rockville, MD, April 1987.
- Grediaga, A., Ibarra, F., Ledesma, B., Brotons, F. *Utilización de redes neuronales para la detección de intrusos*. Departamento de Tecnología Informática y Computación. Universidad de Alicante.
- Gross, Andrew H. *Analyzing Computer Intrusions*. Ph.D. thesis, University of California, San Diego, Department of Computer Sciences, San Diego, CA, 1997.

- Guha, Biswaroop and B. Mukherjee. *Network Security via Reverse Engineering of TCP Code: Vulnerability Analysis and Proposed Solutions*. Proceedings of the IEEE Infocom '96, San Francisco, CA, March 1996.
- Gupta, S. and V. D. Gligor. *Experience with a Penetration Analysis Method and Tool*. Proceedings of the Fifteenth National Computer Security Conference, Baltimore, MD, October 1992.
- Habra, N., B. Le Charlier, and A. Mounji. *Preliminary Report on Advanced Security Audit Trail Analysis on UNIX*. Universitaires Notre Dame de la Paix, Namur, Belgium, Research report, December 1991.
- _____, B. Le Charlier, and A. Mounji. *Advanced Security Audit Trail Analysis on UNIX: Implementation Design of the NADF Evaluator*. Research report, Universitaires Notre Dame de la Paix, Namur, Belgium, March 1993.
- _____, B. Le Charlier, A. Mounji, and I. Mathieu. *ASAXL Software Architecture and Rule-Based Language for Universal Audit Trail Analysis*. Proceedings of the Second European Symposium on Research in Computer Security (ESOBJCS), Toulouse, France, November 1992.
- Halme, Lawrence K and B.. K. Bauer. *AJNT Misbehaving - A Taxonomy of Antiintrusion Techniques*. Proceedings of the Eighteenth National Information Systems Security Conference, Baltimore, MD, October 1995.
- _____, and Brian L. Kahn. *Building a Security Monitor with Adaptive User Work Profiles*. Proceedings of the Eleventh National Computer Security Conference, Washington, DC, October 1988.
- _____, and J. V. Home. *Automated Analysis of Computer System Audit Trails for Security Purposes*. Proceedings of the Ninth National Computer Security Conference, Washington, DC, September 1986.
- Hansen, Stephen E. and T. Atkins. *Automated System Monitoring and Notification with Swatch*. Proceedings of the USENIX Systems Administration (LISA VII) Conference, Monterey, CA, November 1993.
- Haskins, Denis H. *Keeping Watch on a VAX*. Digital Review, December 16, 1988.
- Heady, Richard, G. Luger, A. B. Maccabe, and M. Servilla. *The Architecture of a Network Level Intrusion Detection System*. Technical report CS90 - 20, Department of Computer Science, University of New Mexico, Albuquerque, NM, August 1990.
- _____, G. Luger, A. B. Maccabe, M. Servilla, and J. Sturtevant. *The Prototype Implementation of a Network Level Intrusion Detection System*. Technical Report CS91 - 11, Department of Computer Science, University of New Mexico, Albuquerque, NM, April 1991.
- Heberlein, Todd. *Network Security Monitor (NSM) - Final Report*. Lawrence Livermore National Laboratory, Davis, CA, February 1995.
- _____, and M. Bishop. *Attack Class: Address Spoofing*. Nineteenth National Information Systems Security Conference, Baltimore, MD, October 1996.
- _____, K. Levitt, and B. Mukherjee. *A Network Security Monitor*. Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1990.
- _____, K. Levitt, and B. Mukherjee. *A Method to Detect Intrusive Activity in a Networked Environment*. Proceedings of the Fourteenth National Computer Security Conference, Washington, DC, October 1991.

- _____, B. Mukherjee, and K. N. Levitt. *Internetwork Security Monitor*. Proceedings of the Fifteenth National Computer Security Conference, October 1992.
- _____, B. Mukherjee, K. N. Levitt, and G. Dias (with D. Mansur). *Towards Detecting Intrusions in a Networked Environment*. Proceedings of the Fourteenth Department of Energy Computer Security Group Conference, May 1991.
- Helman, Paul and G. Liepins. *Statistical Foundations of Audit Trail Analysis for the Detection of Computer Misuse*. IEEE Transactions on Software Engineering 19, no. 9(1993): 886 - 901.
- _____, G. Liepins, and W. Richards. *Foundations of Intrusion Detection*. Proceedings of the Fifth Computer Security Foundations Workshop, Franconia, NH, June 1992.
- Hoagland, J., S. Staniford. Silicon Defense. *SPICE / SPADE*. [en línea]. Actualizado con frecuencia [consultado en marzo, 2003]. Disponible desde Internet en <<http://www.silicondefense.com/software/spice/>>.
- _____, C. Wee, and K. N. Levitt. *Audit Log Analysis Using the Visual Audit Browser Toolkit*. University of California, Davis, Computer Science Department technical report CSE-95 - 11, 1995.
- Hochberg, Judith, K. Jackson, C. Stallings, J. F. McClary, D. DuBois, and J. Ford. *NADIR: An Automated System for Detecting Network Intrusion and Misuse*. Computers and Security 12, no. 3 (May 1993): 235 - 248.
- Hofmeyr, Steven A., S. Forrest, and A. Somayaji. *Intrusion Detection Using Sequences of System Calls*. Journal of Computer Security 6, no. 3 (1996): 151 - 180.
- IBM Research, Zurich Research Laboratory. Andreas Wespi, Marc Dacier, and Hervé Debar. *Intrusion Detection Using Variable-Length Audit Trail Patterns*. Springer-Verlag Berlin Heidelberg, 2000.
- Ilgun, Koral. *USTAT: A Real-Time Intrusion Detection System for UNIX*. Master thesis, University of California, Santa Barbara, CA, November 1992.
- _____. *US TAT: A Real-Time Intrusion Detection System for UNIX*. Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1993.
- _____, R. A. Kemmerer, and P. A. Porras. *State Transition Analysis: A Rule-Based Intrusion Detection Approach*. IEEE Transactions on Software Engineering 21, no. 3 (March 1995): 181 - 199.
- Jackson, Kathleen A., D. DuBois, and C. Stallings. *An Expert System Application for Network Intrusion Detection*. Proceedings of the Fourteenth National Computer Security Conference, Washington, DC, October 1991.
- _____, M. C. Neumann, D. Simmonds, C. Stallings, J. Thompson, and G. Christoph. *An Automated Computer Misuse Detection System for UNICOS*. Proceedings of the Cray Users Group Conference, Tours, France, October 1994.
- Jajodia, S., S. K. Gadia, G. Bhargava, and E. H. Sibley. *Audit Trail Organization in Relational Databases*. Proceedings of the 1989 IFIP Workshop on Database Security, Monterey, CA, September 1989.
- Javitz, Harold S. and Valdes, A. *The SRI IDES Statistical Anomaly Detector*. Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1991.

- Josué Kuri, Gonzalo Navarro, Ludovic Mé, Laurent Heye. *A Pattern Matching Based Filter for Audit Reduction and Fast Detection of Potential Intrusions*. Springer-Verlag Berlin Heidelberg, 2000.
- Kahn, Clifford, P. Porras, S. Staniford-Chen, and B. Tung. *A Common Intrusion Detection Framework*. Submitted to the Journal of Computer Security, July 1998.
- Kelsey, John and B. Schneier. *Minimizing Bandwidth for Remote Access to Cryptographically Protected Audit Logs*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Kerchen, Paul, R. Lo, J. Crossley, G. Elkinbard, and R. Olsson. *Static Analysis Virus Detection Tools for UNIX Systems*. Proceedings of the Thirteenth National Computer Security Conference, Washington, DC, October 1990.
- Kim, Gene H. and E. H. Spafford. *Writing, Supporting, and Evaluating Tripwire: A Publicly Available Security Tool*. Proceedings of the USENIX UNIX Applications Development Symposium: 89 - 107, 1994.
- Kim Gene H. and E. H. Spafford. *Tripwire: A Case Study in Integrity Monitoring*. Internet Beseiged: Countering Cyberspace Scofflaws; edited by Dorothy and Peter Denning, Addison-Wesley, 1997.
- King, Maria M. *Identifying and Controlling Undesirable Program Behaviors*. Proceedings of the Fourteenth National Computer Security Conference, Washington, DC, October 1991.
- Ko, Calvin C. W. *Execution Monitoring of Security-Critical Programs in a Distributed System: A Specification-Based Approach*. Ph.D. thesis, University of California, Davis, CA, August 1996.
- _____, G. Fink, and K. Levitt. *Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring*. Proceedings of the Tenth Annual Computer Security Applications Conference, Orlando, FL, December 1994.
- _____, G. Fink, and K. Levitt. 'Execution Monitoring of Security-Critical Programs in Distributed Systems: A Specification-Based Approach. *Proceedings of the IEEE Symposium on Security and Privacy*, May 1997.
- _____, D. Frincke, T. Goan, L. T. Heberlein, K. Levitt, B. Mukherjee, and C. Wee. *Analysis of an Algorithm for Distributed Recognition and Accountability*. Proceedings of the First ACM Conference on Computer and Communication Security. Fairfax, VA, November 1993.
- Kogan, Boris and S. Jajodia. *An Audit Model for Object-Oriented Databases*. Proceedings of the Seventh Computer Security Applications Conference, San Antonio, TX, December 1991.
- Kuhn, Jeffrey D. *Research Toward Intrusion Detection Through the Automated Abstraction of Audit Data*. Proceedings of the Ninth National Computer Security Conference, Washington, DC, September 1986.
- Kumar, Sandeep. *Classification and Detection of Computer Intrusions*. Ph.D. thesis, Purdue University Department of Computer Sciences, W. Lafayette, IN, 1995.
- _____, and E. Spafford. *A Pattern Matching Model for Misuse Intrusion Detection*. Proceedings of the Seventeenth National Computer Security Conference, Baltimore, MD, October 1994.
- _____, and E. Spafford. *A Software Architecture to Support Misuse Intrusion Detection*. CSD-TR-95 - 009, Department of Computer Sciences, Purdue University, W. Lafayette, IN, 1995.

- Lane, Terran and Carla E. Brodley. *An Application of Machine Learning to Anomaly Detection*. Proceedings of the Twentieth National Information System Security Conference, Baltimore, MD, October 1997.
- _____ and Carla E. Brodley. *Detecting the Abnormal: Machine Learning in Computer Security*. Purdue University, January 1997.
- _____ and Carla E. Brodley. *Sequence Matching and Learning in Anomaly Detection for Computer Security*. Purdue University, 1997.
- Lankewicz, Linda and M. Benard. *A Nonparametric Pattern Recognition Approach to Intrusion Detection*. Technical report TUTR 90 - 106, Tulane University Department of Computer Science, New Orleans, LA, October 1990.
- _____ and M. Benard. *Real-Time Anomaly Detection Using a Nonparametric Pattern Recognition Approach*. Proceedings of the Seventh Computer Security Applications Conference, San Antonio, TX, December 1991.
- Leach, John and Gianni Tedesco. *Firestorm*. [en línea]. 2002 [consultado en abril, 2003]. Disponible en <<http://www.scaramanga.co.uk/firestorm/index.html>>.
- Lee, Wenke and S. J. Stolfo. *Combining Knowledge Discovery and Knowledge Engineering to Build IDSs*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- _____, S. J. Stolfo, and K. W. Mok. *A Data Mining Framework for Building Intrusion Detection Models*. Proceedings of the Twentieth IEEE Symposium on Security and Privacy, Oakland, CA, 1999.
- _____, Wei Fan, Matthew Miller, Salvatore J. Stolfo, Philip K. Chan. *Using Anomalies to Detect Unknown and Known Network Intrusions*. College of Computing. Georgia Tech. IBM T.J. Watson Research. Columbia University. Computer Science, Florida Tech. November 2001.
- _____, and Salvatore J. Stolfo. *Adaptive Intrusion Detection: a Data Mining Approach*. Computer Science Department, Columbia University, 2000.
- _____, Rahul A. Nimbalkar, Kam K. Yee, Sunil B. Patil, Pragneshkumar H. Desai, Thuan T. Tran, and Salvatore J. Stolfo. *A Data Mining and CIDF Based Approach for Detecting Novel and Distributed Intrusions*. Computer Science Department, North Carolina State University. Computer Science Department, Columbia University, October 2000.
- _____, Salvatore J. Stolfo, Kui W. Mok. *Mining Audit Data to Build Intrusion Detection Models*. Computer Science Department, Columbia University, August 1998.
- Lemmonier, E. *Protocol Anomaly Detection in Network-based IDSs*. Defcom, Sweden, Stockholm, 28 de junio de 2001.
- Levitt, Karl, ed. *Proceedings of Workshop on Future Directions in Computer Misuse and Anomaly Detection*. University of California, Davis, CA, April 1992.
- Lichtman, Zavdi and John Kimmins. *An Audit Trail Reduction Paradigm Based on Trusted Processes*. Proceedings of the Thirteenth National Computer Security Conference, Washington, DC, October 1990.
- Liepins, Gunar E. and H. S. Vaccaro. *Anomaly Detection: Purpose and Framework*. Proceedings of the Twelfth National Computer Security Conference, Washington, DC, October 1989.
- _____ and H. S. Vaccaro. *Intrusion Detection: Its Role and Validation*. Computers and Security, v 11, Oxford, UK: Elsevier Science Publishers, Ltd, 1992: 347 - 355.

- Lindqvist, Ulf, E. Jonsson, and P. Kaijser. *The Remedy Dimension of Vulnerability Analysis*. Proceedings of Twenty-First National Information System Security Conference, Crystal City, VA, October 1998.
- Lundin, Emilie and E. Jonsson. *Privacy versus Intrusion Detection Analysis*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Lunt, Teresa. *Automated Audit Trail Analysis and Intrusion Detection: A Survey*. Proceedings of the Eleventh National Computer Security Conference, Washington, DC, October 1988.
- _____. *Real-Time Intrusion Detection*. Proceedings of COMPCON Spring '89, San Francisco, CA, February 1989.
- _____ and R. Jagannathan. *A Prototype Real-Time Intrusion Detection Expert System*. Proceedings of the 1988 IEEE Symposium on Security and Privacy, Oaldand, CA, April 1988.
- _____, R. Jagannathan, R. Lee, S. Listgarten, D. L. Edwards, P. G. Neumann, H. S. Javitz, and A. Valdez. *IDES: The Enhanced Prototype*. Computer Science Lab, SRI International, Menlo Park, CA, October 1988.
- _____, et al. *Knowledge-Based Intrusion Detection*. Proceedings of the AT Systems in Government Conference, Washington, DC, March 1989.
- _____, et al. *A Real-Time Intrusion Detection Expert System (IDES)*. Computer Science Lab, SRI International, Menlo Park, CA, May 1990.
- _____, et al. *IDES: A Progress Report*. Proceedings of the Sixth Annual Computer Security Applications Conference, Tucson, AZ, December 1990.
- _____. *A Survey of Intrusion Detection Techniques*. Computers and Security 12, 4 (June 1993): 405-418.
- McAuliffe, Noelle, D. Wolcott, L. Schaefer, N. Kelem, B. Hubbard, and T. Haley. *Is Your Computer Being Misused? A Survey of Current Intrusion Detection Technology*. Proceedings of the Sixth Annual Computer Security Applications Conference, Tucson, AZ, December 1990.
- McConnell, Jesse, D. A. Frincke, D. Tobin, J. Marconi, and D. Polla. *A Framework for Cooperative Intrusion Detection*. Proceedings of Twenty-First National Information System Security Conference, Crystal City, VA, October 1998.
- McKosky, Robert. *An Aposteriori Computer Security System to Identify Computer Viruses*. PhD Thesis, University of Alabama in Huntsville, Huntsville, AL, 1989.
- Mahoney, Matthew, P. K. Chan. *Learning Models of Network Traffic for Detecting Novel Attacks*. Florida Tech. technical report 2002-08. Disponible en <<http://cs.fit.edu/~tr/>>
- _____, P. K. Chan, *Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks*, Edmonton, Alberta: Proc. SIGKDD, 2002, 376-385.
- _____, P. K. Chan, *PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic*, Florida Tech. technical report 2001-04. Disponible en <<http://cs.fit.edu/~tr/>>.
- _____, V. *Network Traffic Anomaly Detection Based on Packet Bytes*. Florida Institute of Technology, Melbourne, Florida, 2003.
- Mandararis, Stefanos, M. Christensen, D. Zerkle, and K. Hermis. *A Data Mining Analysis of RTID Alarms*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.

- Mansfield, Glenn, K. Ohta, Y. Takei, N. Kato, and Y. Nemoto. *Towards Trapping Wily Intruders in the Large*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Mé, Ludovic. *Security Audit Trail Analysis Using Genetic Algorithms*. Proceedings of the Twelfth International Conference on Computer Safety, Reliability, and Security, Poznan, Poland, October 1993.
- _____. *GASSATA, a Genetic Algorithm as an Alternative Tool for Security Audit Trails Analysis*. First International Workshop on the Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium, September 1998.
- Mell, Peter and M. McLarnon. *Mobile Agent Attack Resistant Distributed Hierarchical Intrusion Detection Systems*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Moitra, Abba. *Real-Time Audit Log Viewer and Analyzer*. Proceedings of the Fourth Workshop on Computer Security Incident Handling, Denver, CO, August 1992.
- Mounji, A. *Languages and Tools for Rule-Based Distributed Intrusion Detection*. Thesis, Faculte's Universitaires Notre-Dame de la Paix, Namur, Belgium, September 1997.
- Mukherjee, Biswanath, L. T. Heberlein, and K. N. Levitt. *Network Intrusion Detection*. IEEE Network 8, no. 3 (May - June 1994): 26 - 41.
- Mutaf, Pars. *Defending Against a Denial-of-Service Attack on TCP*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- National Computer Security Center. *Department of Defense Trusted Computer System Evaluation Criteria*. Orange Book, DOD 5200.28-std, December 1985.
- _____. *DoD Trusted Computer System Evaluation Criteria*. DoD 5200.28 - STD, December 1985.
- _____. *Glossary of Computer Security Terms*. Versión 1, Rainbow Series, octubre 1988.
- _____. *A Guide to Understanding Audit in Trusted Systems*. NCSC-TG-OO1, v 2, June 1988.
- Neumann, Peter G. and D. B. Parker. *A Summary of Computer Misuse Techniques*. Proceedings of the Twelfth National Computer Security Conference, October 1989.
- _____ and P. A. Porras. *Experience with EMERALD to Date*. First USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, CA, April 1999.
- NFR. *Network Flight Recorder*. [en línea]. Fecha no disponible [consultado en marzo, 2003]. Disponible desde Internet <<http://www.nfr.net>>.
- O'Brien, David. *Recognizing and Recovering from Rootkit Attacks*. Sys Admin 5, no. 11, November 1996.
- Ong, T. H., C. P. Tan, Y. T. Tan, C. K. Chew, and C. Ting. *SNMS - Shadow Network Management System*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Paxson, Vern. *Bro: A System for Detecting Network Intruders in Real Time*. Seventh USENIX Security Symposium, San Antonio, TX, January 1998.

- _____ and M. Handley. *Defending Against Network IDS Evasion*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- _____, *Bro: A System for Detecting Network Intruders in Real-Time*. Lawrence Berkeley National Laboratory, Berkeley, CA and AT&T Center for Internet Research at ICSI, Berkeley, CA. [en línea]. 14 de diciembre de 1999 [consultado en marzo de 2003]. Disponible desde Internet en <<http://www.icir.org/vern/bro-info.html>>
- Piccioto, Jeffrey. *The Design of an Effective Auditing Subsystem*. Proceedings of the 1987 IEEE Symposium on Security and Privacy, Oakland, CA, April 1987.
- Pinacho, P., Contreras, R. *Una propuesta de Sistemas para Tratamiento de Intrusos Inspirado en la Biología*. Universidad de Santiago de Chile. Facultad de Ingeniería. Universidad de Concepción, Facultad de Ingeniería.
- Porras, Phiffip. *STAT, a State Transition Analysis Tool for Intrusion Detection*. Master thesis, Computer Science Department, University of California, Santa Barbara, CA, July 1992.
- _____ and R. A. Kemmerer. *Penetration State Transition Analysis: A Rule-Based Intrusion Detection Approach*. Proceedings of the Eighth Annual Computer Security Applications Conference, San Antonio, TX, November 1992.
- _____ and Peter Neumann. *EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances*. Proceedings of Twentieth National Information System Security Conference, Baltimore, MD, October 1997.
- Price, Katherine E. *Host-Based Misuse Detection and Conventional Operating Systems' Audit Data Collection*. Master thesis, Purdue University, W. Lafayette, IN, December 1997.
- Ptacek, Thomas H. and T. Newsham. *Insertions, Evasion, and Denial of Service: Eluding Network Intrusion Detection*. [en línea]. Enero 1998 [consultado en junio, 2003] Disponible desde <<http://www.securityfocus.com/data/library/ids.ps>>.
- Puketza, Nick, M. Chung, R. A. Olsson, and B. Mukherjee. *A Software Platform for Testing Intrusion Detection Systems*. IEEE Software 14, no. 5 (1997): 43 - 51.
- _____, B. Mukherjee, R. A. Olsson, and K. Zhang. *Testing Intrusion Detection Systems: Design Methodologies and Results from an Early Prototype*. Proceedings of the Seventeenth National Computer Security Conference, Baltimore, MD, October 1994.
- _____, K. Zhang, M. Chung, B. Mukherjee, and R. A. Olsson. *A Methodology for Testing Intrusion Detection Systems*. IEEE Transactions on Software Engineering 22, no. 10: 719 - 729, October 1996.
- Rao, K. N. *Security Audit for Embedded Avionics Systems*. Proceedings of the Fifth Annual Computer Security Applications Conference, Tucson, AZ, December 1989.
- Roesch, Marty et al. *Snort.org*. [en línea]. Actualizado semanalmente [consultado en marzo de 2003]. Disponible en <<http://www.snort.org>>.
- Saltzer, Jerome H. and Michael D. Schroeder. *The Protection of Information in Computer Systems*. Proceedings of the IEEE, 63, no. 9: 1278 - 1308, September 1975.
- Schaefer, Marvin, B. Hubbard, D. Sterne, T. K. Haley, J. N. McAuliffe, and D. Woolcott. *Auditing: A Relevant Contribution To Trusted Database Management Systems*. Proceedings of the Fifth Annual Computer Security Applications Conference, Tucson, AZ, December 1989.

- Schaen, Samuel I. and B. McKenney. *Network Auditing: Issues and Recommendations*. Proceedings of the Seventh Computer Security Applications Conference, San Antonio, TX, December 1991.
- Schneier, Bruce and J. Kelsey. *Cryptographic Support for Secure Logs on Untrusted Machines*. Proceedings of Seventh USENIX Security Symposium San Antonio, TX: 53 - 62, January 1998.
- _____ and J. Kelsey. *Secure Audit Logs to Support Computer Forensics*. ACM Transactions on Information and System Security 1, no. 3 (1999), to appear.
- Sebring, Michael M., E. Shellhouse, M. E. Hanna, and R. A. Whitehurst. *Expert Systems in Intrusion Detection: A Case Study*. Proceedings of the Eleventh National Computer Security Conference, Washington, DC, October 1988.
- Seiden, Kenneth F. and J. P. Melanson. *The Auditing Facility for a VMM Security Kernel*. Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1990.
- Sekar, R., M. Bendre, D. Dhurjati, P. Bollineni, *A Fast Automaton-based Method for Detecting Anomalous Program Behaviors*. Proceedings of the 2001 IEEE Symposium on Security and Privacy.
- Selezniov, Alexandr and S. Puuronen. *Anomaly Intrusion Detection Systems: Handling Temporal Relations Between Events*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Shieh, S. W. and V. D. Gligor. *Auditing the Use of Covert Storage Channels in Secure Systems*. Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1990.
- _____ and V. D. Gligor. *A Pattern-Oriented Intrusion Detection Model and Its Applications*. Proceedings of the 1991 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1991.
- Shostack, Adam and Scott Blake. *Towards a Taxonomy of Network Security Assessment Techniques*. Proceedings of 1999 Black Hat Briefings, Las Vegas, NV, July 1999.
- Sibert, W. Olin. *Auditing in a Distributed System: SunOS MLS Audit Trails*. Proceedings of the Eleventh National Computer Security Conference, Washington, DC, October 1988.
- _____. *Malicious Data and Computer Security*. Proceedings of Nineteenth National Information System Security Conference, Baltimore, MD, October 1996.
- Simonian, Richard, et al. *A Neural Network Approach Towards Intrusion Detection*. Proceedings of the Thirteenth National Computer Security Conference, Washington, DC, October 1990.
- Smaha Steve E. *An Intrusion Detection System for the Air Force*. Proceedings of the Fourth Aerospace Computer Security Applications Conference, Orlando, FL, December 1988.
- _____. *Haystack: An Intrusion Detection System*. Proceedings of the IEEE Fourth Aerospace Computer Security Applications Conference, Orlando, FL, December 1988.
- _____ and S. Snapp. *Method and System for Detecting Intrusion into and Misuse of a Data Processing System*. US555742, U.S. Patent Office, September 17, 1996.
- _____ and J. Winslow. *Misuse Detection Tools*. Computer Security Journal 10, no. 1, Spring 1994.

- Smith, C. Fred. *Some Unintended Legal Consequences of Intentional Technological Disasters*. Second Pacific Institute of Computer Security Workshop, San Diego, CA, February 1999.
- _____ and Erin Kenneally. *The Ties That Bind and Set Them Pleaing - Testimony from the Envisioned Trial of Kevin Mitnick*. Second Pacific Institute of Computer Security Workshop, San Diego, CA, February 1999.
- Snapp, Steven R., J. Brentano, G. Dias, T. Goan, T. Grance, T. Heberlein, C. Ho, K. Levitt, B. Mukhejee, D. Mansur, K. Pon, and S. Smaha. *A System for Distributed Intrusion Detection*. Proceedings of COMPCON Spring '91, San Francisco, CA, February 1991.
- _____, J. Brentano, G. Dias, T. Goan, T. Heberlein, C. Ho, K. Levitt, B. Mukherjee, S. Smaha, T. Grance, D. Teal, and D. Mansur. *DIDS (Distributed Intrusion Detection System) Motivation, Architecture, and an Early Prototype*. Proceedings of the Fourteenth National Computer Security Conference, Washington, DC, October 1991.
- _____, B. Mukherjee, and K. N. Levitt. *Detecting Intrusions Through Attack Signature Analysis*. Proceedings of the Third Workshop on Computer Security Incident Handling. Herndon, VA, August 1991.
- Sobirey, M., B. Richter, and H. Konig. *The Intrusion Detection System AID: Architecture, and Experiences in Automated Audit Analysis*. Proceedings of the IFIPTC6/TC1 1 International Conference on Communications and Multimedia Security, Essen, Germany, September 1996.
- Sommer, Peter. *Intrusion Detection Systems as Evidence*. First International Workshop on the Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium, September 1998.
- Spafford, Eugene H. The Internet Worm: Crisis and Aftermath; Communications of the ACM; 32(6): 678 - 687, June 1989.
- SRI International. *System Design Laboratory Laboratory - Intrusion Detection*. [en línea]. Fecha no disponible [consultado en enero, 2003]. Next-Generation IDES (NIDES). Disponible desde Internet <<http://www.sdl.sri.com/programs/intrusion/history.html>>.
- Staniford-Chen, Stuart, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle. *GrIDS - A Graph-Based Intrusion Detection System for Large Networks*. Nineteenth National Information Systems Security Conference, Baltimore, MD, October 1996.
- _____, and L. Todd Heberlein. *Holding Intruders Accountable on the Internet*. Proceedings of the 1995 IEEE Symposium on Security and Privacy, Oakland, CA, May 1995.
- Sundaram, Aurobindo. *An Introduction to Intrusion Detection*. Crossroads: The ACM Student Magazine 2, no.4 (April 1996) available at www.acm.org/crossroads/xrds2-4/intrus.html.
- Sytek, Inc. *Analysis of Computer System Audit Trails*. Sytek technical reports 85009, 85018, 86005, 86007, Mountain View, CA, 1985 - 1986.
- Tener, William T. *Discovery: An Expert System in the Commercial Data Security Environment*. Proceedings of the IFIP Security Conference, Monte Carlo, 1986.
- _____. *AI and 4GL: Automated Detection and Investigation and Detection Tools*. Proceedings of the IFIP Security Conference, Sydney, Australia, 1988.
- Teng, H. S., K. Chen, and S. C. Y. Lu. *Adaptive Real-Time Anomaly Detection Using Inductively Generated Sequential Patterns*. Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1990.

- _____, Kaihu Chen and Stephen C. Lu. *Security Audit Trail Analysis Using Inductively Generated Predictive Rules*. Proceedings of the 11th National Conference on Artificial Intelligence Applications, pages 24-29, IEEE, IEEE Service Center, Piscataway, NJ, March 1990.
- Ting, Christopher, T. H. Ong, Y. T. Tan, and P. Y. Ng. *Intrusion Detection, Internet Law Enforcement, and Insurance Coverage to Accelerate the Proliferation of Internet Business*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- TRW Defense Systems Group. *Intrusion Detection Expert System Feasibility Study*. Final report 46761, 1986.
- Tsudik, G. and R. Summers. *AudES - An Expert System for Security Auditing*. Proceedings of the AAAI Conference on Innovative Applications in AI, San Jose, CA, May 1990, reprinted in Computer Security Journal 6, no. 1 (1990): 89 - 93.
- United Nations Committee on Crime Prevention and Control. *International Review of Criminal Policy - United Nations Manual on the Prevention and Control of Computer Related Crime*. Revisions 43 and 44, New York, NY, 1999.
- Vaccaro, Henry S. and G. E. Liepins. *Detection of Anomalous Computer Session Activity*. Proceedings of the 1989 IEEE Symposium on Security and Privacy, Oakland, CA, May 1989.
- Valcarce, E. M., G. W. Hoglund, L. Jansen, and L. Baillie. *ESSENSE: An Experiment in Knowledge-Based Security Monitoring and Control*. Proceedings of the Third USENIX Unix Security Symposium, Baltimore, MD, September 1992.
- Vert, Greg, D. A. Frincke, and J. McConnell. *A Visual Mathematical Model for Intrusion Detection*. Proceedings of Twenty-First National Information System Security Conference, Crystal City, VA, October 1998.
- Warrender, C., S. Forrest, and B. Pearimutter. *Detecting Intrusions Using System Calls: Alternative Data Models*. Proceedings of Twenty-Fifth IEEE Symposium on Security and Privacy, Oakland, CA, May 1999.
- Wasserman, Joseph J. *The Vanishing Trail*. Bell Telephone Magazine 47, no. 4, July - August 1968: 12 - 15.
- Wee, Christopher. *LAFS: A Logging and Auditing File System*. Proceedings of the Eleventh Computer Security Applications Conference, New Orleans, LA, December 1995.
- _____. *Policy-Directed Auditing and Logging*. Ph.D. thesis, University of California, Davis, CA, April 1996.
- Weiss, Winfried R. E. and A. Baur. *Analysis of Audit and Protocol Data Using Methods from Artificial Intelligence*. Proceedings of the Thirteenth National Computer Security Conference, Washington, DC, October 1990.
- Wetmore, Brad. *Audit Browsing*. Master thesis, University of California, Davis, CA, 1993.
- White, Greg, E. A. Fisch, and U. W. Pooch. *Cooperating Security Managers: A Peer-Based Intrusion Detection System*. IEEE Network 10, no. 1: 20 - 23, January - February 1996.
- _____, and Udo Pooch. *Cooperating Security Managers: Distributed Intrusion Detection Systems*. Oxford, UK: Elsevier Science Publishers, Ltd, Computers and Security, v 15, no. 5: 441 - 450, September/October 1996.

- Winkler, J. B.. *A UNIX Prototype for Intrusion and Anomaly Detection in Secure Networks*. Proceedings of the Thirteenth National Computer Security Conference, Washington, DC, October 1990.
- _____ and W. J. Page. *Intrusion and Anomaly Detection in Trusted Systems*. Proceedings of the Fifth Annual Computer Security Applications Conference, Tucson, AZ, December 1989.
- Wood, Mark. *Intrusion Detection Exchange Format Requirements*. Internet draft, Internet Engineering Task Force, June 1999.
- Yao-Tsung Lin, Shian-Shyong Tseng And Shun-Chieh Lin. *An Intrusion Detection Model Based Upon Intrusion Detection Markup Language (IDML)*. Department of Computer and Information Science, National Chiao Tung University, Taiwan, agosto de 2001.
- Yip, Raymond and K. Levitt. *Data Level Inference Detection in Database Systems*. Proceedings of the Eleventh IEEE Computer Security Foundations Workshop, Rockport, MA, June 1998.
- _____ and K. Levitt. *The Design and Implementation of a Data Level Database Inference Detection System*. Proceedings of the Twelfth Annual IFIP WG 11.3 Working Conference on Database Security, Chalkidiki, Greece, July 1998.
- Yuill, Jim, S. F. Wu, F. Gong, and M-Y. Huang. *Intrusion Detection for an Ongoing Attack*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Zamoni, Diego M. *SMNT: A Security Analysis Integration Tool*. Systems Administration, Networking and Security (SANS) Conference, Washington, DC, May 1996.
- _____ and E. H. Spafford. *New Directions for the AAFID Architecture*. Proceedings of the Second International Workshop on Recent Advances in Intrusion Detection, W. Lafayette, IN, September 1999.
- Zerkle, Dan and K. Levitt. *NetKuang - A Multi-Host Configuration Vulnerability Checker*. Proceedings of the Sixth USENIX Security Symposium, San Jose, CA, July 1996.

Apéndice D - Normativa legal

Marco general

Nombrar toda la normativa legal que puede hacer referencia a temas de seguridad informática en España puede resultar una tarea laboriosa. A continuación se mencionan las normas más importantes.

- Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal (B.O.E. núm. 298, 14/12/1999).
- Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y Comercio Electrónico (LSSICE).
- Ley 7/1998, de 13 de abril, sobre Condiciones Generales de la Contratación (B.O.E. núm. 89, 14/4/1998).
- Real Decreto 1/1996, de 12 de abril (BOE 22-4-1996), por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual.
- Real Decreto 14/1999, de 17 de septiembre, sobre Firma Electrónica.
- Real Decreto 1906/1999, de 17 de diciembre, por el que se regula la contratación telefónica o electrónica con condiciones generales, en desarrollo del artículo 5.3 de la Ley 7/1998, de 13 de abril, de Condiciones Generales de la Contratación.
- Real Decreto 1133/1997, de 11 de julio, por el que se regula la autorización de las ventas a distancia e inscripción en el Registro de empresas de ventas a distancia.
- Directiva 98/27/CE del Parlamento Europeo y del Consejo, de 19 de mayo de 1998, relativa a las acciones de cesación en materia de protección de los intereses de los consumidores (D.O. L 166, 11/6/1998).
- Directiva 97/7/CE del Parlamento Europeo y del Consejo, de 20 de mayo, relativa a la protección de los consumidores en materia de contratos a distancia (D.O. L 144, 4/6/1997).
- Directiva 2002/65/CE del Parlamento Europeo y del Consejo, de 23 de septiembre, relativa a la comercialización a distancia de servicios financieros destinados a los consumidores, y por la que se modifican la Directiva 90/619/CEE del Consejo y las Directivas 97/7/CE y 98/27/CE.
- Directiva 95/46/CE del Parlamento Europeo y del Consejo, de 24 de octubre, relativa a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (DO L 281, 23/11/1995).
- Directiva 2002/58/CE del Parlamento Europeo y del Consejo, de 12 de julio, relativa al tratamiento de los datos personales y a la protección de la intimidad en el sector de las comunicaciones electrónicas (Directiva sobre la privacidad y las comunicaciones electrónicas) (DO L 201, 31/7/2002).
- Directiva 93/13/CEE del Consejo, de 5 de abril, sobre las cláusulas abusivas en los contratos celebrados con consumidores (D.O. L 95, 21/4/1993).

Intrusiones, ataques

Como se puede observar más abajo, algunos de los delitos informáticos tipificados en el Código Penal son aplicables en caso de intrusión o ataque. Por otra parte, de forma adicional, también conviene indicar algunas de las normas a que acudir en caso de análisis forense del sistema comprometido.

- Ley Orgánica 10/1995, de 23 de Noviembre, del Código Penal.
 - Delitos contra la confidencialidad, la integridad y la disponibilidad de los datos y sistemas informáticos.
 - Artículos 197, 264.2, 278.1, 278.3
 - Delitos informáticos.
 - Artículo 248.2, 256
 - Delitos relacionados con el contenido.
 - Artículo 186, 189
 - Delitos relacionados con infracciones de la propiedad intelectual y derechos afines.
 - Artículo 270, 273
- Ley 1/2000, de 7 de enero, de Enjuiciamiento Civil.
 - Artículos 335 a 352, "Del dictamen de peritos".
- Ley de 14 de septiembre de 1882, de Enjuiciamiento Criminal.
 - Artículos 456 a 485, "Del informe pericial".

Ley Orgánica 10/1995, de 23 de noviembre, del Código Penal

(B.O.E. 24-11-1995).

El texto que sigue es un extracto del escrito oficial, y a pesar de servir de referencia, a efectos legales no tiene valor alguno.

LIBRO II

Delitos y sus penas

TÍTULO VIII

Delitos contra la libertad e indemnidad sexuales

CAPÍTULO IV

De los delitos de exhibicionismo y provocación sexual

Artículo 186.

El que, por cualquier medio directo, vendiere, difundiere o exhibiere material pornográfico entre menores de edad o incapaces, será castigado con la pena de prisión de seis meses a un año, o multa de seis a doce meses.

CAPÍTULO V

De los delitos relativos a la prostitución y la corrupción de menores

Artículo 189.

1. Será castigado con la pena de prisión de uno a tres años:

a) El que utilizare a menores de edad o a incapaces con fines o en espectáculos exhibicionistas o pornográficos, tanto públicos como privados, o para elaborar cualquier clase de material pornográfico, o financiare cualquiera de estas actividades.

b) El que produjere, vendiere, distribuyere, exhibiere o facilitare la producción, venta, difusión o exhibición por cualquier medio de material pornográfico en cuya elaboración hayan sido utilizados menores de edad o incapaces, aunque el material tuviere su origen en el extranjero o fuere desconocido.

A quien poseyera dicho material para la realización de cualquiera de estas conductas se le impondrá la pena en su mitad inferior.

2. Se impondrá la pena superior en grado cuando el culpable perteneciere a una organización o asociación, incluso de carácter transitorio, que se dedicare a la realización de tales actividades.

3. El que haga participar a un menor o incapaz en un comportamiento de naturaleza sexual que perjudique la evolución o desarrollo de la personalidad de éste, será castigado con la pena de prisión de seis meses a un año o multa de seis a doce meses.

4. El que tuviere bajo su potestad, tutela, guarda o acogimiento, a un menor de edad o incapaz, y que, con conocimiento de su estado de prostitución o corrupción, no haga lo posible para impedir

su continuación en tal estado, o no acuda a la autoridad competente para el mismo fin si carece de medios para la custodia del menor o incapaz, será castigado con la pena de multa de seis a doce meses.

5. El Ministerio Fiscal promoverá las acciones pertinentes con objeto de privar de la patria potestad, tutela, guarda o acogimiento familiar, en su caso, a la persona que incurra en alguna de las conductas descritas en el apartado anterior.

TITULO X

Delitos contra la intimidad, el derecho a la propia imagen y la inviolabilidad del domicilio

CAPITULO I

Del descubrimiento y revelación de secretos

Artículo 197.

1. El que, para descubrir los secretos o vulnerar la intimidad de otro, sin su consentimiento, se apodere de sus papeles, cartas, mensajes de correo electrónico o cualesquiera otros documentos o efectos personales o intercepte sus telecomunicaciones o utilice artificios técnicos de escucha, transmisión, grabación o reproducción del sonido o de la imagen, o de cualquier otra señal de comunicación, será castigado con las penas de prisión de uno a cuatro años y multa de doce a veinticuatro meses. 2. Las mismas penas se impondrán al que, sin estar autorizado, se apodere, utilice o modifique, en perjuicio de tercero, datos reservados de carácter personal o familiar de otro que se hallen registrados en ficheros o soportes informáticos, electrónicos o telemáticos, o en cualquier otro tipo de archivo o registro público o privado. Iguales penas se impondrán a quien, sin estar autorizado, acceda por cualquier medio a los mismos y a quien los altere o utilice en perjuicio del titular de los datos o de un tercero.

3. Se impondrá la pena de prisión de dos a cinco años si se difunden, revelan o ceden a terceros los datos o hechos descubiertos o las imágenes captadas a que se refieren los números anteriores.

Será castigado con las penas de prisión de uno a tres años y multa de doce a veinticuatro meses, el que, con conocimiento de su origen ilícito y sin haber tomado parte en su descubrimiento, realizare la conducta descrita en el párrafo anterior.

4. Si los hechos descritos en los apartados 1 y 2 de este Artículo se realizan por las personas encargadas o responsables de los ficheros, soportes informáticos, electrónicos o telemáticos, archivos o registros, se impondrá la pena de prisión de tres a cinco años, y si se difunden, ceden o revelan los datos reservados, se impondrá la pena en su mitad superior.

5. Igualmente, cuando los hechos descritos en los apartados anteriores afecten a datos de carácter personal que revelen la ideología, religión, creencias, salud, origen racial o vida sexual, o la víctima fuere un menor de edad o un incapaz, se impondrán las penas previstas en su mitad superior.

6. Si los hechos se realizan con fines lucrativos, se impondrán las penas respectivamente previstas en los apartados 1 al 4 de este Artículo en su mitad superior. Si además afectan a datos de los mencionados en el apartado 5, la pena a imponer será la de prisión de cuatro a siete años.

Artículo 198.

La autoridad o funcionario público que, fuera de los casos permitidos por la Ley, sin mediar causa legal por delito, y prevaliéndose de su cargo, realizare cualquiera de las conductas descritas en el Artículo anterior, será castigado con las penas respectivamente previstas en el mismo, en su mitad superior y, además, con la de inhabilitación absoluta por tiempo de seis a doce años.

Artículo 199.

1. El que revelare secretos ajenos, de los que tenga conocimiento por razón de su oficio o sus relaciones laborales, será castigado con la pena de prisión de uno a tres años y multa de seis a doce meses.
2. El profesional que, con incumplimiento de su obligación de sigilo o reserva, divulgue los secretos de otra persona, será castigado con la pena de prisión de uno a cuatro años, multa de doce a veinticuatro meses e inhabilitación especial para dicha profesión por tiempo de dos a seis años.

Artículo 200.

Lo dispuesto en este capítulo será aplicable al que descubriere, revelare o cediere datos reservados de personas jurídicas, sin el consentimiento de sus representantes, salvo lo dispuesto en otros preceptos de este Código.

Artículo 201.

1. Para proceder por los delitos previstos en este capítulo será necesaria denuncia de la persona agraviada o de su representante legal. Cuando aquélla sea menor de edad, incapaz o una persona desvalida, también podrá denunciar el Ministerio Fiscal.
2. No será precisa la denuncia exigida en el apartado anterior para proceder por los hechos descritos en el Artículo 198 de este Código, ni cuando la comisión del delito afecte a los intereses generales o a una pluralidad de personas.
3. El perdón del ofendido o de su representante legal, en su caso, extingue la acción penal o la pena impuesta, sin perjuicio de lo dispuesto en el segundo párrafo del número 4.º del Artículo 130.

TITULO XIII

Delitos contra el patrimonio y contra el orden socioeconómico

CAPITULO VI

De las defraudaciones

SECCIÓN 1.ª DE LAS ESTAFAS

Artículo 248.

1. Cometen estafa los que, con ánimo de lucro, utilizaren engaño bastante para producir error en otro, induciéndolo a realizar un acto de disposición en perjuicio propio o ajeno.
2. También se consideran reos de estafa los que, con ánimo de lucro, y valiéndose de alguna manipulación informática o artificio semejante consigan la transferencia no consentida de cualquier activo patrimonial en perjuicio de tercero.

SECCIÓN 3.ª DE LAS DEFRAUDACIONES DE FLUIDO ELÉCTRICO Y ANÁLOGAS

Artículo 256.

El que hiciere uso de cualquier equipo terminal de telecomunicación, sin consentimiento de su titular, ocasionando a éste un perjuicio superior a cincuenta mil pesetas, será castigado con la pena de multa de tres a doce meses.

CAPITULO IX

De los daños

Artículo 264.

1. Será castigado con la pena de prisión de uno a tres años y multa de doce a veinticuatro meses el que causare daños expresados en el Artículo anterior, si concurriere alguno de los supuestos siguientes:

1.º Que se realicen para impedir el libre ejercicio de la autoridad o en venganza de sus determinaciones, bien se cometiere el delito contra funcionarios públicos, bien contra particulares que, como testigos o de cualquier otra manera, hayan contribuido o puedan contribuir a la ejecución o aplicación de las Leyes o disposiciones generales.

2.º Que se cause por cualquier medio infección o contagio de ganado.

3.º Que se empleen sustancias venenosas o corrosivas.

4.º Que afecten a bienes de dominio o uso público o comunal.

5.º Que arruinen al perjudicado o se le coloque en grave situación económica.

2. La misma pena se impondrá al que por cualquier medio destruya, altere, inutilice o de cualquier otro modo dañe los datos, programas o documentos electrónicos ajenos contenidos en redes, soportes o sistemas informáticos.

CAPITULO XI

De los delitos relativos a la propiedad intelectual e industrial, al mercado y a los consumidores

SECCIÓN 1.ª DE LOS DELITOS RELATIVOS A LA PROPIEDAD INTELECTUAL

Artículo 270.

Será castigado con la pena de prisión de seis meses a dos años o de multa de seis a veinticuatro meses quien, con ánimo de lucro y en perjuicio de tercero, reproduzca, plagie, distribuya o comunique públicamente, en todo o en parte, una obra literaria, artística o científica, o su transformación, interpretación o ejecución artística fijada en cualquier tipo de soporte o comunicada a través de cualquier medio, sin la autorización de los titulares de los correspondientes derechos de propiedad intelectual o de sus cesionarios.

La misma pena se impondrá a quien intencionadamente importe, exporte o almacene ejemplares de dichas obras o producciones o ejecuciones sin la referida autorización.

Será castigada también con la misma pena la fabricación, puesta en circulación y tenencia de cualquier medio específicamente destinada a facilitar la supresión no autorizada o la neutralización de cualquier dispositivo técnico que se haya utilizado para proteger programas de ordenador.

SECCIÓN 2.ª DE LOS DELITOS RELATIVOS A LA PROPIEDAD INDUSTRIAL

Artículo 273.

1. Será castigado con las penas de prisión de seis meses a dos años y multa de seis a veinticuatro meses el que, con fines industriales o comerciales, sin consentimiento del titular de una patente o modelo de utilidad y con conocimiento de su registro, fabrique, importe, posea, utilice, ofrezca o introduzca en el comercio objetos amparados por tales derechos.

2. Las mismas penas se impondrán al que, de igual manera, y para los citados fines, utilice u ofrezca la utilización de un procedimiento objeto de una patente, o posea, ofrezca, introduzca en el comercio, o utilice el producto directamente obtenido por el procedimiento patentado.
3. Será castigado con las mismas penas el que realice cualquiera de los actos tipificados en el párrafo primero de este Artículo concurriendo iguales circunstancias en relación con objetos amparados en favor de tercero por un modelo o dibujo industrial o artístico o topografía de un producto semiconductor.

SECCIÓN 3.^a DE LOS DELITOS RELATIVOS AL MERCADO Y A LOS CONSUMIDORES

Artículo 278.

1. El que, para descubrir un secreto de empresa se apoderare por cualquier medio de datos, documentos escritos o electrónicos, soportes informáticos u otros objetos que se refieran al mismo, o empleare alguno de los medios o instrumentos señalados en el apartado 1 del Artículo 197, será castigado con la pena de prisión de dos a cuatro años y multa de doce a veinticuatro meses.
2. Se impondrá la pena de prisión de tres a cinco años y multa de doce a veinticuatro meses si se difundieren, revelaren o cedieren a terceros los secretos descubiertos.
3. Lo dispuesto en el presente Artículo se entenderá sin perjuicio de las penas que pudieran corresponder por el apoderamiento o destrucción de los soportes informáticos.

Ley 1/2000, de 7 de enero, de Enjuiciamiento Civil

(BOE núm. 7, de 8 de enero del 2000, pp. 575-728. Corrección de errores BOE núm. 90, de 14-04-2000, p. 15278 y BOE núm. 180, de 28-07-2001, p. 27746).

[Modificada por la Ley 39/2002, de 28 de octubre, de transposición al ordenamiento jurídico español de diversas directivas comunitarias en materia de protección de los intereses de los consumidores y usuarios (BOE núm. 259, de 29-10-2002, pp. 37922-37933). Esta modificación afecta a los artículos 6, 11, 15, 52, 221, 249, 250, 711 y 728.]

El texto que sigue es un extracto del escrito oficial, y a pesar de servir de referencia, a efectos legales no tiene valor alguno.

LIBRO II

De los procesos declarativos

TÍTULO I

De las disposiciones comunes a los procesos declarativos

CAPÍTULO VI

De los medios de prueba y las presunciones

SECCIÓN 5.^a DEL DICTAMEN DE PERITOS

Artículo 335. Objeto y finalidad del dictamen de peritos. Juramento o promesa de actuar con objetividad.

1. Cuando sean necesarios conocimientos científicos, artísticos, técnicos o prácticos para valorar hechos o circunstancias relevantes en el asunto o adquirir certeza sobre ellos, las partes podrán aportar al proceso el dictamen de peritos que posean los conocimientos correspondientes o solicitar, en los casos previstos en esta ley, que se emita dictamen por perito designado por el tribunal.

2. Al emitir el dictamen, todo perito deberá manifestar, bajo juramento o promesa de decir verdad, que ha actuado y, en su caso, actuará con la mayor objetividad posible, tomando en consideración tanto lo que pueda favorecer como lo que sea susceptible de causar perjuicio a cualquiera de las partes, y que conoce las sanciones penales en las que podría incurrir si incumpliere su deber como perito.

Artículo 336. Aportación con la demanda y la contestación de dictámenes elaborados por peritos designados por las partes.

1. Los dictámenes de que los litigantes dispongan, elaborados por peritos por ellos designados, y que estimen necesarios o convenientes para la defensa de sus derechos, habrán de aportarlos con la demanda o con la contestación, si ésta hubiere de realizarse en forma escrita, sin perjuicio de lo dispuesto en el artículo 337 de la presente Ley.

2. Los dictámenes se formularán por escrito, acompañados, en su caso, de los demás documentos, instrumentos o materiales adecuados para exponer el parecer del perito sobre lo que haya sido objeto de la pericia.

Si no fuese posible o conveniente aportar estos materiales e instrumentos, el escrito de dictamen contendrá sobre ellos las indicaciones suficientes. Podrán, asimismo, acompañarse al dictamen los documentos que se estimen adecuados para su más acertada valoración.

3. Se entenderá que al demandante le es posible aportar con la demanda dictámenes escritos elaborados por perito por él designado, si no justifica cumplidamente que la defensa de su derecho no ha permitido demorar la interposición de aquella hasta la obtención del dictamen.

4. En los juicios con contestación a la demanda por escrito, el demandado que no pueda aportar dictámenes escritos con aquella contestación a la demanda deberá justificar la imposibilidad de pedirlos y obtenerlos dentro del plazo para contestar.

Artículo 337. Anuncio de dictámenes cuando no se puedan aportar con la demanda o con la contestación. Aportación posterior.

1. Si no les fuese posible a las partes aportar dictámenes elaborados por peritos por ellas designados, junto con la demanda o contestación, expresarán en una u otra los dictámenes de que, en su caso, pretendan valerse, que habrán de aportar, para su traslado a la parte contraria, en cuanto dispongan de ellos, y en todo caso antes de iniciarse la audiencia previa al juicio ordinario o antes de la vista en el verbal.

2. Aportados los dictámenes conforme a lo dispuesto en el apartado anterior, las partes habrán de manifestar si desean que los peritos autores de los dictámenes comparezcan en el juicio regulado en los artículos 43 1 y siguientes de esta Ley o, en su caso, en la vista del juicio verbal, expresando SI deberán exponer o explicar el dictamen o responder a preguntas, objeciones o propuestas de rectificación o intervenir de cualquier otra forma útil para entender y valorar el dictamen en relación con lo que sea objeto del pleito.

Artículo 338. Aportación de dictámenes en función de actuaciones procesales posteriores a la demanda. Solicitud de intervención de los peritos en el juicio o vista.

1. Lo dispuesto en el artículo anterior no será de aplicación a los dictámenes cuya necesidad o utilidad se ponga de manifiesto a causa de alegaciones del demandado en la contestación a la demanda o de las alegaciones o pretensiones complementarias admitidas en la audiencia, a tenor del artículo 426 de esta Ley.

2. Los dictámenes cuya necesidad o utilidad venga suscitada por la contestación a la demanda o por lo alegado y pretendido en la audiencia previa al juicio se aportarán por las partes, para su traslado a las contrarias, con al menos cinco días de antelación a la celebración del juicio o de la vista, en los juicios verbales, manifestando las partes al tribunal si consideran necesario que concurran a dichos juicio o vista los peritos autores de los dictámenes, con expresión de lo que se señala en el apartado 2 del artículo 337.

El tribunal podrá acordar también en este caso la presencia de los peritos en el juicio o vista en los términos señalados en el apartado 2 del artículo anterior.

Artículo 339. Solicitud de designación de peritos por el tribunal y resolución judicial sobre dicha solicitud. Designación de peritos por el tribunal, sin instancia de parte.

1. Si cualquiera de las partes fuese titular del derecho de asistencia jurídica gratuita, no tendrá que aportar con la demanda o la contestación el dictamen pericial, sino simplemente anunciarlo, a los efectos de que se proceda a la designación judicial de perito, conforme a lo que se establece en la Ley de Asistencia Jurídica Gratuita.

2. El demandante o el demandado, aunque no se hallen en el caso del apartado anterior, también podrán solicitar en sus respectivos escritos iniciales que se proceda a la designación judicial de perito, si entienden conveniente o necesario para sus intereses la emisión de informe pericial. En

tal caso, el tribunal procederá a la designación, siempre que considere pertinente y útil el dictamen pericial solicitado. Dicho dictamen será a costa de quien lo haya pedido, sin perjuicio de lo que pudiere acordarse en materia de costas.

Salvo que se refiera a alegaciones o pretensiones no contenidas en la demanda, no se podrá solicitar, con posterioridad a la demanda o a la contestación, informe pericial elaborado por perito designado judicialmente.

La designación judicial de perito deberá realizarse en el plazo de cinco días desde la presentación de la contestación a la demanda, con independencia de quien haya solicitado dicha designación. Cuando ambas partes la hubiesen pedido inicialmente, el tribunal podrá designar, si aquéllas se muestran conformes, un único perito que emita el informe solicitado. En tal caso, el abono de los honorarios del perito corresponderá realizarlo a ambos litigantes por partes iguales, sin perjuicio de lo que pudiere acordarse en materia de costas.

3. En el juicio ordinario, si, a consecuencia de las alegaciones o pretensiones complementarias permitidas en la audiencia, las partes solicitasen, conforme previene el apartado cuarto del artículo 427, la designación por el tribunal de un perito que dictamine, lo acordará éste así, siempre que considere pertinente y útil el dictamen, y ambas partes se muestren conformes en el objeto de la pericia y en aceptar el dictamen del perito que el tribunal nombre.

Lo mismo podrá hacer el tribunal cuando se trate de juicio verbal y las partes solicitasen designación de perito, con los requisitos del párrafo anterior.

4. En los casos señalados en los dos apartados anteriores, si las partes que solicitasen la designación de un perito por el tribunal estuviesen además de acuerdo en que el dictamen sea emitido por una determinada persona o entidad, así lo acordará el tribunal. Si no hubiese acuerdo de las partes, el perito será designado por el procedimiento establecido en el artículo 341.

5. El tribunal podrá, de oficio, designar perito cuando la pericia sea pertinente en procesos sobre declaración o impugnación de la filiación, paternidad y maternidad, sobre la capacidad de las personas o en procesos matrimoniales.

6. El tribunal no designará más que un perito titular por cada cuestión o conjunto de cuestiones que hayan de ser objeto de pericia y que no requieran, por la diversidad de su materia, el parecer de expertos distintos.

Artículo 340. Condiciones de los peritos.

1. Los peritos deberán poseer el título oficial que corresponda a la materia objeto del dictamen ya la naturaleza de éste. Si se tratare de materias que no estén comprendidas en títulos profesionales oficiales, habrán de ser nombrados entre personas entendidas en aquellas materias.

2. Podrá asimismo solicitarse dictamen de Academias e instituciones culturales y científicas que se ocupen del estudio de las materias correspondientes al objeto de la pericia. También podrán emitir dictamen sobre cuestiones específicas las personas jurídicas legalmente habilitadas para ello.

3. En los casos del apartado anterior, la institución a la que se encargue el dictamen expresará a la mayor brevedad qué persona o personas se encargarán directamente de prepararlo, a las que se exigirá el juramento o promesa previsto en el apartado segundo del artículo 335.

Artículo 341. Procedimiento para la designación judicial de perito.

1. En el mes de enero de cada año se interesará de los distintos Colegios profesionales o, en su defecto, de entidades análogas, así como de las Academias e instituciones culturales y científicas a que se refiere el apartado segundo del artículo anterior el envío de una lista de colegiados o asociados dispuestos a actuar como peritos. La primera designación de cada lista se efectuará por

sorteo realizado en presencia del Secretario Judicial, y a partir de ella se efectuarán las siguientes designaciones por orden correlativo.

2. Cuando haya de designarse perito a persona sin título oficial, práctica o entendida en la materia, previa citación de las partes, se realizará la designación por el procedimiento establecido en el apartado anterior, usándose para ello una lista de personas que cada año se solicitará de sindicatos, asociaciones y entidades apropiadas, y que deberá estar integrada por al menos cinco de aquellas personas. Si, por razón de la singularidad de la materia de dictamen, únicamente se dispusiera del nombre de una persona entendida o práctica, se recabará de las partes su consentimiento y sólo si todas lo otorgan se designará perito a esa persona.

Artículo 342. Llamamiento al perito designado, aceptación y nombramiento. Provisión de fondos.

1. En el plazo de cinco días desde la designación, se comunicará ésta al perito titular, requiriéndole para que, dentro de otros cinco días, manifieste si acepta el cargo. En caso afirmativo, se efectuará el nombramiento y el perito hará, en la forma en que se disponga, la manifestación bajo juramento o promesa que ordena el apartado 2 del artículo 335.

2. Si el perito designado adujere justa causa que le impidiere la aceptación, y el tribunal la considerare suficiente, será sustituido por el siguiente de la lista, y así sucesivamente, hasta que se pudiere efectuar el nombramiento.

3. El perito designado podrá solicitar, en los tres días siguientes a su nombramiento, la provisión de fondos que considere necesaria, que será a cuenta de la liquidación final. El tribunal, mediante providencia, decidirá sobre la provisión solicitada y ordenará a la parte o partes que hubiesen propuesto la prueba pericial y no tuviesen derecho a la asistencia jurídica gratuita, que procedan a abonar la cantidad fijada en la Cuenta de

Depósitos y Consignaciones del tribunal, en el plazo de cinco días.

Transcurrido dicho plazo, si no se hubiere depositado la cantidad establecida, el perito quedará eximido de emitir el dictamen, sin que pueda procederse a una nueva designación.

Cuando el perito designado lo hubiese sido de común acuerdo, y uno de los litigantes no realizare la parte de la consignación que le correspondiere, se ofrecerá al otro litigante la posibilidad de completar la cantidad que faltare, indicando en tal caso los puntos sobre los que deba pronunciarse el dictamen, o de recuperar la cantidad depositada, en cuyo caso se aplicará lo dispuesto en el párrafo anterior.

Artículo 343. Tachas de los peritos. Tiempo y forma de las tachas.

1. Sólo podrán ser objeto de recusación los peritos designados judicialmente.

En cambio, los peritos no recusables podrán ser objeto de tacha cuando concurra en ellos alguna de las siguientes circunstancias:

1.º Ser cónyuge o pariente por consanguinidad o afinidad, dentro del cuarto grado civil de una de las partes o de sus abogados o procuradores.

2.º Tener interés directo o indirecto en el asunto o en otro semejante.

3.º Estar o haber estado en situación de dependencia o de comunidad o contraposición de intereses con alguna de las partes o con sus abogados o procuradores.

4.º Amistad íntima o enemistad con cualquiera de las partes o sus procuradores o abogados.

5.º Cualquier otra circunstancia, debidamente acreditada, que les haga desmerecer en el concepto profesional.

2. Las tachas no podrán formularse después del juicio o de la vista, en los juicios verbales. Si se tratare de juicio ordinario, las tachas de los peritos autores de dictámenes aportados con demanda o contestación se propondrán en la audiencia previa al juicio.

Al formular tachas de peritos, se podrá proponer la prueba conducente a justificarlas, excepto la testifical.

Artículo 344. Contradicción y valoración de la tacha. Sanción en caso de tacha temeraria o desleal.

1. Cualquier parte interesada podrá dirigirse al tribunal a fin de negar o contradecir la tacha, aportando los documentos que consideren pertinentes a tal efecto.

Si la tacha menoscabara la consideración profesional o personal del perito, podrá éste solicitar del tribunal que, al término del proceso, declare, mediante providencia, que la tacha carece de fundamento.

2. Sin más trámites, el tribunal tendrá en cuenta la tacha y su eventual negación o contradicción en el momento de valorar la prueba, formulando, en su caso, mediante providencia, la declaración de falta de fundamento de la tacha prevista en el apartado anterior. Si apreciase temeridad o deslealtad procesal en la tacha, a causa de su motivación o del tiempo en que se formulara, podrá imponer a la parte responsable, con previa audiencia, una multa de diez mil a cien mil pesetas.

Artículo 345. Operaciones periciales y posible intervención de las partes en ellas.

1. Cuando la emisión del dictamen requiera algún reconocimiento de lugares, objetos o personas o la realización de operaciones análogas, las partes y sus defensores podrán presenciar uno y otras, si con ello no se impide o estorba la labor del perito y se puede garantizar el acierto e imparcialidad del dictamen.

2. Si alguna de las partes solicitare estar presente en las operaciones periciales del apartado anterior, el tribunal decidirá lo que proceda y, en caso de admitir esa presencia, ordenará al perito que dé aviso directamente a las partes, con antelación de al menos cuarenta y ocho horas, del día, hora y lugar en que aquellas operaciones se llevarán a cabo.

Artículo 346. Emisión y ratificación del dictamen por el perito que el tribunal designe.

El perito que el tribunal designe emitirá por escrito su dictamen, que hará llegar al tribunal en el plazo que se le haya señalado. De dicho dictamen se dará traslado a las partes por si consideran necesario que el perito concurra al juicio o a la vista a los efectos de que aporte las aclaraciones o explicaciones que sean oportunas. El tribunal podrá acordar, en todo caso, mediante providencia, que considera necesaria la presencia del perito en el juicio o la vista para comprender y valorar mejor el dictamen realizado.

Artículo 347. Posible actuación de los peritos en el juicio o en la vista.

1. Los peritos tendrán en el juicio o en la vista la intervención solicitada por las partes, que el tribunal admita.

El tribunal sólo denegará las solicitudes de intervención que, por su finalidad y contenido, hayan de estimarse impertinentes o inútiles.

En especial, las partes y sus defensores podrán pedir:

1.º Exposición completa del dictamen, cuando esa exposición requiera la realización de otras operaciones, complementarias del escrito aportado, mediante el empleo de los documentos, materiales y otros elementos a que se refiere el apartado 2 del artículo 336.

2.º Explicación del dictamen o de alguno o algunos de sus puntos, cuyo significado no se considerase suficientemente expresivo a los efectos de la prueba.

3.º Respuestas a preguntas y objeciones, sobre método, premisas, conclusiones y otros aspectos del dictamen.

4.º Respuestas a solicitudes de ampliación del dictamen a otros puntos conexos, por si pudiera llevarse a cabo en el mismo acto y a efectos, en cualquier caso, de conocer la opinión del perito sobre la posibilidad y utilidad de la ampliación, así como del plazo necesario para llevarla a cabo.

5.º Crítica del dictamen de que se trate por el perito de la parte contraria.

6.º Formulación de las tachas que pudieren afectar al perito.

2. El tribunal podrá también formular preguntas a los peritos y requerir de ellos explicaciones sobre lo que sea objeto del dictamen aportado, pero sin poder acordar, de oficio, que se amplíe, salvo que se trate de peritos designados de oficio conforme a lo dispuesto en el apartado 5 del artículo 339.

Artículo 348. Valoración del dictamen pericial.

El tribunal valorará los dictámenes periciales según las reglas de la sana crítica.

Artículo 349. Cotejo de letras.

1. Se practicará por perito el cotejo de letras cuando la autenticidad de un documento privado se niegue o se ponga en duda por la parte a quien perjudique.

2. También podrá practicarse cotejo de letras cuando se niegue o discuta la autenticidad de cualquier documento público que carezca de matriz y de copias fehacientes según lo dispuesto en el artículo 122 1 del Código Civil, siempre que dicho documento no pueda ser reconocido por el funcionario que lo hubiese expedido o por quien aparezca como fedatario interviniente.

3. El cotejo de letras se practicará por perito designado por el tribunal conforme a lo dispuesto en los artículos 341 y 342 de esta Ley.

Artículo 350. Documentos indubitados o cuerpo de escritura para el cotejo.

1. La parte que solicite el cotejo de letras designará el documento o documentos indubitados con que deba hacerse.

2. Se considerarán documentos indubitados a los efectos de cotejar las letras:

1.º Los documentos que reconozcan como tales todas las partes a las que pueda afectar esta prueba pericial.

2.º Las escrituras públicas y los que consten en los archivos públicos relativos al Documento Nacional de Identidad:

3.º Los documentos privados cuya letra o firma haya sido reconocida en juicio por aquel a quien se atribuya la dudosa.

4.º El escrito impugnado, en la parte en que reconozca la letra como suya aquel a quien perjudique.

3. A falta de los documentos enumerados en el apartado anterior, la parte a la que se atribuya el documento impugnado o la firma que lo autorice podrá ser requerida, a instancia de la contraria, para que forme un cuerpo de escritura que le dictará el tribunal o el Secretario Judicial.

Si el requerido se negase, el documento impugnado se considerará reconocido.

4. Si no hubiese documentos indubitados y fuese imposible el cotejo con un cuerpo de escritura por fallecimiento o ausencia de quien debiera formarlo, el tribunal apreciará el valor del documento impugnado conforme a las reglas de la sana crítica.

Artículo 351. Producción y valoración del dictamen sobre el cotejo de letras.

1. El perito que lleve a cabo el cotejo de letras consignará por escrito las operaciones de comprobación y sus resultados.
2. Será de aplicación al dictamen pericial de cotejo de letras lo dispuesto en los artículos 346, 347 y 348 de esta Ley.

Artículo 352. Otros dictámenes periciales instrumentales de pruebas distintas.

Cuando sea necesario o conveniente para conocer el contenido o sentido de una prueba o para proceder a su más acertada valoración, podrán las partes aportar o proponer dictámenes periciales sobre otros medios de prueba admitidos por el tribunal al amparo de lo previsto en los apartados 2 y 3 del artículo 299.

Ley de 14 de septiembre de 1882, de Enjuiciamiento Criminal.

El texto que sigue es un extracto del escrito oficial, y a pesar de servir de referencia, a efectos legales no tiene valor alguno.

LIBRO II

TITULO V

De la comprobación del delito y averiguación del delincuente

CAPITULO VII

Del informe pericial

456. El Juez acordará el informe pericial cuando, para conocer o apreciar algún hecho o circunstancia importante en el sumario, fuesen necesarios o convenientes conocimientos científicos o artísticos.

457. Los peritos pueden ser o no titulares.

Son peritos titulares los que tienen título oficial de una ciencia o arte cuyo ejercicio esté reglamentado por la Administración.

Son peritos no titulares los que, careciendo de título oficial, tienen, sin embargo, conocimientos o práctica especiales en alguna ciencia o arte.

458. El Juez se valdrá de peritos titulares con preferencia a los que no tuviesen título.

459. Todo reconocimiento pericial se hará por dos peritos.

Se exceptúa el caso en que no hubiese más de uno en el lugar y no fuere posible esperar la llegada de otro sin graves inconvenientes para el curso del sumario.

460. El nombramiento se hará saber a los peritos por medio de oficio, que les será entregado por alguacil o portero del Juzgado, con las formalidades prevenidas para la citación de los testigos, reemplazándose la cédula original, para los efectos del artículo 175, por un atestado que extenderá el alguacil o portero encargado de la entrega.

461. Si la urgencia del encargo lo exige, podrá hacerse el llamamiento verbalmente de orden del Juez, haciéndolo constar así en los autos; pero extendiendo siempre el atestado prevenido en el artículo anterior el encargado del cumplimiento de la orden de llamamiento.

462. Nadie podrá negarse a acudir al llamamiento del Juez para desempeñar un servicio pericial, si no estuviere legítimamente impedido.

En este caso deberá ponerlo en conocimiento del Juez en el acto de recibir el nombramiento, para que se provea a lo que haya lugar.

463. El perito, que sin alegar excusa fundada, deje de acudir al llamamiento del Juez o se niegue a prestar el informe, incurrirá en las responsabilidades señaladas para los testigos en el artículo 420.

464. No podrán prestar informe pericial acerca del delito, cualquiera que sea la persona ofendida, los que según el artículo 416 no están obligados a declarar como testigos.

El perito que, hallándose comprendido en alguno de los casos de dicho artículo, preste el informe sin poner antes esa circunstancia en conocimiento del Juez que le hubiese nombrado incurrirá en la multa de 200 a 5.000 euros, a no ser que el hecho diere lugar a responsabilidad criminal.

465. Los que presten informe como peritos en virtud de orden judicial tendrán derecho a reclamar los honorarios e indemnizaciones que sean justos, si no tuvieren en concepto de tales peritos, retribución fija satisfecha por el Estado, por la Provincia o por el Municipio.

466. Hecho el nombramiento de peritos, se notificará inmediatamente así al actor particular, si lo hubiere, como al procesado si estuviere a disposición del Juez o se encontrare en el mismo lugar de la instrucción, o a su representante si le tuviere.

467. Si el reconocimiento e informe periciales pudieren tener lugar de nuevo en el juicio oral, los peritos nombrados no podrán ser recusados por las partes.

Si no pudiere reproducirse en el juicio oral, habrá lugar a la recusación.

468. Son causa de recusación de los peritos:

1ª) El parentesco de consanguinidad o de afinidad dentro del cuarto grado con el querellante o con el reo.

2ª) El interés directo o indirecto en la causa o en otra semejante.

3ª) La amistad íntima o la enemistad manifiesta.

469. El actor o procesado que intente recusar al perito o peritos nombrados por el Juez deberá hacerlo por escrito antes de empezar la diligencia pericial, expresando la causa de la recusación y la prueba testifical que ofrezca, y acompañando la documental o designando el lugar en que ésta se halle si no la tuviere a su disposición.

Para la presentación de este escrito, no estará obligado a valerse de Procurador.

470. El Juez, sin levantar mano, examinará los documentos que produzca el recusante y oír a los testigos que presente en el acto, resolviendo lo que estime justo respecto de la recusación.

Si hubiere lugar a ella, suspenderá el acto pericial por el tiempo estrictamente necesario para nombrar el perito que haya de sustituir al recusado, hacérselo saber y constituirse el nombrado en el lugar correspondiente.

Si no la admitiere, se procederá como si no se hubiese usado de la facultad de recusar.

Cuando el recusante no produjese los documentos, pero designare el archivo o lugar en que se encuentren, el Juez instructor los reclamará y examinará una vez recibidos sin detener por esto el curso de las actuaciones; y si de ellos resultase justificada la causa de la recusación, anulará el informe pericial que se hubiese dado, mandando que se practique de nuevo esta diligencia.

471. En el caso del párrafo segundo del artículo 467, el querellante tendrá derecho a nombrar a su costa un perito que intervenga en el acto pericial.

El mismo derecho tendrá el procesado.

Si los querellantes o los procesados fuesen varios, se pondrán respectivamente de acuerdo entre sí para hacer el nombramiento.

Estos peritos deberán ser titulares, a no ser que no los hubiere de esta clase en el partido o demarcación, en cuyo caso podrán ser nombrados sin título.

Si la práctica de la diligencia pericial no admitiere espera, se procederá como las circunstancias lo permitan para que el actor y el procesado puedan intervenir en ella.

472. Si las partes hicieren uso de la facultad que se les concede en el artículo anterior, manifestarán al Juez el nombre del perito, y ofrecerán, al hacer esta manifestación, los comprobantes de tener la cualidad de tal perito la persona designada.

En ningún caso podrán hacer uso de dicha facultad después de empezada la operación de reconocimiento.

473. El Juez resolverá sobre la admisión de dichos peritos en la forma determinada en el artículo 470 para las recusaciones.

474. Antes de darse principio al acto pericial, todos los peritos, así los nombrados por el Juez como los que lo hubieren sido por las partes, prestarán juramento, conforme al artículo 434, de proceder bien y fielmente en sus operaciones, y de no proponerse otro fin más que el de descubrir y declarar la verdad.

475. El Juez manifestará clara y determinadamente a los peritos el objeto de su informe.

476. Al acto pericial podrán concurrir, en el caso del párrafo 2º artículo 467, el querellante, si lo hubiere, con su representación, y el procesado con la suya aún cuando estuviere preso, en cuyo caso adoptará el Juez las precauciones oportunas.

477. El acto pericial será presidido por el Juez instructor o, en virtud de su delegación, por el Juez municipal. Podrá también delegar en el caso del artículo 353 en un funcionario de Policía judicial.

Asistirá siempre el Secretario que actúe en la causa.

478. El informe pericial comprenderá, si fuere posible:

1º) Descripción de la persona o cosa que sea objeto del mismo, en el estado o del modo en que se halle.

El Secretario extenderá esta descripción, dictándola los peritos y suscribiéndola todos los concurrentes.

2º) Relación detallada de todas las operaciones practicadas por los peritos y de su resultado, extendida y autorizada en la misma forma que la anterior.

3º) Las conclusiones que en vista de tales datos formulen los peritos, conforme a los principios y reglas de su ciencia o arte.

479. Si los peritos tuvieren necesidad de destruir o alterar los objetos que analicen, deberán conservarse, a ser posible, parte de ellos en poder del Juez para que, en caso necesario, pueda hacerse nuevo análisis.

480. Las partes que asistieren a las operaciones o reconocimientos podrán someter a los peritos las observaciones que estimen convenientes, haciéndose constar todas en la diligencia.

481. Hecho el reconocimiento, podrán los peritos, si lo pidieren, retirarse por el tiempo absolutamente preciso al sitio que el Juez les señale para deliberar y redactar las conclusiones.

482. Si los peritos necesitaren descanso, el Juez o el funcionario que le represente podrá concederles para ello el tiempo necesario.

También podrá suspender la diligencia hasta otra hora u otro día, cuando lo exigiere su naturaleza.

En este caso, el Juez o quien lo represente adoptará las precauciones convenientes para evitar cualquier alteración en la materia de la diligencia pericial.

483. El Juez podrá, por su propia iniciativa o por reclamación de las partes presentes o de sus defensores, hacer a los peritos, cuando produzcan sus conclusiones, las preguntas que estime pertinentes y pedirles las aclaraciones necesarias.

Las contestaciones de los peritos se considerarán como parte de su informe.

484. Si los peritos estuvieren discordes y su número fuere par, nombrará otro el Juez.

Con intervención del nuevamente nombrado, se repetirán, si fuere posible, las operaciones que hubiesen practicado aquéllos y se ejecutarán las demás que parecieren oportunas.

Si no fuere posible la repetición de las operaciones ni la práctica de otras nuevas, la intervención del perito últimamente nombrado se limitará a deliberar con los demás, con vista de las diligencias de reconocimiento practicadas, y a formular luego con quien estuviere conforme, o separadamente si no lo estuviere con ninguno, sus conclusiones motivadas.

485. El Juez facilitará a los peritos los medios materiales necesarios para practicar la diligencia que les encomiende, reclamándolos de la Administración pública, o dirigiendo a la Autoridad correspondiente un aviso previo si existieren preparados para tal objeto, salvo lo dispuesto especialmente en el artículo 362.

Apéndice E - Recursos

Libros

Encontrar libros sobre seguridad informática no siempre es tarea fácil. En esta sección se incluye un conjunto de referencias que pueden servir de punto de partida. Se podrán encontrar entre ellas tanto escritos sobre Detección de Intrusiones como sobre seguridad en general.

Detección de Intrusiones y tecnologías similares

Amoroso, Edward G. *Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Trace Back, Traps, and Response*. Intrusion.Net Books, febrero de 1999.

Bace, Rebecca. *Intrusion Detection*. Macmillan Technical Publishing, 2000.

Caswell, Brian , Jay Beale, James C. Foster, Jeremy Faircloth. *Snort 2.0 Intrusion Detection*. Syngress, 2003.

Escamilla, Terry. *Intrusion Detection: Network Security Beyond the Firewall*. John Wiley and Sons, 1998.

Freiss, Martin and R. Bach. *Protecting Networks with Satan: Internet Security for System Administrators*. O'Reilly and Associates, 1998.

Lance Spitzner. *Honeypots: Tracking Hackers*. Addison Wesley Professional, 2002.

Murray, James D. and D. Russell (ed.). *Windows NT Event Logging*. O'Reilly and Associates, 1998.

Northcutt, Stephen. *Network Intrusion Detection: An Analysts' Handbook*. Que, 1999.

Northcutt, Stephen and Judy Novak. *Network Intrusion Detection*. Que, 2002.

Northcutt, Stephen, Lenny Zeltser, Scott Winters, Karen Fredrick, Ronald W. Ritchey. *Inside Network Perimeter Security: The Definitive Guide to Firewalls, Virtual Private Networks (VPNs), Routers, and Intrusion Detection Systems*. Que, 2002.

Northcutt, Stephen, Mark Cooper, Matt Fearnow, Karen Frederick. *Intrusion Signatures and Analysis*. Que, 2001.

Proctor, Paul E. *Practical Intrusion Detection Handbook*. Prentice Hall, 2000.

The Honeynet Project. *Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community*. Addison-Wesley Pub Co, 2001.

Seguridad general

Atkins, Derek. *Internet Security: Professional Reference*. New Riders Press, 1997.

Garfinkel, Simson and E. H. Spafford. *Practical UNIX and Internet Security*. O'Reilly and Associates, 1996.

Gollmann, Dieter. *Computer Security*. John Wiley & Son Ltd, 1999.

Jumes, James and Coopers and Lybrand. *Microsoft Windows NT 4.0 Security, Audit, and Control*. Microsoft Press, 1998.

Kaufman, Charlie, R. Perlman, M. Speciner, C. Kaufman. *Network Security: Private Communication in a Public World*. Prentice Hall, 2002.

Mann, Scott, E. L. Mitchell. *Linux System Security: The Administrator's Guide to Open Source Security Tools*. Prentice Hall, 1999.

Pfleeger, Charles P. *Security in Computing*. Prentice Hall, 2002.

Pipkin, Donald, Donald L. Pipkin. *Information Security: Protecting the Global Enterprise*. Prentice Hall, 2000.

Pooch, Udo and Gregory White. *Computer System and Network Security*. CRC Press, 1995.

Russell, Deborah. *Computer Security Basics*. O'Reilly and Associates, 1991.

Criptografía

Ferguson, Niels, Bruce Schneier. *Practical Cryptography*. John Wiley & Sons, 2003.

Garfinkel, Simson. *PGP: Pretty Good Privacy*. O'Reilly and Associates, 1995.

Menezes, Alfred J., Paul C. Van Oorschot, Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

Schneier, Bruce. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley and Sons, 1995.

Schneier, Bruce. *Secrets and Lies: Digital Security in a Networked World*. John Wiley & Sons, 2000.

Stallings, William. *Cryptography and Network Security: Principles and Practice*. Prentice Hall, 2002.

Análisis forense

Casey, Eoghan. *Digital Evidence and Computer Crime*. Academic Press, 2000.

Casey, Eoghan. *Handbook of Computer Crime Investigation: Forensic Tools & Technology*. Academic Press, 2001.

Kruse II, Warren G., Jay G. Heiser. *Computer Forensics: Incident Response Essentials*. Addison-Wesley Pub Co., 2001.

Marcella, Albert J., R. S. Greenfield. *Cyber Forensics: A Field Manual for Collecting, Examining, and Preserving Evidence of Computer Crimes*. Auerbach Publications, 2002.

Prosis, Chris, Kevin Mandia. *Incident Response: Investigating Computer Crime*. McGraw-Hill Osborne Media, 2001.

Vacca, John R., Michael Erbschloe. *Computer Forensics: Computer Crime Scene Investigation*. Charles River Media, 2002.

Casos concretos de aplicación

Garfinkel, Simson and E. H. Spafford. *Web Security and Commerce*. O'Reilly and Associates, 2002.

Ghosh, Anup K. *B-Commerce Security: Weak Links, Best Defenses*. John Wiley and Sons, 1998.

McGraw, Gary and E. Felten. *Securing Java: Getting Down to Business with Mobile Code*. John Wiley and Sons, 1999.

Historias sobre seguridad

Freedman, David and C. Mann. *At Large: The Strange Case of the World's Biggest Internet Invasion*. Touchstone Books, 1998.

Slatala, Michelle. and J. Quittner. *Masters of Deception: The Gang That Ruled Cyberspace*. Perennial, 1996.

Stoll, Clifford. *The Cuckoo's Egg*. Pocket Books, 2000.

Recursos WWW

(revisados en junio de 2003)

El mundo de la seguridad de las tecnologías de la información es muy dinámico y exigente. Los expertos necesitan utilizar medios de comunicación capaces de adaptarse a estos cambios. Los recursos de Internet son perfectos para esta labor. Aunque las referencias a continuación no eximen de la lectura de los libros ya mencionados, es conveniente conocerlas para estar al tanto de las últimas noticias y novedades.

Portales de seguridad

Center for Education and Research in Information Assurance and Security, Purdue University
<http://www.cerias.purdue.edu/>

EnGarde System's Secure Zone
<http://www.securezone.com/>

Hacking and Hackers - Computer Security Programs Downloading Search Engines Portal News
<http://www.infosyssec.org/infosyssec/index.html>

National Institute of Standards and Technology Computer Security Resource Clearinghouse
<http://csrc.nist.gov>

Información sobre vulnerabilidades y seguridad

@stake, Inc.
<http://www.atstake.com/>

CERT Coordination Center, Carnegie Mellon University
<http://www.cert.org>

Computer Security News Daily
<http://www.mountainwave.com>

CriptoRed
<http://www.lpsi.eui.upm.es/criptored/criptored.htm>

esCERT
<http://escert.upc.es/>

Internet Security Systems's Xforce vulnerability database
<http://xforce.iss.net>

IrisCERT
<http://www.rediris.es/cert/>

NT-Bugtraq
<http://www.ntbugtraq.com>

Security Focus (Bugtraq, IDS, y otros foros de discusión)
<http://www.securityfocus.com/>

Documentos sobre Detección de Intrusiones

- Ranum, Marcus J. ICSA Labs IDSC. *False Positives: a User's Guide to Making Sense of IDS Alarms*. Febrero, 2003.
<http://www.icsalabs.com/html/communities/ids/whitepaper/FalsePositives.pdf>
- Bace, Rebecca, Peter Mell. ICSA Labs. *An Introduction to Intrusion Detection And Assessment*.
http://www.infidel.net/Articles/ICSA_whitepaper.pdf
- Bace, Rebecca. *NIST Special Publication on Intrusion Detection Systems*. 1999.
<http://csrc.nist.gov/publications/nistpubs/800-31/sp800-31.pdf>
- Lee, Wenke, Sal. Stolfo, and Kui Mok. *A Data Mining Framework for Building Intrusion Detection Models*. Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, CA, May 1999.
http://www.cc.gatech.edu/~wenke/papers/ieee_sp99_lee.ps
- Kruegel, Christopher, Thomas Toth and Engin Kirda. *Service Specific Anomaly Detection for Network Intrusion Detection*. Symposium on Applied Computing (SAC), ACM Digital Library, Spain, March 2002.
http://www.infosys.tuwien.ac.at/Staff/chris/doc/2002_03.ps
- Ptacek, Thomas H. and T. Newsham. *Insertions, Evasion, and Denial of Service: Eluding Network Intrusion Detection*. Enero 1998 .
<http://www.securityfocus.com/data/library/ids.ps>

Productos, desarrollo

Agnitum - Outpost Firewall
<http://www.agnitum.com/products/outpost/>

Bindview Development
<http://www.bindview.com>

Bro
<http://www.icir.org/vern/bro-info.html>

Counterpane Systems
<http://www.counterpane.com>

EnGarde Systems
<http://www.engage.com>

Enterasys - Dragon Intrusion Detection System
<http://www.enterasys.com/products/ids/>

Enterasys Networks
<http://www.enterasys.com/home.html>

IDSwakeup
<http://www.hsc.fr/ressources/outils/idswakeup/index.html>

Internet Security Systems
<http://www.iss.net>

Intrusion Detection Cybersafe
<http://www.cybersafe.com>

Network Associates
<http://www.nai.com>

NIDSbench
<http://packetstorm.widexs.nl/UNIX/IDS/nidsbench/nidsbench.html>

Portcullis Computer Security Ltd. - Dragon Intrusion Detection System
<http://www.portcullis-security.com/Products/>

Prelude
<http://www.prelude-ids.org>

RSA Security Inc.
<http://www.rsasecurity.com/>

Snort
<http://www.snort.org/>

SRI International
<http://www.csl.sri.com/programs/security/>

Tripwire Security Systems
<http://www.tripwiresecurity.com>

Referencias variadas sobre Detección de Intrusiones

Dan Farmer's security pages
<http://www.fish.com>

IDS FAQ
<http://www.robertgraham.com/pubs/network-intrusion-detection.html>

Michael Sobiley's Intrusion Detection Systems page
<http://www.rnks.informatik.tu-cottbus.de/~sobiley/ids.html>

SANS InfoSec Reading Room - Intrusion Detection
http://www.sans.org/rr/catindex.php?cat_id=30

SANS Institute Intrusion Detection FAQ
<http://www.sans.org/resources/idfaq/>

TruSecure Corporation
<http://www.trusecure.com/>

Intrusion Detection

<http://cnscenter.future.co.kr/security/ids.html>

SecurityFocus IDS

<http://www.securityfocus.com/ids>

Página personal de Wenke Lee

<http://www.cc.gatech.edu/~wenke/>

Organizaciones

Advanced Computing Systems Association

<http://www.usenix.org>

Association for Computing Machinery

<http://www.acm.org>

Computer Security Institute

<http://www.gocsi.com/>

Information Systems Audit and Control Association (ISACA)

<http://www.isaca.org>

Institute of Electrical and Electronic Engineers (IEEE)

<http://www.ieee.org>

International Information Systems Security Association (ISSA)

<http://www.issa-intl.org/>

International Information Systems Security Certification Consortium (ISCC)

<https://www.isc2.org/>

Internet Engineering Task Force

<http://www.ietf.org>

Internet Society

<http://www.isoc.org/>

Intrusion Detection Working Group of IETF

<http://www.ietf.org/html.charters/idwg-charter.html>

System Administration, Networking, and Security Institute

<http://www.sans.org/>

Grupos de discusión, listas de correo

SecurityFocus - IDS Mailing List (Focus-ids)

<http://www.securityfocus.com/archive/96>

SecurityFocus - Forensics Mailing List (Forensics)

<http://www.securityfocus.com/archive/104>

SecurityFocus - Forensics in Spanish Mailing List (Forensics-es)

<http://www.securityfocus.com/archive/128>

SecurityFocus - Honeypots Mailing List (Honeypots)

<http://www.securityfocus.com/archive/119>

Normativa legal, organismos oficiales

Guardia Civil - Grupo de Delitos Telemáticos

<http://www.guardiacivil.org/00telematicos/>

C.N.P. - Brigada de Investigación tecnológica

<http://www.mir.es/policia/bit/>

Delitos Informáticos -- Información legal Nuevas Tecnologías

<http://www.delitosinformaticos.com/>

Universidades

Iowa State University

<http://www.issl.org>

Purdue University

<http://www.cs.purdue.edu>

University of California, Davis

<http://seclab.cs.ucdavis.edu>

University of California, Santa Barbara

<http://cs.ucsb.edu>

University of Idaho

<http://www.cs.uidaho.edu>

University of Wollongong

<http://www.uow.edu.au/>

Apéndice F - Índice

A

AAFID (Agentes Autónomos para la Detección de Intrusiones), 72, 73, 74, 195
accountability, 43, 157, 187
ACL (Lista de Control de Acceso), 157, 174
ADAM, 61, 62
agente, 31, 70, 72, 73, 74, 76, 103, 119, 120, 123, 144, 149, 151, 153, 157, 163, 165, 167, 177, 242
 de usuario, 31, 165
Agentes Autónomos para la Detección de Intrusiones. Véase AAFID
aislamiento, 101, 161
ALAD, 61, 63, 64
amenaza, 14, 45, 133, 164, 167
análisis
 basado en intervalo, 161, 167
 con acreditaciones, 88, 159, 167
 de vulnerabilidades, 88, 89, 91, 92, 165, 167, 172, 175
 dinámico, 159, 167
 en modo por lotes, 158, 167
 en tiempo real, 20, 22, 44, 163, 167
 estático, 87, 164, 167
analizador de vulnerabilidades, 88, 89, 108, 165, 172
ancho de banda, 31, 97, 104, 144, 158, 167
Anderson, James P., 7, 11, 181
anomalía, 8, 9, 19, 20, 43, 44, 45, 46, 47, 59, 60, 61, 62, 63, 64, 66, 67, 68, 69, 71, 74, 75, 76, 103, 114, 115, 143, 144, 146, 155, 157, 163, 167, 171, 172
anomaly, 44, 62, 63, 83, 84, 85, 157, 163, 164, 186, 188, 189, 192, 193, 195, 218
API (Interfaz de programación de aplicaciones), 54, 57, 157, 174
aplicación engañosa, 103, 108, 109, 159, 167
application based, 19, 157
Application Program Interface. Véase API
archivo, 27, 121, 157, 159, 173, 200, 212, 240
ARPAnet, 3, 35
arrays, 9, 46
ASL, 57
assesment system, 87, 157
atributo, 60, 63, 64, 65, 87, 88, 157, 169, 174, 175
audit, 6, 7, 8, 11, 13, 26, 28, 72, 81, 82, 83, 84, 85, 147, 157, 159, 162, 164, 182, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 216, 220

 classes, 26
 flags, 26
 reduction, 7, 28, 157
 trail, 13, 157, 162
auditoría, 5, 6, 7, 9, 13, 18, 23, 24, 25, 26, 27, 28, 29, 30, 46, 47, 53, 54, 58, 60, 66, 67, 71, 72, 113, 115, 138, 139, 157, 159, 162, 164, 168, 172, 174, 177
 clases de, 26, 157
 informes de, 25
 rastros de, 13, 28
 reducción de, 28, 30, 60, 113, 157, 159, 176
autenticación, 8, 16, 31, 89, 91, 116, 157, 160, 164, 168, 173, 176
autenticación, 43, 157, 168
auto contenido, 163, 168
auto curación, 163
auto-curación, 77
Autonomous Agents for Intrusion Detection. Véase AAFID
autorización, 77, 132, 157, 168, 197, 202

B

backup, 73, 158
basado en
 agentes, 70, 72, 74
 aplicación, 19, 32, 103, 108, 109, 157, 168
 máquina, 9, 18, 23, 32, 34, 87, 88, 89, 91, 102, 103, 120, 124, 160, 161, 168
 modelos, 53
 multi-máquina, 18, 161, 168
 objetivo, 19, 32, 60, 164, 168
 red, 9, 19, 33, 34, 46, 50, 52, 61, 67, 74, 75, 88, 89, 90, 91, 103, 114, 116, 119, 120, 122, 123, 124, 125, 145, 149, 161, 168, 171
 redes neuronales, 67
 reglas, 8, 51, 66, 71, 115, 120, 163, 168, 178
 testigos, 16
base de reglas, 66, 67, 163, 168
Basic Security Module. Véase BSM, Véase BSM
batch mode, 20, 44, 158
Berkeley Packet Filter. Véase BPF
binary log format, 32, 158
Biología. Véase sistema inmune
Bloque de mensajes de servidor. Véase SMB
bridge, 158
Bro, 50, 52, 53, 83, 120, 126, 190, 191, 218

BSM (*Basic Security Module*), 25, 26, 81, 158, 175
 BSM (Módulo de Seguridad Básico), 25, 26, 81, 158, 175
 búfer, 158, 169, 170, 175
buffer, 17, 51, 158

C

C2 (nivel de TSEC), 6, 24, 25, 26, 175
 cable de sólo recepción, 33, 34, 120, 163, 169
 Capa de Conexión Segura. Véase SSL
 célula de aislamiento, 101, 162, 170
Cerberus Internet Scanner. Véase CIS
 CGI (Interfaz común de acceso), 51, 158, 173

Ch

checksum, 32, 60, 158

C

Ciberdelincuencia, Convenio sobre la, 130, 131
 CIDF (Marco de Detección de Intrusiones Común), 72, 85, 138, 147, 150, 154, 158, 174, 188
 cifrado, 16, 17, 19, 79, 94, 102, 113, 122, 123, 124, 141, 144, 145, 146, 151, 159, 168, 169, 171, 172, 176
 CIS (*Cerberus Internet Scanner*), 89, 109
 clave, 16, 17, 28, 43, 141, 145, 162
 CLIPS, 57
clusterig analysis, 65, 158
 código abierto, 98, 120, 160, 162, 170, 178
 Código Penal, 131, 132, 133, 134, 135, 136, 198, 199
Colored Petri Net, 54, 55, 158, Véase CP-net
Colored Petri Net (CP-net), 55, 158, 175, 184
Common Gateway Interface. Véase CGI
Common Intrusion Detection Framework. Véase CDIF
Common Log Format. Véase CFL
 composición, 32, 152, 158, 169
 comprobador de integridad, 160, 169
 comprometido, 55, 79, 95, 122, 138, 158, 169, 198
 concentrador, 34, 123, 160, 163, 169, 171, 177
 condición de carrera, 17, 70, 163, 169
 confianza, 3, 6, 13, 14, 17, 18, 43, 59, 88, 119, 164, 165, 169, 170, 174, 175, 177, 179
 confidencialidad, 6, 14, 130, 131, 158, 169, 177, 198

conmutador, 19, 34, 95, 103, 105, 106, 107, 108, 116, 122, 123, 125, 144, 158, 164, 169, 171, 176
 híbrido, 143
 conmutador híbrido, 103, 107, 108
 contraseña, 16, 44, 60, 88, 91, 115, 162, 163, 167, 171, 177
 contraseña oculta, 163
 control de acceso, 16, 17, 24, 39, 95, 157, 159, 161, 170
 Control de acceso discrecional. Véase DAC
 Control de accesos obligatorio. Véase MAC
 Convenio sobre la Ciberdelincuencia, 130, 131
 copia de seguridad, 26, 158
 correlación, 32, 89, 113, 152, 153, 155, 158, 170
 cortafuegos, 17, 39, 40, 93, 95, 96, 97, 100, 102, 103, 105, 106, 107, 108, 115, 116, 120, 121, 122, 149, 153, 160, 170, 178
cracking, 134
credentialled, 88, 159
 CRISIS (*Critical Resource Allocation and Intrusion Response for Survivable Information Systems*), 138, 147
 Criterio de Evaluación de Sistemas Informáticos Fiables. Véase TCSEC
Critical Resource Allocation and Intrusion Response for Survivable Information Systems. Véase CRISIS

D

DAC (Control de acceso discrecional), 16, 159, 170
 DARPA (*Defense Advanced Research Projects Agency*), 138, 141
data mining, 71, 72, 74, 85, 152, 159, 182, 188, 189, 218
 datagrama, 37, 38, 159, 170
 DDoS (Denegación de servicio distribuida), 105, 141, 159, 170
deceptive application, 108, 109, 159
decoy, 78, 159
Defense Advanced Research Projects Agency. Véase DARPA
Definable Log Format. Véase DLF
 delitos informáticos, 127, 130, 131, 132, 133, 135, 136, 142, 198
De-Militarized Zone. Véase DMZ
 Denegación de servicio. Véase DoS
 Denegación de servicio distribuida. Véase DDoS
Denial of Service. Véase DoS
 Denning, Dorothy, 7, 11, 46, 59, 67, 82, 83, 110, 139, 147, 183, 187

Derecho

- civil, 128
- penal, 130
- procesal, 130

desbordamiento de búfer, 106, 115, 158, 164, 170, 173

detección

- de anomalías, 19, 45, 46, 47, 48, 49, 51, 58, 59, 60, 61, 62, 63, 65, 66, 67, 68, 69, 72, 75, 80, 105, 107, 114, 115, 120, 143, 152, 153, 157, 171
- de intrusiones, 3, 5, 7, 8, 9, 10, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 29, 30, 32, 33, 38, 39, 41, 42, 43, 44, 47, 49, 51, 56, 57, 61, 64, 66, 67, 69, 70, 71, 72, 73, 74, 75, 76, 78, 79, 80, 81, 87, 89, 91, 92, 95, 100, 101, 102, 113, 114, 120, 130, 137, 138, 139, 140, 141, 142, 144, 149, 150, 151, 152, 153, 155, 161, 169, 170, 171, 172, 173, 174, 175, 178, 181
- de máquina, 124
- de red, 119
- difusa, 74
- de usos indebidos, 19, 45, 46, 47, 48, 49, 50, 53, 54, 57, 59, 60, 62, 64, 65, 69, 105, 113, 115, 152, 161, 171

Detección de intrusiones de nodo de red.

Véase NNID

Detección de intrusiones de red. Véase NID detector

- de máquina, 146
- de nodo de red, 123
- de red, 146

Detector de Intrusiones de máquina. Véase HID

determinista, 159, 171

DIDS (*Distributed Intrusion Detection System*), 9, 10, 11, 50, 193

dirección, 31, 33, 36, 37, 40, 77, 78, 97, 104, 116, 120, 125, 136, 145, 157, 172, 176, 242

***Discretionary Access Control*. Véase DAC**

disparador, 60, 165

disponibilidad, 14, 130, 131, 158, 171, 177, 198

dispositivo de escucha de red, 34, 38, 120, 162, 171

***Distributed Denial of Service*. Véase DDoS**

***Distributed Intrusion Detection System*.**

Véase DIDS

DLF (*Definable Log Format*), 31

DMZ (Zona desmilitarizada), 121, 159, 179

DNS (Sistema de Nombres de Dominio), 31, 37, 105, 121, 153, 159, 178, 179

***Domain Name System*. Véase DNS**

Dorothy Denning, 7, 11, 46, 59, 67, 82, 83, 110, 139, 147, 183, 187

DoS (Denegación de servicio), 20, 105, 106, 159, 170

Dragon, 152, 154, 219

E

EDP (*Electronic Data Process*), 5, 6, 159

***Electronic Data Process*. Véase EDP**

ELF (*Extended Log Format*), 31

encaminador, 163, 171

enmascarado, 161, 171

enmascaramiento, 16, 40, 70, 161, 164, 172

enrutador, 163, 171

entorno conmutado, 144, 164, 171

entrenamiento, 63, 64, 67, 68, 75, 107, 142, 164

error de Tipo I, 165, 171

error de Tipo II, 165, 171

escalabilidad, 140, 141, 151, 163, 171

escaneo de puertos, 74, 94, 162, 164, 172, 178

escaneo sigiloso de puertos, 51, 162, 164, 172, 178

escáner de vulnerabilidades, 68, 87, 91, 141, 143

esteganografía, 121, 164, 172

***Ethernet*, 38, 40, 63, 172, 178**

Eugene H. Spafford, 11, 29, 33, 82, 83, 110, 181, 183, 184, 187, 193, 195, 215, 217

***event horizon*, 49, 159**

***expert system*, 46, 159**

***exploit*, 46, 89, 159, 164**

***Extended Log Format*. Véase ELF**

F

falseamiento, 161, 164, 172

falsificar, 125

falso negativo, 68, 79, 93, 114, 143, 150, 159, 165, 171, 172

falso positivo, 40, 49, 67, 68, 71, 79, 93, 114, 115, 140, 143, 150, 159, 165, 171, 172

***File Transfer Protocol*. Véase FTP**

filtro

basado en concordancia de patrones, 28

de anomalías de protocolo, 61, 64, 163, 172

de anomalías estadísticas, 61, 164, 172

de paquetes, 17, 38, 158, 172

Filtro de paquetes Berkeley. Véase BPF

***fingerprinting*, 94**

***FIRE (Fuzzy Intrusion Recognition Engine)*, 74**

firma, 19, 20, 46, 55, 57, 62, 64, 81, 134, 136, 163, 172, 197, 209

formato de registro binario, 158, 172

FTP (Protocolo de Transferencia de Ficheros), 29, 53, 63, 64, 94, 105, 108, 121, 159, 160, 176, 179
función resumen, 20, 102, 124, 160, 161, 168, 169, 172
fuzzy data mining, 74, 160
Fuzzy Intrusion Recognition Engine. Véase FIRE2
fuzzy logic, 74, 160

G

gateway, 35, 97, 103, 158, 160
generación de patrones probables, 66, 67
generación de Patrones Probables, 162, 172
gestión
 de bases de datos, 30
 de memoria, 106
 de red, 22, 78, 79, 125, 162, 172, 173
 de rendimiento, 162, 172, 173
 de seguridad, 41, 42, 43, 75, 163, 173
gestor de registro de actividades, 161, 173
GID (Identificador de grupo), 33, 160
GLIMPSE, 58
Graph-Dased Intrusion Detection System. Véase GrIDS
GrIDS (Graph-Dased Intrusion Detection System), 141, 182, 193
guard, 56, 110, 160
guardias, 160
gusano, 9, 165, 173

H

hacking, 134, 135, 217
hash, 20, 32, 60, 160
hash function, 20
Haystack, 8, 10, 57, 61, 62, 83, 139, 147, 181, 192
híbrido, 19, 70, 120, 125, 152
HID (Detección de intrusiones de máquina), 160
historial, 27, 48, 161, 163
Honeynet, 92, 95, 96, 97, 98, 99, 100, 102, 110, 160, 215
 GenI, 95, 96, 97
 GenII, 95, 96, 97
 The Honeynet Project, 95, 110, 215
 virtual
 auto-contenida, 98, 99
 híbrida, 99, 100
honeypot, 78, 92, 93, 95, 99, 109, 110, 117, 159, 160, 215, 220
horizonte de evento, 49, 159, 173
host based, 18, 32, 160
Host based Intrusion Detection. Véase HID

HTTP (Protocolo de Transferencia de Hipertexto), 31, 32, 46, 63, 82, 105, 121, 153, 160, 176
hub, 34
huella dactilar, 94, 159
huella digital, 159
hybrid, 19, 126, 154
Hypertext Transfer Protocol. Véase HTTP

I

I&A (Identificación y autenticación), 16, 160, 173
IDES (Sistema Experto de Detección de Intrusiones), 7, 8, 11, 46, 47, 49, 50, 61, 62, 65, 82, 83, 139, 160, 161, 162, 178, 183, 186, 189, 193
IDIoT (Intrusion Detection In Our Time), 55, 56, 57, 183
IDS (Sistema de detección de intrusiones), 15, 21, 68, 97, 99, 103, 106, 107, 108, 110, 114, 119, 120, 123, 125, 126, 132, 139, 149, 151, 154, 160, 161, 177, 178, 191, 218, 219, 220
IDWG, 139, 150
IETF (Internet Engineering Task Force), 138, 139, 150, 160, 173, 177, 220
IIDS (Sistema de Detección de Intrusiones Inteligente), 74
IMDL (Intrusion Detection Markup Language), 57
inalámbrico, 165
inducción, 71
information retrieval, 58, 83, 160, 181
in-line, mode, 103, 104, 105, 160
inode, 33
inodo, 33
integración, 22, 78, 90, 140, 160, 173
integridad, 14, 20, 32, 60, 87, 102, 113, 124, 130, 131, 160, 173, 177, 179, 198
inteligencia artificial, 32, 142, 152, 157, 173, 178
Interconexión de Sistemas Abiertos. Véase OSI
interfaces, 34, 90, 101, 102, 103, 137, 138, 143, 145, 170, 174, 176
interfaz, 8, 38, 39, 40, 51, 73, 98, 104, 114, 120, 146, 157, 158, 160, 162, 163, 173, 174, 175, 176
Interfaz común de acceso. Véase CGI
interfaz de comandos, 8, 162, 163, 173
 polimórfica, 162, 173
Interfaz de comandos segura. Véase SSH
Interfaz de programación de aplicaciones. Véase API

Internet, 3, 8, 9, 11, 35, 36, 37, 52, 58, 82, 83, 89, 94, 96, 102, 105, 109, 110, 116, 121, 126, 132, 134, 138, 139, 147, 151, 154, 160, 161, 164, 165, 173, 176, 177, 179, 181, 184, 186, 187, 190, 191, 193, 194, 195, 215, 217, 218, 219, 220
Internet Engineering Task Force. Véase IETF
Internet Protocol Security. Véase IPsec
Internet Protocol version 6. Véase IPv6
interoperabilidad, 161, 174
intrusión, 10, 45, 48, 50, 55, 58, 59, 60, 62, 67, 71, 78, 80, 94, 114, 115, 134, 135, 140, 142, 143, 150, 161, 167, 168, 172, 173, 174, 177, 198
Intrusion Detection Expert System. Véase IDES
Intrusion Detection In Our Time. Véase IDIOT
Intrusion Detection Markup Language. Véase IMDL
Intrusion Detection System. Véase IDS
Intrusion Prevention System. Véase IPS
invadir, 159, 170
IPS (Sistema de prevención de intrusiones), 103, 104, 107, 108, 109, 149, 161, 178
IPSec (Seguridad de Procotolo Internet), 94, 116, 144, 146, 160, 177
IPv6 (Protocolo Internet versión 6), 94, 116, 151, 161

J

James P. Anderson, 7, 11, 181

K

kernel-level audit events, 26

L

Lee, Wenke, 72, 75, 85, 188, 218, 220
LERAD, 61, 63, 64
libpcap, 38, 51, 52, 172, 174
Libro Marrón, 6, 138, 163, 164, 174, 178
Libro Naranja, 6, 23, 24, 138, 162, 163, 170, 178
Linux, 29, 33, 38, 51, 95, 97, 98, 102, 110, 216
log, 5, 9, 25, 28, 29, 30, 31, 82, 126, 147, 157, 159, 161, 163, 164, 182, 186, 190
lógica difusa, 73, 74, 160, 174
LOPD, 134
lote, 158, 174
LSSICE, 134, 197

M

MAC (Control de accesos obligatorio), 16, 40, 161, 170
Mandatory Access Control. Véase MAC
Marco de Detección de Intrusiones Común. Véase CIDE
Markov, Proceso, 60, 67
MIDAS (*Multics Intrusion Detection and Alerting System*), 8, 50
minería de datos, 71, 72, 114, 152, 159, 160, 174
difusa, 160
misuse, 7, 19, 44, 81, 83, 161, 183, 186, 187, 188, 190, 191, 192
modo
de escucha, 38, 104, 164
en línea, 104, 160, 174
promiscuo, 9, 19, 34, 162, 175, 176
módulo, 25, 90, 97, 158, 162, 175, 177
Módulo de Seguridad Básico. Véase BSM
monitor, 7, 9, 11, 19, 23, 32, 33, 34, 38, 53, 73, 126, 144, 161, 168, 174, 175, 185, 186
monitor de referencias, 7
monitorizar, 3, 8, 9, 10, 13, 18, 34, 38, 43, 92, 95, 113, 120, 141, 142, 145, 151, 161, 167, 171, 175
Multics Intrusion Detection and Alerting System. Véase MIDAS
multi-host, 18, 73
based, 18, 161

N

NADIR (*Network Audit Director and Intrusion Reporter*), 8, 60, 83, 186
National Center for Supercomputing Alliances. Véase NCSA
NCSA (*National Center for Supercomputing Alliances*), 30, 32, 161
Nessus, 68, 90, 109, 141
NETAD, 61, 63, 64
Network Audit Director and Intrusion Reporter. Véase NADIR
network based, 19, 161
Network Flight Recorder. Véase NFR
Network Intrusion Detection. Véase NID
Network Node Intrusion Detection. Véase NNID
Network System Monitor. Véase NSM
network tap, 34, 38, 104, 120, 144
Network Time Protocol. Véase NTP
Neumann, Peter, 7, 8, 191
Next-Generation Intrusion Detection Expert System. Véase NIDES

NFR (*Network Flight Recorder*), 57, 83, 190
 NID (Detección de intrusiones de red), 162
 NIDES (Sistema Experto de detección de intrusiones de siguiente generación), 11, 50, 61, 62, 161, 162, 178, 193
 NNID (Detección de intrusiones de nodo de red), 19, 162
 no paramétrico, 44, 65, 66, 162, 175
 NSM (*Network System Monitor*), 9, 11, 185
 NTP (Protocolo de Tiempo de Red), 40, 162, 176

O

Open Systems Interconnection. Véase OSI
OS fingerprinting, 51, 162
 OSI (Interconexión de Sistemas Abiertos), 35, 36, 106, 162, 173, 176

P

padded cell, 101, 162
 paquete, 34, 37, 40, 47, 64, 96, 162, 172, 175
 parche, 162, 175
 pasarela, 160
password cracking, 91
 patrón, 45, 56, 57, 94, 163, 172
 P-BEST, 57
 Peter Neumann, 7, 8, 191
 Petri Net Coloreada. Véase CP-net
 PHAD, 61, 63, 64, 84, 189
 pila, 35, 158, 163, 164, 170, 175, 176
 pila de protocolos, 35, 163, 175, 176
plugin, 62, 162
 política de monitorización, 161, 175
predictive pattern generation, 60, 66, 162
 Prelude, 120, 125, 126, 152, 154, 219
 privacidad, 3, 134, 162, 175, 176, 197
 privilegio, 6, 17, 24, 54, 89, 162, 174, 175
 procesamiento por lotes, 158, 175
promiscuous mode, 9, 162
 Protocolo de Control de Transmisión / Protocolo Internet. Véase TCP/IP
 Protocolo de Tiempo de Red. Véase NTP
 Protocolo de Transferencia de Ficheros. Véase FTP
 Protocolo de Transferencia de Hipertexto. Véase HTTP
 Protocolo Internet versión 6. Véase IPv6
 Protocolo Simple de Transferencia de Correo. Véase SMTP
 puente, 97, 104, 158, 164, 169, 174, 176
 puerta de enlace, 97, 160
 puerta trasera, 102, 158, 165, 169, 176

R

race condition, 17, 163
 rastreador, 19, 34, 35, 38, 39, 44, 52, 123, 149, 163, 171, 174, 176
record, 25, 26, 163
 Red Privada Virtual. Véase VPN
 red trampa, 95, 97, 160, 176
 redes neuronales. Véase basado en redes neuronales
 reducción de datos, 61, 66, 69, 73, 159
 registro
 de aplicación, 157, 176
 de auditoría, 23, 24, 25, 28, 29, 30, 32, 47, 53, 54, 57, 61, 66, 125, 138, 139, 140, 144, 157, 176, 177
 de eventos, 27, 28, 44, 48, 144, 146, 159, 168, 176, 177
 de seguridad, 163, 177
 de sistema, 23, 28, 29, 30, 44, 48, 58, 60, 80, 125, 146, 164, 177
 repetidor, 160, 163, 169, 177
 respuesta activa, 21, 76, 77, 157, 163, 176, 177
 respuesta pasiva, 78, 80, 162, 177
 resumen de mensaje, 32, 161
 rompedor de contraseñas, 162, 177
 router, 35, 95, 96, 153, 182, 215
rule-based, 66
 RUSSEL, 57

S

salto de red, 161, 177
 SATAN, 90, 109, 141, 184
script, 44, 163
Secure Shell. Véase SSH
Secure Socket Layer. Véase SSL
 segmento, 33, 34, 35, 37, 38, 104, 115, 121, 163, 169, 171, 175, 176, 177
 seguridad, 3, 5, 6, 7, 8, 9, 10, 13, 14, 15, 16, 17, 18, 22, 23, 24, 25, 26, 27, 28, 29, 30, 35, 39, 41, 42, 43, 44, 45, 46, 49, 54, 55, 58, 64, 68, 69, 75, 76, 77, 79, 80, 81, 87, 88, 90, 91, 92, 93, 94, 95, 98, 99, 100, 101, 102, 103, 106, 113, 114, 115, 116, 119, 121, 122, 123, 124, 134, 135, 139, 140, 142, 143, 145, 146, 149, 153, 155, 158, 160, 163, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 197, 215, 217
 política de, 5, 6, 13, 14, 15, 32, 46, 52, 73, 80, 114, 163, 174, 175, 178, 179
 Seguridad de Protocolo Internet. Véase IPSec
self-healing, 77, 163
 sensor, 99, 141, 146, 151, 157, 163, 167, 177

serie Arco Iris, 163, 174, 178
Server Message Block. Véase SMB
 Servidor de Gestión de Sistemas. Véase SMS
 servidor señuelo, 159
session creep, 59, 67, 68, 70, 163
shell, 8, 162, 163, 164, 173
 Short Message Service. Véase SMS
Simple Mail Transfer Protocol. Véase SMTP
 sistema
 experto, 8, 46, 50, 159, 178
 híbrido, 8
 inmune, 69, 70, 114, 152
 Sistema de Detección de Intrusiones. Véase IDS
 Sistema de Detección de Intrusiones Inteligente. Véase IIDS
 Sistema de Nombres de Dominio. Véase DNS
 Sistema de prevención de intrusiones. Véase IPS
 Sistema Experto de Detección de Intrusiones. Véase IDES
 Sistema Experto de Detección de Intrusiones de Siguiente Generación. Véase NIDES
 Sistema inmune, 69, 70, 114, 152
 sistema trampa, 92, 93, 94, 95, 101, 102, 160, 176, 178
 sistemas de confianza, 6, 165, 178
 Smaha, Steve, 8, 10, 11, 57, 83, 139, 147, 177, 181, 182, 192, 193
 SMB (Bloque de mensajes de servidor), 51, 163, 168
 SMS (Servidor de Gestión de Sistemas), 38, 163, 164
 SMS (Short Message Service), 114, 125
 SMTP (Protocolo Simple de Transferencia de Correo), 108, 121, 163, 179
sniffer, 19, 34, 38, 52, 163
sniffing cable, 33, 163
 Snort, 15, 50, 51, 52, 62, 68, 83, 102, 104, 110, 120, 126, 191, 215, 219
 software libre, 160, 162, 170, 178, 240, 246
 Solaris, 25, 27, 29, 33, 39, 51, 95
 SPADE, 61, 62, 83, 186
 Spafford, Eugene H., 11, 29, 33, 82, 83, 110, 181, 183, 184, 187, 193, 195, 215, 217
 Spafford, Gene, 11, 29, 33, 82, 83, 110, 181, 183, 184, 187, 193, 195, 215, 217
spanning port, 144
 SPICE, 62, 83, 186
spoofing, 77, 164, 185
 SSH (Interfaz de comandos segura), 94, 116, 144, 163, 164, 173

SSL (Capa de Conexión Segura), 94, 116, 144, 146, 163, 164, 169
 STALKER, 57, 58, 139
 STAT, 54, 55, 56, 83, 191
 STAT (*State Transition Analysis Tool*), 54, 55, 56, 83, 191
State Transition Analysis Tool. Véase STAT
state transitions, 54
stealth port scans, 51
 Steve Smaha, 8, 10, 11, 57, 83, 139, 147, 177, 181, 182, 192, 193
stream, 39, 164
 STREAMS, 39
subnetting, 36
 subred, 36, 121, 122, 164, 179
 suma de comprobación, 32, 158, 179
 suma de control, 60, 158, 179
 suma de verificación, 158, 179
 Sun, 9, 25, 27, 28, 39, 57, 81, 113, 175
 SVR4++, 177
switch, 101, 102, 105, 110, 164
switched enviroments, 116
 syslogd, 28, 29
Systems Management Server. Véase SMS

T

tap mode, 104, 164
target based, 19, 32, 164
 TCP/IP (Protocolo de Control de Transmisión / Protocolo Internet), 35, 36, 82, 116, 164, 165, 176
tcpdump, 38, 52, 172, 179
 TCSEC (Criterio de Evaluación de Sistemas Informáticos Fiables), 23, 25, 26, 162, 164, 165, 170, 175
 testigo, 16, 25, 164, 168, 202, 211, 212
thread, 66, 164
threshold, 60, 164
 TIM (*Time-Based Inductive Machine*), 66, 67
Time-Based Inductive Machine. Véase TIM
 transiciones de estados, 50, 54, 60
 transmisor-receptor, 73, 164, 167
Transmission Control Protocol / Internet Protocol. Véase TCP/IP
trigger, 60
 Tripwire, 33, 60, 82, 102, 110, 125, 126, 187, 219
 troyano, 16, 43, 74, 102, 165, 169
Trusted Computer System Evaluation Criteria. Véase TCSEC

U

umbral, 59, 60, 164
UNIX, 8, 9, 18, 28, 29, 31, 32, 33, 38, 39, 43,
 51, 54, 58, 69, 82, 83, 89, 139, 147, 173,
 177, 181, 185, 186, 187, 195, 215, 219
user-agent, 31, 165
uso indebido, 7, 19, 20, 44, 45, 46, 47, 48,
 49, 50, 57, 69, 71, 72, 75, 139, 144, 146,
 150, 152, 155, 161, 172, 179

V

verificador de integridad, 32, 102
Virtual Private Network. Véase VPN
virus polimórfico, 162, 179
VPN (Red Privada Virtual), 17, 165, 176

vulnerabilidad, 13, 14, 43, 44, 45, 78, 87,
 88, 89, 90, 91, 113, 114, 119, 158, 164, 165,
 176, 179, 217
vulnerability scanner, 87, 165

W

Wenke Lee, 72, 75, 220
Windows, 25, 26, 27, 28, 30, 31, 33, 38, 43,
 51, 89, 95, 98, 168, 176, 177, 178, 215, 216
Wisdom and Sense, 9, 66, 67
wrapper, 70, 165

Z

Zona desmilitarizada. Véase DMZ

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's

overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

Licencia de Documentación Libre GNU (traducción)

Versión 1.1, Marzo de 2000

This is an unofficial translation of the GNU Free Documentation License into spanish. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU FDL--only the original English text of the GNU FDL does that. However, we hope that this translation will help spanish speakers understand the GNU FDL better.

Esta es una traducción NO oficial de la "GNU Free Documentation License" al español. No fue publicada por la "FSF Free Software Foundation", y no respalda legalmente los términos de distribución del software que utiliza la "GNU FDL", sólo el texto original en inglés lo hace. Sin embargo esperamos que esta traducción ayude a las personas de habla hispana a entender mejor la "GNU FDL".

Los autores de esta traducción son:

Igor Támara <ikks@bigfoot.com>

Pablo Reyes <reyes_pablo@hotmail.com>

Revisión : Vladimir Támara P. <vtamara@gnu.org>

Copyright © 2000

Free Software Foundation, Inc. 59 Temple Place, Suite 330,
Boston, MA 02111-1307 USA

Se permite la copia y distribución de copias literales de este documento de licencia, pero no se permiten cambios.

0. Preámbulo

El propósito de esta licencia es permitir que un manual, libro de texto, u otro documento escrito sea "libre" en el sentido de libertad: asegurar a todo el mundo la libertad efectiva de copiarlo y redistribuirlo, con o sin modificaciones, de manera comercial o no. En segundo término, esta licencia preserva para el autor o para quien publica una manera de obtener reconocimiento por su trabajo, al tiempo que no se consideran responsables de las modificaciones realizadas por terceros.

Esta licencia es una especie de "copyleft" que significa que los trabajos derivados del documento deben a su vez ser libres en el mismo sentido. Esto complementa la Licencia Pública General GNU, que es una licencia de copyleft diseñada para el software libre.

Hemos diseñado esta Licencia para usarla en manuales de software libre, ya que el software libre necesita documentación libre: Un programa libre debe venir con los manuales que ofrezcan la mismas libertades que da el software. Pero esta licencia no se limita a manuales de software; puede ser usada para cualquier trabajo textual, sin tener en cuenta su temática o si se publica como libro impreso. Recomendamos esta licencia principalmente para trabajos cuyo fin sea instructivo o de referencia.

1. Aplicabilidad y definiciones

Esta Licencia se aplica a cualquier manual u otro documento que contenga una nota del propietario de los derechos que indique que puede ser distribuido bajo los términos de la Licencia. El "Documento", en adelante, se refiere a cualquiera de dichos manuales o trabajos. Cualquier miembro del público es un licenciatario, y será denominado como "Usted".

Una "Versión Modificada" del Documento significa cualquier trabajo que contenga el Documento o una porción del mismo, ya sea una copia literal o con modificaciones y/o traducciones a otro idioma.

Una "Sección Secundaria" es un apéndice titulado o una sección preliminar al prólogo del Documento que tiene que ver exclusivamente con la relación de quien publica o, los autores del Documento o, el tema general del Documento(o asuntos relacionados) y cuyo contenido no entra directamente en este tema general. (Por ejemplo, si el Documento es en parte un texto de matemáticas, una Sección Secundaria puede no explicar matemáticas.) La relación puede ser un asunto de conexión histórica, o de posición legal, comercial, filosófica, ética o política con el tema o la materia del texto.

Las "Secciones Invariantes" son ciertas Secciones Secundarias cuyos títulos son denominados como Secciones Invariantes, en la nota que indica que el documento es liberado bajo esta licencia.

Los "Textos de Cubierta" son ciertos pasajes cortos de texto que se listan, como Textos de Portada o Textos de Contra Portada, en la nota que indica que el documento es liberado bajo esta Licencia.

Una copia "Transparente" del Documento, significa una copia para lectura en máquina, representada en un formato cuya especificación está disponible al público general, cuyos contenidos pueden ser vistos y editados directamente con editores de texto genéricos o (para imágenes compuestas por píxeles) de programas genéricos de dibujo o (para dibujos) algún editor gráfico ampliamente disponible, y que sea adecuado para exportar a formateadores de texto o para traducción automática a una variedad de formatos adecuados para ingresar a formateadores de texto. Una copia hecha en un formato de un archivo que no sea Transparente, cuyo formato ha sido

diseñado para impedir o dificultar subsecuentes modificaciones posteriores por parte de los lectores no es Transparente. Una copia que no es "Transparente" es llamada "Opaca".

Como ejemplos de formatos adecuados para copias Transparentes están el ASCII plano sin formato, formato de Texinfo, formato de LaTeX, SGML o XML usando un DTD disponible ampliamente, y HTML simple que sigue los estándares, diseñado para modificaciones humanas. Los formatos Opacos incluyen PostScript, PDF, formatos propietarios que pueden ser leídos y editados unicamente en procesadores de palabras propietarios, SGML o XML para los cuáles los DTD y/o herramientas de procesamiento no están disponibles generalmente, y el HTML generado por máquinas producto de algún procesador de palabras solo para propósitos de salida.

La "Portada" en un libro impreso significa, la portada misma, más las páginas siguientes necesarias para mantener la legibilidad del material, que esta Licencia requiere que aparezca en la portada. Para trabajos en formatos que no tienen Portada como tal, "Portada" significa el texto cerca a la aparición más prominente del título del trabajo, precediendo el comienzo del cuerpo del trabajo.

2. Copia literal

Puede copiar y distribuir el Documento en cualquier medio, sea en forma comercial o no, siempre y cuando esta Licencia, las notas de derecho de autor, y la nota de licencia que indica que esta Licencia se aplica al Documento se reproduzca en todas las copias, y que usted no adicione ninguna otra condición a las expuestas en esta Licencia. No puede usar medidas técnicas para obstruir o controlar la lectura o copia posterior de las copias que usted haga o distribuya. Sin embargo, usted puede aceptar compensación a cambio de las copias. Si distribuye un número suficientemente grande de copias también deberá seguir las condiciones de la sección 3.

También puede prestar copias, bajo las mismas condiciones establecidas anteriormente, y puede exhibir copias publicamente.

3. Copiado en cantidades

Si publica copias impresas del Documento que sobrepasen las 100, y la nota de Licencia del Documento exige Textos de Cubierta, debe incluir las copias con cubiertas que lleven en forma clara y legible, todos esos textos de Cubierta: Textos Frontales en la cubierta frontal, y Textos Posteriores de Cubierta en la Cubierta Posterior. Ambas cubiertas deben identificarlo a Usted clara y legiblemente como quien publica tales copias. La Cubierta Frontal debe mostrar el título completo con todas las palabras igualmente prominentes y visibles. Además puede adicionar otro material en la cubierta. Las copias con cambios limitados en las cubiertas, siempre que preserven el título del Documento y satisfagan estas condiciones, puede considerarse como copia literal.

Si los textos requeridos para la cubierta son muy voluminosos para que ajusten legiblemente, debe colocar los primeros (tantos como sea razonable colocar) en la cubierta real, y continuar el resto en páginas adyacentes.

Si publica o distribuye copias Opacas del Documento cuya cantidad exceda las 100, debe incluir una copia Transparente que pueda ser leída por una máquina con cada copia Opaca, o entregar en o con cada copia Opaca una dirección en red de computador publicamente-accesible conteniendo una copia completa Transparente del Documento, sin material adicional, a la cual el público en general de la red pueda acceder a bajar anónimamente sin cargo usando protocolos de standard público. Si usted hace uso de la última opción, deberá tomar medidas necesarias, cuando comience la distribución de las copias Opacas en cantidad, para asegurar que esta copia Transparente permanecerá accesible en el sitio por lo menos un año después de su última distribución de copias Opacas (directamente o a través de sus agentes o distribuidores) de esa edición al público.

Se solicita, aunque no es requisito, que contacte a los autores del Documento antes de redistribuir cualquier gran número de copias, para permitirle la oportunidad de que le provean una versión del Documento.

4. Modificaciones

Puede copiar y distribuir una Versión Modificada del Documento bajo las condiciones de las secciones 2 y 3 anteriores, siempre que usted libere la Versión Modificada bajo esta misma Licencia, con la Versión Modificada haciendo el rol del Documento, por lo tanto licenciando la distribución y modificación de la Versión Modificada a quienquiera que posea una copia de este. En adición, debe hacer lo siguiente en la Versión Modificada:

A. Uso en la Portada (y en las cubiertas, si hay alguna) de un título distinto al del Documento, y de versiones anteriores (que deberían, si hay alguna, estar listados en la sección de Historia del Documento). Puede usar el mismo título que versiones anteriores al original siempre que quien publicó la primera versión lo permita.

B. Listar en la Portada, como autores, una o más personas o entidades responsables por la autoría o las modificaciones en la Versión Modificada, junto con por lo menos cinco de los autores principales del Documento (Todos sus autores principales, si hay menos de cinco).

C. Estado en la Portada del nombre de quién publica la Versión Modificada, como quien publica.

D. Preservar todas las notas de derechos de autor del Documento.

E. Adicionar una nota de derecho de autor apropiada a sus modificaciones adyacentes a las otras notas de derecho de autor.

F. Incluir, inmediatamente después de la nota de derecho de autor, una nota de licencia dando el permiso público para usar la Versión Modificada bajo los términos de esta Licencia, de la forma mostrada en la Adición (LEGAL)abajo.

G. Preservar en esa nota de licencia el listado completo de Secciones Invariantes y en los Textos de las Cubiertas que sean requeridos como se especifique en la nota de Licencia del Documento

H. Incluir una copia sin modificación de esta Licencia.

I. Preservar la sección llamada "Historia", y su título, y adicionar a esta una sección estableciendo al menos el título, el año, los nuevos autores, y quién publicó la Versión Modificada como reza en la Portada. Si no hay una sección titulada "Historia" en el Documento, crear una estableciendo el

título, el año, los autores y quien publicó el Documento como reza en la Portada, añadiendo además un artículo describiendo la Versión Modificada como se estableció en el punto anterior.

J. Preservar la localización en red, si hay, dada en la Documentación para acceder públicamente a una copia Transparente del Documento, tanto como las otras direcciones de red dadas en el Documento para versiones anteriores en las cuáles estuviese basado. Estas pueden ubicarse en la sección "Historia". Se puede omitir la ubicación en red para un trabajo que sea publicado por lo menos 4 años antes que el mismo Documento, o si quien publica originalmente la versión da permiso explícitamente.

K. En cualquier sección titulada "Agradecimientos" o "Dedicatorias", preservar el título de la sección, y preservar en la sección toda la sustancia y el tono de los agradecimientos y/o dedicatorias de cada contribuyente que estén incluidas.

L. Preservar todas las Secciones Invariantes del Documento, sin alterar su texto ni sus títulos. Números de sección o el equivalente no son considerados parte de los títulos de la sección. M. Borrar cualquier sección titulada "Aprobaciones". Tales secciones no pueden estar incluidas en las Versiones Modificadas.

M. Borrar cualquier sección titulada "Aprobaciones". Tales secciones no pueden estar incluidas en las Versiones Modificadas.

N. No retitular ninguna sección existente como "Aprobaciones" o conflictuar con título con alguna Sección Invariante.

Si la Versión Modificada incluye secciones o apéndices nuevos o preliminares al prólogo que califican como Secciones Secundarias y contienen material no copiado del Documento, puede opcionalmente designar algunas o todas esas secciones como invariantes. Para hacerlo, adicione sus títulos a la lista de Secciones Invariantes en la nota de licencia de la Versión Modificada. Tales títulos deben ser distintos de cualquier otro título de sección.

Puede adicionar una sección titulada "Aprobaciones", siempre que contenga únicamente aprobaciones de su Versión Modificada por varias fuentes--por ejemplo, observaciones de peritos o que el texto ha sido aprobado por una organización como un standard.

Puede adicionar un pasaje de hasta cinco palabras como un Texto de Cubierta Frontal, y un pasaje de hasta 25 palabras como un texto de Cubierta Posterior, al final de la lista de Textos de Cubierta en la Versión Modificada. Solamente un pasaje de Texto de Cubierta Frontal y un Texto de Cubierta Posterior puede ser adicionado por (o a manera de arreglos hechos por) una entidad. Si el Documento ya incluye un texto de cubierta para la misma cubierta, previamente adicionado por usted o por arreglo hecho por la misma entidad, a nombre de la cual está actuando, no puede adicionar otra; pero puede reemplazar la anterior, con permiso explícito de quien publicó anteriormente tal cubierta.

El(los) autor(es) y quien(es) publica(n) el Documento no dan con esta Licencia permiso para usar sus nombres para publicidad o para asegurar o implicar aprobación de cualquier Versión Modificada.

5. Combinando documentos

Puede combinar el Documento con otros documentos liberados bajo esta Licencia, bajo los términos definidos en la sección 4 anterior para versiones modificadas, siempre que incluya en la combinación todas las Secciones Invariantes de todos los documentos originales, sin modificar, y listadas todas como Secciones Invariantes del trabajo combinado en su nota de licencia.

El trabajo combinado necesita contener solamente una copia de esta Licencia, y múltiples Secciones Invariantes Idénticas pueden ser reemplazadas por una sola copia. Si hay múltiples Secciones Invariantes con el mismo nombre pero con contenidos diferentes, haga el título de cada una de estas secciones único adicionándole al final de este, en paréntesis, el nombre del autor o de quien publicó originalmente esa sección, si es conocido, o si no, un número único. Haga el mismo ajuste a los títulos de sección en la lista de Secciones Invariantes en la nota de licencia del trabajo combinado.

En la combinación, debe combinar cualquier sección titulada "Historia" de los varios documentos originales, formando una sección titulada "Historia"; de la misma forma combine cualquier sección titulada "Agradecimientos", y cualquier sección titulada "Dedicatorias". Debe borrar todas las secciones tituladas "Aprobaciones."

6. Colecciones de documentos

Puede hacer una colección consistente del Documento y otros documentos liberados bajo esta Licencia, y reemplazar las copias individuales de esta Licencia en los varios documentos con una sola copia que esté incluida en la colección, siempre que siga las reglas de esta Licencia para una copia literal de cada uno de los documentos en cualquiera de todos los aspectos.

Puede extraer un solo documento de una de tales colecciones, y distribuirlo individualmente bajo esta Licencia, siempre que inserte una copia de esta Licencia en el documento extraído, y siga esta Licencia en todos los otros aspectos concernientes a la copia literal de tal documento.

7. Agregación con trabajos independientes

Una recopilación del Documento o de sus derivados con otros documentos o trabajos separados o independientes, en cualquier tipo de distribución o medio de almacenamiento, no como un todo, cuenta como una Versión Modificada del Documento, teniendo en cuenta que ninguna compilación de derechos de autor sea clamada por la recopilación. Tal recopilación es llamada un "agregado", y esta Licencia no aplica a los otros trabajos auto-contenidos y por lo tanto compilados con el Documento, o a cuenta de haber sido compilados, si no son ellos mismos trabajos derivados del Documento.

Si el requerimiento de la sección 3 del Texto de la Cubierta es aplicable a estas copias del Documento, entonces si el Documento es menor que un cuarto del agregado entero, Los Textos de la Cubierta del Documento pueden ser colocados en cubiertas que enmarquen solamente el

Documento entre el agregado. De otra forma deben aparecer en cubiertas enmarcando todo el agregado.

8. Traducción

La Traducción es considerada como una clase de modificación, Así que puede distribuir traducciones del Documento bajo los términos de la sección 4. Reemplazar las Secciones Invariantes con traducciones requiere permiso especial de los dueños de derecho de autor, pero puede incluir traducciones de algunas o todas las Secciones Invariantes adicionalmente a las versiones originales de las Secciones Invariantes. Puede incluir una traducción de esta Licencia siempre que incluya también la versión Inglesa de esta Licencia. En caso de un desacuerdo entre la traducción y la versión original en Inglés de esta Licencia, la versión original en Inglés prevalecerá.

9. Terminación

No se puede copiar, modificar, sublicenciar, o distribuir el Documento excepto por lo permitido expresamente bajo esta Licencia. Cualquier otro intento de copia, modificación, sublicenciamiento o distribución del Documento es nulo, y serán automáticamente terminados sus derechos bajo esa licencia. De todas maneras, los terceros que hayan recibido copias, o derechos, de su parte bajo esta Licencia no tendrán por terminadas sus licencias siempre que tales personas o entidades se encuentren en total conformidad con la licencia original.

10. Futuras revisiones de esta licencia

La Free Software Foundation puede publicar nuevas, revisadas versiones de la Licencia de Documentación Libre GNU de tiempo en tiempo. Tales nuevas versiones serán similares en espíritu a la presente versión, pero pueden diferir en detalles para solucionar problemas o intereses. Vea <http://www.gnu.org/copyleft/>.

Cada versión de la Licencia tiene un número de versión que la distingue. Si el Documento especifica que una versión numerada particularmente de esta licencia o "cualquier versión posterior" se aplica a esta, tiene la opción de seguir los términos y condiciones de la versión especificada o cualquiera posterior que ha sido publicada(no como un borrador)por la Free Software Foundation. Si el Documento no especifica un número de versión de esta Licencia, puede escoger cualquier versión que haya sido publicada(no como un borrador) por la Free Software Foundation.

Addendum

Para usar esta licencia en un documento que usted haya escrito, incluya una copia de la Licencia en el documento y ponga el siguiente derecho de autor y nota de licencia justo después del título de la página:

Derecho de Autor © Año Su Nombre.

Permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.1 o cualquier otra versión posterior publicada por la Free Software Foundation; con las Secciones Invariantes siendo LISTE SUS TÍTULOS, con los siendo LISTELO el texto de la Cubierta Frontal, y siendo LISTELO el texto de la Cubierta Posterior. Una copia de la licencia es incluida en la sección titulada "Licencia de Documentación Libre GNU".

Si no tiene Secciones Invariantes, escriba "Sin Secciones Invariantes" en vez de decir cuáles son invariantes. Si no tiene Texto de Cubierta Frontal, escriba "Sin Texto de Cubierta Frontal" en vez de "siendo LISTELO el texto de la Cubierta Frontal"; Así como para la Cubierta Posterior.

Si su documento contiene ejemplos de código de programa no triviales, recomendamos liberar estos ejemplos en paralelo bajo su elección de licencia de software libre, tal como la Licencia de Público General GNU, para permitir su uso en software libre.