



# **TRABAJO DE ADSCRIPCIÓN**

## **“SEGURIDAD DE SISTEMAS DISTRIBUIDOS”**

**DIRECTOR DE LA ADSCRIPCIÓN  
MAGISTER DAVID LUIS LA RED MARTÍNEZ**

**ALUMNA ADSCRIPTA:  
ESTELA EVANGELINA VOGELMANN MARTINEZ**

**SISTEMAS OPERATIVOS 2007**

## ÍNDICE

ÍNDICE.....	3
INTRODUCCIÓN .....	6
CONCEPTOS DE SEGURIDAD .....	8
1.1 - ¿Cuál puede ser el valor de los datos? .....	8
1.2 - Definiciones.....	9
1.3 - Seguridad Global.....	10
1.4 - Impacto en la organización .....	10
1.5 - Visibilidad del proceso.....	11
1.6 - Implementación .....	12
CONCEPTOS DE SISTEMAS DISTRIBUIDOS .....	13
2.1-Concepto de hardware .....	13
2.2-Conceptos de software.....	13
2.3-Ventajas de los sistemas distribuidos .....	13
2.4-Desventajas de los sistemas distribuidos .....	14
POLÍTICAS GENERALES DE SEGURIDAD .....	15
3.1 - ¿Qué son las políticas de seguridad informática (PSI)? .....	15
3.2 - Elementos de una política de seguridad informática .....	15
3.3 - Algunos parámetros para establecer políticas de seguridad .....	16
3.4 - Proposición de una forma de realizar el análisis para llevar a cabo un sistema de seguridad informática.....	17
3.5 - ¿Por qué las políticas de seguridad informática generalmente no consiguen implantarse?.....	18
3.6 - Las políticas de seguridad informática como base de la administración de la seguridad integral.....	19
3.7 - Riesgos .....	20
3.8- Niveles de trabajo.....	21
3.8.1 – Confidencialidad .....	21
3.8.2 – Integridad .....	22
3.8.3 - Autenticidad .....	22
3.8.4 - No – repudio .....	22
3.8.5 - Disponibilidad de los recursos y de la información.....	22
3.8.6 – Consistencia .....	22
3.8.7 - Control de acceso a los recursos .....	22
3.8.8 – Auditoría .....	23
3.9 - Algoritmo .....	23
¿CÓMO ESTABLECER LOS NIVELES DE RIESGO DE LOS RECURSOS INVOLUCRADOS? .....	27
4.1 - Ejemplo práctico.....	28
4.2 - Procedimiento de alta de cuenta de usuario .....	32
4.3 - Procedimiento de baja de cuenta de usuario .....	32
4.4 - Procedimiento para determinar las buenas passwords .....	32
4.5 - Procedimientos de verificación de accesos .....	33
4.6 - Procedimiento para el chequeo del tráfico de la red.....	33
4.7 - Procedimiento para el monitoreo de los volúmenes de correo .....	33
4.8 - Procedimientos para el monitoreo de conexiones activas.....	34
4.9 - Procedimiento de modificación de archivos .....	34
4.10 - Procedimientos para el resguardo de copias de seguridad.....	34
4.11 - Procedimientos para la verificación de las máquinas de los usuarios .....	34
4.12 - Procedimientos para el monitoreo de los puertos en la red .....	34

4.13 - Procedimientos de cómo dar a publicidad las nuevas normas de seguridad ....	35
4.14 - Procedimientos para recuperar información .....	35
4.15 - Check-Lists.....	35
<b>TIPOS DE ATAQUES Y VULNERABILIDADES .....</b>	<b>37</b>
5.1 - Negación de servicio (denial of service ).....	37
5.1.1 - ¿Qué es “Denial of service”? Descripción de ataques.....	37
5.1.2 - Modos de ataque .....	37
5.1.3 - Consumo de recursos escasos, limitados, o no renovables.....	38
5.1.4 - Destrucción o alteración de la información de configuración .....	40
5.1.5. - Destrucción o alteración física de los componentes de la red .....	40
5.1.6 - Prevención y respuesta .....	40
5.2 - Cracking de passwords.....	41
5.2.1 - El archivo “/etc/password”: descripción .....	41
5.2.2 - Descubrir una password .....	43
5.3 - E-mail bombing y spamming .....	44
5.3.1 - Descripción.....	44
5.3.2 - Detalles técnicos.....	45
5.3.3. - ¿Cómo proceder?.....	45
5.4 - Problemas de seguridad en el FTP .....	46
5.4.1 - El comando PORT .....	46
5.4.2. - El Protocolo FTP.....	46
5.4.3. - El ataque “Ftp bounce” .....	46
5.4.4 - Bypass de dispositivos de filtrado dinámicos.....	47
5.4.5 - Soluciones .....	48
5.4.6 - Otros problemas del FTP .....	49
5.5 - TFTP.....	50
5.6 - TELNET .....	50
5.7 - Los comandos “r”.....	51
5.8 - Seguridad en NetBIOS .....	52
5.8.1 - Qué hacer.....	52
<b>DESCRIPCIÓN DE ALGUNAS HERRAMIENTAS DE CONTROL Y</b>	
<b>SEGUIMIENTO DE ACCESOS .....</b>	<b>53</b>
6.1 - TCP-WRAPPERS .....	53
6.2. - NETLOG .....	56
6.2.1. - Tcpllogger.....	56
6.2.2. - UDPLOGGER.....	57
6.2.3. - ICMPLOGGER .....	57
6.2.4. - ETHERSCAN .....	58
6.2.5. - NSTAT .....	59
6.3. - ARGUS .....	59
6.4. - TCPDUMP .....	61
6.5. - SATAN (Security Administrator Tool for Analyzing Networks) .....	61
6.6. - ISS (Internet Security Scanner).....	62
6.7. - Courtney .....	62
6.8. - Gabriel .....	63
6.9. - TCPLIST .....	63
6.10. - NOCOL (Network Operations Center On-Line).....	63
<b>HERRAMIENTAS QUE CHEQUEAN LA INTEGRIDAD DEL SISTEMA.....</b>	<b>65</b>
7.1. - COPS (Computer Oracle and Password System).....	65
7.2. - Tiger .....	66

7.3. - Crack.....	67
7.4. - Tripwire .....	67
7.5.- SPAR .....	68
7.6.- lsof (List Open Files).....	68
7.7. - CPM (Check Promiscuous Mode) .....	69
7.8. - IFSTATUS .....	69
7.9. - OSH (Operator Shell).....	69
7.10. - NOSHELL.....	70
7.11. - TRINUX .....	70
BIBLIOGRAFÍA .....	72

## INTRODUCCIÓN

En la actualidad, las organizaciones son cada vez más dependientes de sus redes informáticas y un problema que las afecte, por mínimo que sea, puede llegar a comprometer la continuidad de las operaciones [9].

La falta de medidas de seguridad en las redes es un problema que está en crecimiento. Cada vez es mayor el número de atacantes y cada vez están más organizados, por lo que van adquiriendo día a día habilidades más especializadas que les permiten obtener mayores beneficios. Tampoco deben subestimarse las fallas de seguridad provenientes del interior mismo de la organización.

La propia complejidad de la red es una dificultad para la detección y corrección de los múltiples y variados problemas de seguridad que van apareciendo. En medio de esta variedad, han ido aumentando las acciones poco respetuosas de la privacidad y de la propiedad de recursos y sistemas. “Hackers”, “crakers”, entre otros, han hecho aparición en el vocabulario ordinario de los usuarios y de los administradores de las redes

Además de las técnicas y herramientas criptográficas, es importante recalcar que un componente muy importante para la protección de los sistemas consiste en la atención y vigilancia continua y sistemática por parte de los responsables de la red.

A la hora de plantearse en qué elementos del sistema se deben de ubicar los servicios de seguridad podrían distinguirse dos tendencias principales:

Protección de los sistemas de transferencia o transporte. En este caso, el administrador de un servicio asume la responsabilidad de garantizar la transferencia segura al usuario final de la información de forma lo más transparente posible.

Ejemplos de este tipo de planteamientos serían el establecimiento de un nivel de transporte seguro, de un servicio de mensajería con MTAs (Mail Transport Agents) seguras, o la instalación de un firewall, que defiende el acceso a una parte protegida de una red.

Aplicaciones seguras extremo a extremo. Si pensamos, por ejemplo, en el correo electrónico, consistiría en construir un mensaje en el cual el contenido ha sido asegurado mediante un procedimiento de encapsulado previo al envío. De esta forma, el mensaje puede atravesar sistemas heterogéneos y poco fiables sin por ello perder la validez de los servicios de seguridad provistos. Aunque el acto de asegurar el mensaje cae bajo la responsabilidad del usuario final, es razonable pensar que dicho usuario deberá usar una herramienta amigable proporcionada por el responsable de seguridad de su organización. Esta misma operatoria, puede usarse para abordar el problema de la seguridad en otras aplicaciones tales como videoconferencia, acceso a bases de datos, etc.

En ambos casos, un problema de capital importancia es la gestión de passwords. Este problema es inherente al uso de la criptografía y debe estar

resuelto antes de que el usuario esté en condiciones de enviar un solo bit seguro.

## CONCEPTOS DE SEGURIDAD

En la actualidad, la seguridad informática ha adquirido gran auge, dadas las cambiantes condiciones y las nuevas plataformas de computación disponibles. La posibilidad de interconectarse a través de redes, ha abierto nuevos horizontes que permiten explorar más allá de las fronteras de la organización. Esta situación ha llevado a la aparición de nuevas amenazas en los sistemas computarizados [9].

Consecuentemente, muchas organizaciones gubernamentales y no gubernamentales internacionales [1,2] han desarrollado documentos y directrices que orientan en el uso adecuado de estas destrezas tecnológicas y recomendaciones con el objeto de obtener el mayor provecho de estas ventajas, y evitar el uso indebido de la mismas. Esto puede ocasionar serios problemas en los bienes y servicios de las empresas en el mundo.

En este sentido, las políticas de seguridad informática (PSI) surgen como una herramienta organizacional para concientizar a cada uno de los miembros de una organización sobre la importancia y la sensibilidad de la información y servicios críticos que favorecen el desarrollo de la organización y su buen funcionamiento.

### 1.1 - ¿Cuál puede ser el valor de los datos?

Establecer el valor de los datos es algo totalmente relativo, pues la información constituye un recurso que, en muchos casos, no se valora adecuadamente debido a su intangibilidad, cosa que no ocurre con los equipos, la documentación o las aplicaciones. Además, las medidas de seguridad no influyen en la productividad del sistema por lo que las organizaciones son reticentes a dedicar recursos a esta tarea.

Cuando hablamos del valor de la información nos referimos, por ejemplo, a qué tan peligroso es enviar la información de mi tarjeta de crédito a través de Internet para hacer una compra, en una red gigantesca donde viajan no únicamente los 16 dígitos de mi tarjeta de crédito sino millones de datos más, gráficas, voz y vídeo.

De hecho, este tema es complejo. Algunos expertos opinan que se corre más peligro cuando se entrega una tarjeta de crédito al empleado de un restaurante o cuando se la emplea telefónicamente para efectivizar alguna compra.

El peligro más grande radica no en enviar la información sino una vez que esta información, unida a la de miles de clientes más, reposa en una base de datos de la compañía con las que se concretó el negocio. Con un único acceso no autorizado a esta base de datos, es posible que alguien obtenga no únicamente mis datos y los de mi tarjeta, sino que tendrá acceso a los datos y tarjetas de todos los clientes de esta compañía.

En efecto, el tema no está restringido únicamente a Internet. Aunque no se esté conectado a Internet, una red está expuesta a distintos tipos de ataques electrónicos, incluidos los virus.

Para tratar de asignar un valor al costo del delito electrónico podríamos

mencionar el reporte de la agencia norteamericana Defense Information Systems Agency titulado "Defending the Defense Information Infrastructure- Defense Information Systems Agency", del 9 de julio de 1996. En dicho informe las corporaciones más grandes de los Estados Unidos reportan haber experimentado pérdidas estimadas en U\$S 800 millones dólares en 1996 debido a ataques a la red. Asimismo el informe de marzo de 1997 de The Computer Security Institute (CSI) indica que el crimen de cómputo continúa en alza y se reportan pérdidas superiores a los U\$S 100 millones de dólares y esto es tan solo durante el primer cuarto del año 1997. Si, además, tenemos en cuenta que según las estadísticas de estas agencias norteamericanas sólo 1 de cada 500 ataques son detectados y reportados, ya es posible hacerse una idea de los valores involucrados en este tipo de delito.

Por esto, y por cualquier otro tipo de consideración que se tenga en mente, es realmente válido pensar que cualquier organización que trabaje con computadoras - y hoy en día más específicamente con redes de computadoras - debe tener normativas que hacen al buen uso de los recursos y de los contenidos, es decir, al buen uso de la información.

## 1.2 - Definiciones

Dado que se está tratando con conceptos que pueden tener múltiples interpretaciones, parece prudente acordar ciertos significados específicos. Por tanto, algunas definiciones, todas ellas extraídas del diccionario Espasa Calpe.

- *Seguridad*: es "calidad de seguro", y, seguro está definido como "libre de riesgo".
- *Información*: es "acción y efecto de informar".
- *Informar*: es "dar noticia de una cosa".
- *Redes*: es "el conjunto sistemático de caños o de hilos conductores o de vías de comunicación o de agencias y servicios o recursos para determinado fin".

Uniendo todas estas definiciones, podemos establecer qué se entiende por Seguridad en redes.

*Seguridad en Redes o Sistemas Distribuidos*: es mantener la provisión de información libre de riesgo y brindar servicios para un determinado fin.

Si trabajamos en definir Seguridad en Redes con los elementos que conocemos, podemos llegar a una definición más acertada:

*Seguridad en sistemas distribuidos es mantener bajo protección los recursos y la información con que se cuenta en la red, a través de procedimientos basados en una política de seguridad tales que permitan el control de lo actuado.*

### 1.3 - Seguridad Global

¿Qué es una red global?. El concepto de red global incluye todos los recursos informáticos de una organización, aún cuando estos no estén interconectados:

- Redes de área local (LAN),
- Redes de área metropolitana (MAN),
- Redes nacionales y supranacionales (WAN),
- Computadoras personales, minis y grandes sistemas.

De manera que, seguridad global es mantener bajo protección todos los componentes de una red global.

Al fin de cuentas, los usuarios de un sistema son una parte a la que no hay que olvidar ni menospreciar. Siempre hay que tener en cuenta que la seguridad comienza y termina con personas.

Obtener de los usuarios la concientización de los conceptos, usos y costumbres referentes a la seguridad, requiere tiempo y esfuerzo. Que los usuarios se concienticen de la necesidad y, más que nada, de las ganancias que se obtienen implementando planes de seguridad, exige trabajar directamente con ellos, de tal manera que se apoderen de los beneficios de tener un buen plan de seguridad. (Por ejemplo: permite que se determine exactamente lo que debe hacer cada uno y cómo debe hacerlo, y, también las desviaciones que se pueden producir). De esta forma, ante cualquier problema, es muy fácil determinar dónde se produjo o de dónde proviene.

Para realizar esto, lo más usado, y que da muy buenos resultados es hacer “grupos de trabajo” en los cuales se informen los fines, objetivos y ganancias de establecer medidas de seguridad, de tal manera que los destinatarios finales se sientan informados y tomen para sí los conceptos. Este tipo de acciones favorece, la adhesión a estas medidas.

### 1.4 - Impacto en la organización

La implementación de políticas de seguridad, trae aparejados varios tipos de problemas que afectan el funcionamiento de la organización. ¿Cómo pueden impactar si se implementan para hacer más seguro el sistema? En realidad, la implementación de un sistema de seguridad conlleva a incrementar la complejidad en la operatoria de la organización, tanto técnica como administrativa.

Por ejemplo, la disminución de la funcionalidad o el decremento de la operatividad tal vez sea uno de los mayores problemas. Esto se puede aclarar de la siguiente manera: en un primer momento, el usuario, para acceder a tal recurso, debía realizar un solo login. Ahora, con la implementación del nuevo esquema de seguridad, debe realizar dos logines: uno para ingresar al sistema y otro para acceder al recurso. El usuario visualiza esto como un nuevo impedimento en su tarea, en lugar de

verlo como una razón de seguridad para él, pues de esta manera, se puede controlar más el uso del recurso y, ante algún problema, será mucho más fácil establecer responsabilidades.

Por otro lado, al poner en funcionamiento una nueva norma de seguridad, ésta traerá una nueva tarea para la parte técnica (por ejemplo, cambiar los derechos a algo de algunos usuarios) y administrativamente, se les deberá avisar por medio de una nota de los cambios realizados y en qué les afectará.

### **1.5 - Visibilidad del proceso**

En un reciente estudio de Datapro Research Corp. se resumía que los problemas de seguridad en sistemas basados en redes responden a la siguiente distribución:

- Errores de los empleados 50%
- Empleados deshonestos 15%
- Empleados descuidados 15%
- Otros 20% (Intrusos ajenos a la Empresa 10%; Integridad física de instalaciones 10%)

Se puede notar que el 80% de los problemas, son generados por los empleados de la organización, y, éstos se podrían tipificar en tres grandes grupos:

- Problemas por ignorancia
- Problemas por haraganería
- Problemas por malicia

Entre estas razones, la ignorancia es la más fácil de direccionar. Desarrollando tácticas de entrenamiento y procedimientos formales e informales son fácilmente neutralizadas. Los usuarios, además, necesitan de tiempo en tiempo, que se les recuerden cosas que ellos deberían conocer.

La haraganería será siempre una tentación –tanto para los administradores de sistemas como para los usuarios – pero, se encuentra que éste es un problema menor cuando los usuarios ven las metas de los sistemas de seguridad. Esto requiere soporte de las Gerencias y de la Administración, y de la organización como un todo formado por usuarios individuales. En adición, una atmósfera que se focalice en las soluciones, en lugar de censurar, es generalmente más eficiente que aquella que tiende a la coerción o la intimidación.

La malicia, se debe combatir creando una cultura en la organización que aliente la lealtad de los empleados.

La visibilidad es permitir el aporte de las personas de la organización y, dar a conocer las acciones tomadas. Es decir que, cuando se deben producir cambios en las políticas no es necesario que se decidan unilateralmente. Es altamente deseable que se formen grupos de trabajo para discutir y/o conocer el alcance y el tipo de medidas a llevar a cabo (esta metodología de trabajo se puede implementar por lo menos en un

80 % de las veces en que se presenta esta cuestión). Esto, además de llevar algunas veces a obtener soluciones que son más efectivas que las que se pensaban tomar, hace que aquellos que sean tocados por las modificaciones no se sientan recelosos de los cambios realizados y se comprometan con el cambio. Luego, una vez tomada la decisión, se debe comunicar fehacientemente a los involucrados de los cambios realizados por medio de minutas, notas o boletines informativos.

De esta manera, aseguramos que los hechos son visibles al resto de la organización. Como consecuencia, las personas no sienten resquemores o celos de las nuevas medidas implementadas y adhieren rápidamente a ellas. Cabría, asimismo, tener en cuenta cuando deban realizarse modificaciones, hacerlas contando con la asesoría de la parte legal. Así, se pueden llegar a establecer los alcances de las penalidades en caso de infringir las normas dictadas. Respecto de este punto, es también aconsejable que las modificaciones o nuevas normativas, así como las penalizaciones, estén bien publicitadas, ya sea por medio de boletines, en tableros de novedades, en transparencias o por cualquier medio que permita llevar a cabo estas acciones con razonable éxito.

## **1.6 - Implementación**

La implementación de medidas de seguridad, es un proceso técnico-administrativo. Como este proceso debe abarcar toda la organización, sin exclusión alguna, ha de estar fuertemente apoyado por el sector gerencial, ya que sin ese apoyo, las medidas que se tomen no tendrán la fuerza necesaria.

Hay que tener muy en cuenta la complejidad que suma a la operatoria de la organización la implementación de estas medidas (ver párrafo "Impacto en la Organización"). Será necesario sopesar cuidadosamente la ganancia en seguridad respecto de los costos administrativos y técnicos que se generen.

También, como hemos mencionado anteriormente, es fundamental no dejar de lado la notificación a todos los involucrados en las nuevas disposiciones y, darlas a conocer al resto de la organización con el fin de otorgar visibilidad a los actos de la administración.

De todo lo expuesto anteriormente, resulta claro que proponer o identificar una política de seguridad requiere de un alto compromiso con la organización, agudeza técnica para establecer fallas y debilidades, y constancia para renovar y actualizar dicha política en función del dinámico ambiente que rodea las organizaciones modernas.

Un reciente estudio indica que para el 25% de los responsables de la información en empresas de más de 100 empleados, a la seguridad no se le da una prioridad alta dentro de la organización y, en los casos en que se da esta prioridad, creen que el resto de la organización no presta a este tema la atención que requiere.

## CONCEPTOS DE SISTEMAS DISTRIBUIDOS

En general se consideran sistemas distribuidos, en sentido amplio, a los sistemas en que existen varias cpu conectadas entre sí y las distintas cpu trabajan de manera conjunta [8].

### 2.1-Concepto de hardware

Todos los sistemas distribuidos constan de varias cpu, organizadas de diversas formas [7]. Por ello, existen diversos esquemas de clasificación para los sistemas de cómputos con varias cpu. Uno de los más conocidos es la "Taxonomía de Flynn" que considera como características esenciales el número de flujo de instrucciones y el número de flujos de datos (SISD, SIMD, MISD y MIMD). Un avance sobre esta clasificación incluye la división de las computadoras MIMD (Multiple Instruction Multiple Data) en dos grupos: uno los multiprocesadores que poseen memoria compartida y otro las multicomputadoras que no poseen memoria compartida.

Cada una de las categorías indicadas se pueden clasificar según la arquitectura de la red de interconexión en: esquema de bus donde existe una sola red, bus, cable u otro medio que conecta todas las máquinas; y esquema con conmutador donde no existe una sola columna vertebral de conexión.

Otro aspecto de la clasificación considera el acoplamiento entre los equipos, los cuales son: sistemas fuertemente acoplados y Sistemas débilmente acoplados. A estos últimos, por lo general se los utiliza como sistemas distribuidos.

### 2.2-Conceptos de software

La importancia del software supera frecuentemente a la del hardware [7]. Es así que la imagen que un sistema presenta queda determinada en gran medida por el software del S. O. y no por el hardware.

Los S. O. no se pueden encasillar fácilmente, como el hardware, pero se los puede clasificar en dos tipos: débilmente acoplados y fuertemente acoplados. El software débilmente acoplado de un sistema distribuido permite que las máquinas y usuarios sean independientes entre sí, facilitando que interactúen en cierto grado cuando sea necesario.

Combinando los distintos tipos de hardware distribuido con software distribuido se logran distintas soluciones, las cuales no todas interesan desde el punto de vista funcional del usuario.

### 2.3-Ventajas de los sistemas distribuidos

Una ventaja potencial de un sistema distribuido es una mayor confiabilidad al distribuir la carga de trabajo en muchas máquinas, la falla de una de ellas no afectará a las demás. Es decir, que la carga de trabajo podría distribuirse. Entonces, si una máquina se descompone sobrevive el sistema como un todo.

Otra ventaja importante es la posibilidad del crecimiento incremental o por incrementos. Por tanto, podrían añadirse procesadores al sistema, permitiendo un desarrollo gradual según las necesidades. También satisfacen la necesidad de muchos

usuarios de compartir ciertos datos [7]. Se pueden compartir otros recursos como programas y periféricos costosos. Otra importante razón es lograr una mejor comunicación entre las personas. Como así también, la mayor flexibilidad es también importante, ya que la carga de trabajo se puede difundir (distribuir) entre las máquinas disponibles en la forma más eficaz según el criterio adoptado (por ej. costos), teniendo en cuenta que los equipos distribuidos pueden no ser siempre PC.

#### **2.4-Desventajas de los sistemas distribuidos**

El principal problema es el software, ya que el diseño, implantación y uso del software distribuido presenta numerosos inconvenientes [7].

Otro problema potencial tiene que ver con las redes de comunicaciones, ya que se deben considerar problemas debidos a pérdidas de mensajes, saturación en el tráfico, expansión, etc.

El hecho de que sea fácil compartir los datos es una ventaja pero se puede convertir en un gran problema, por lo que la seguridad debe organizarse adecuadamente.

En general se considera que las ventajas superan a las desventajas, si estas últimas se administran seriamente.

## **POLÍTICAS GENERALES DE SEGURIDAD**

### **3.1 - ¿Qué son las políticas de seguridad informática (PSI)?**

Una política de seguridad informática es una forma de comunicarse con los usuarios y los gerentes [3, pág.382]. Las PSI establecen el canal formal de actuación del personal, en relación con los recursos y servicios informáticos, importantes de la organización [9].

No se trata de una descripción técnica de mecanismos de seguridad, ni de una expresión legal que involucre sanciones a conductas de los empleados. Es más bien una descripción de los que deseamos proteger y el por qué de ello.

Cada PSI es consciente y vigilante del personal por el uso y limitaciones de los recursos y servicios informáticos críticos de la compañía.

### **3.2 - Elementos de una política de seguridad informática**

Una PSI debe orientar las decisiones que se toman en relación con la seguridad. Por tanto, requiere de una disposición por parte de cada uno de los miembros de la empresa para lograr una visión conjunta de lo que se considera importante.

Las PSI deben considerar entre otros, los siguientes elementos: [4]

- Alcance de las políticas, incluyendo facilidades, sistemas y personal sobre la cual se aplica. Es una invitación de la organización a cada uno de sus miembros a reconocer la información como uno de sus principales activos así como, un motor de intercambio y desarrollo en el ámbito de sus negocios. Invitación que debe concluir en una posición.
- Objetivos de la política y descripción clara de los elementos involucrados en su definición.
- Responsabilidades por cada uno de los servicios y recursos informáticos en todos los niveles de la organización.
- Requerimientos mínimos para configuración de la seguridad de los sistemas que cubija el alcance de la política.
- Definición de violaciones y de las consecuencias del no cumplimiento de la política.
- Responsabilidades de los usuarios con respecto a la información a la que ella tiene acceso.

Las PSI deben ofrecer explicaciones comprensibles acerca de por qué deben tomarse ciertas decisiones, transmitir por qué son importantes estos u otros recursos o servicios.

De igual forma, las PSI establecen las expectativas de la organización

en relación con la seguridad y lo que ella puede esperar de las acciones que la materializan en la compañía. Deben mantener un lenguaje común libre de tecnicismos y términos legales que impidan una comprensión clara de las mismas, sin sacrificar su precisión y formalidad dentro de la empresa.

Por otra parte, la política debe especificar la autoridad que debe hacer que las cosas ocurran, el rango de los correctivos y sus actuaciones que permitan dar indicaciones sobre la clase de sanciones que se puedan imponer. No debe especificar con exactitud qué pasará o cuándo algo sucederá; no es una sentencia obligatoria de la ley. [4, pág.383]

Finalmente, las PSI como documentos dinámicos de la organización, deben seguir un proceso de actualización periódica sujeto a los cambios organizacionales relevantes: crecimiento de la planta de personal, cambio en la infraestructura computacional, alta rotación de personal, desarrollo de nuevos servicios, cambio o diversificación de negocios entre otros.

### **3.3 - Algunos parámetros para establecer políticas de seguridad**

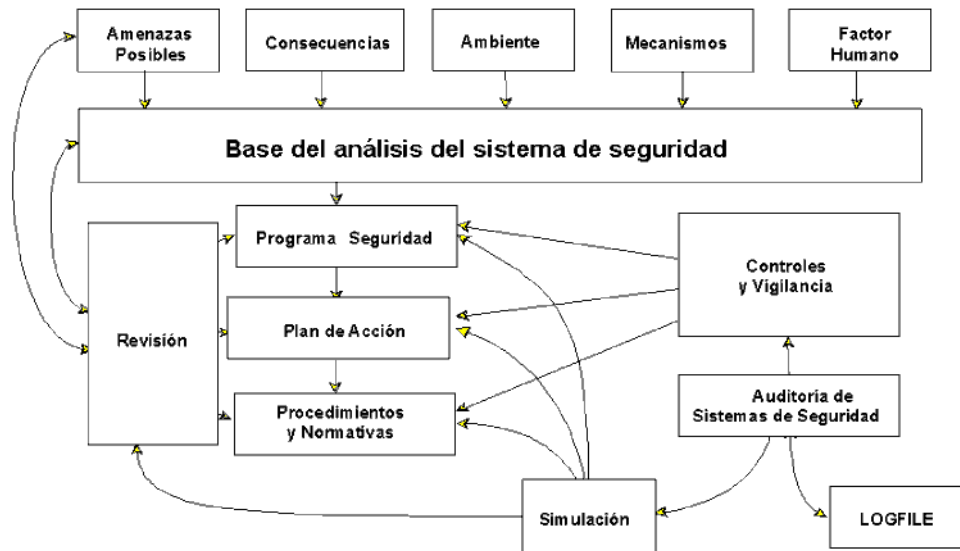
Algunos aspectos generales recomendados para la formulación de las mismas.

- Considerar efectuar un ejercicio de análisis de riesgos informático, a través del cual valore sus activos, el cual le permitirá afinar las PSI de su organización.
- Involucrar a las áreas propietarias de los recursos o servicios, pues ellos poseen la experiencia y son fuente principal para establecer el alcance y las definiciones de violaciones a la PSI.
- Comunicar a todo el personal involucrado en el desarrollo de las PSI, los beneficios y riesgos relacionados con los recursos y bienes, y sus elementos de seguridad.
- Recordar que es necesario identificar quién tiene la autoridad para tomar decisiones, pues son ellos los responsables de salvaguardar los activos críticos de la funcionalidad de su área u organización.
- Desarrollar un proceso de monitoreo periódico de las directrices en el hacer de la organización, que permita una actualización oportuna de las mismas.

En este ámbito también es bueno, no dar por hecho algo que es obvio. Hacer explícito y concreto los alcances y propuestas de seguridad, con el propósito de evitar sorpresas y malos entendidos en el momento de establecer los mecanismos de seguridad que respondan a las PSI trazadas.

### 3.4 - Proposición de una forma de realizar el análisis para llevar a cabo un sistema de seguridad informática

#### Diagrama para el análisis de un sistema de seguridad



Tal como puede visualizarse, en el gráfico están plasmados todos los elementos que intervienen para el estudio de una política de seguridad.

Se comienza realizando una evaluación del factor humano interviniente- teniendo en cuenta que éste es el punto más vulnerable en toda la cadena de seguridad -, de los mecanismos con que se cuentan para llevar a cabo los procesos necesarios ( mecanismos técnicos, físicos ó lógicos), luego, el medio ambiente en que se desempeña el sistema, las consecuencias que puede traer aparejado defectos en la seguridad (pérdidas físicas, pérdidas económicas, en la imagen de la organización, etc.), y cuáles son las amenazas posibles.

Una vez evaluado todo lo anterior, se origina un programa de seguridad, que involucra los pasos a tomar para poder asegurar el umbral de seguridad que se desea. Luego, se pasa al plan de acción, que es cómo se va a llevar a cabo el programa de seguridad. Finalmente, se redactan los procedimientos y normas que permiten llegar a buen destino.

Con el propósito de asegurar el cumplimiento de todo lo anterior, se realizan los controles y la vigilancia que aseguran el fiel cumplimiento de los tres puntos antepuestos. Para asegurar un marco efectivo, se realizan auditorías a los controles y a los archivos logísticos que se generen en los procesos implementados (de nada vale tener archivos logísticos si nunca se los analizan o se los analizan cuando ya ha ocurrido un problema).

Con el objeto de confirmar el buen funcionamiento de lo creado, se procede a simular eventos que atenten contra la seguridad del sistema. Como el proceso de seguridad es un proceso dinámico, es necesario realizar revisiones al programa de seguridad, al plan de acción y a los procedimientos y normas. Estas revisiones, tendrán efecto sobre los puntos tratados en el primer párrafo y, de esta manera, el proceso se vuelve a repetir.

El establecimiento de políticas de seguridad es un proceso dinámico sobre el que hay que estar actuando permanentemente, de manera tal que no quede desactualizado; que, cuando se le descubran debilidades, éstas sean subsanadas y, finalmente, que su práctica por los integrantes de la organización no caiga en desuso.

### **3.5 - ¿Por qué las políticas de seguridad informática generalmente no consiguen implantarse?**

Muchas veces, las organizaciones realizan grandes esfuerzos para definir sus directrices de seguridad y concretarlas en documentos que orienten las acciones de las mismas, con relativo éxito. Según algunos estudios [5] resulta una labor ardua convencer a los altos ejecutivos de la necesidad de buenas políticas y prácticas de seguridad informática.

Muchos de los inconvenientes se inician por los tecnicismos informáticos y por la falta de una estrategia de mercadeo de los especialistas en seguridad que, llevan a los altos directivos a pensamientos como: "más dinero para los juguetes de los ingenieros". Esta situación ha llevado a que muchas empresas con activos muy importantes, se encuentren expuestas a graves problemas de seguridad que, en muchos de los casos, lleva a comprometer su información sensible y por ende su imagen corporativa.

Ante esta encrucijada, los encargados de la seguridad deben asegurarse de que las personas relevantes entienden los asuntos importantes de la seguridad, conocen sus alcances y están de acuerdo con las decisiones tomadas en relación con esos asuntos. En particular, la gente debe conocer las consecuencias de sus decisiones, incluyendo lo mejor y lo peor que podría ocurrir. [3, pág.394] Una intrusión o una travesura puede convertir a las personas que no entendieron, en blanco de las políticas o en señuelos de los verdaderos vándalos. Luego, para que las PSI logren abrirse espacio en el interior de una organización deben integrarse a las estrategias del negocio, a su misión y visión, con el propósito de que los que toman las decisiones reconozcan su importancia e incidencias en las proyecciones y utilidades de la compañía.

De igual forma, las PSI deben ir acompañadas de una visión de negocio que promueva actividades que involucren a las personas en su hacer diario, donde se identifiquen las necesidades y acciones que materializan las políticas. En este contexto, entender la organización, sus elementos culturales y comportamientos nos debe llevar a reconocer las pautas de seguridad necesarias y suficientes que aseguren confiabilidad en las operaciones y funcionalidad de la compañía.

A continuación, se mencionan algunas recomendaciones para

concientizar sobre la seguridad informática:

- Desarrollar ejemplos organizacionales relacionados con fallas de seguridad que capten la atención de sus interlocutores.
- Asociar el punto anterior a las estrategias de la organización y a la imagen que se tiene de la organización en el desarrollo de sus actividades.
- Articular las estrategias de seguridad informática con el proceso de toma de decisiones y los principios de integridad, confidencialidad y disponibilidad de la información. Mostrar una valoración costo-beneficio, ante una falla de seguridad.
- Justificar la importancia de la seguridad informática en función de hechos y preguntas concretas, que muestren el impacto, limitaciones y beneficios sobre los activos claves de la organización.

### **3.6 - Las políticas de seguridad informática como base de la administración de la seguridad integral.**

Las políticas de seguridad informática conforman el conjunto de lineamientos que una organización debe seguir para asegurar la confiabilidad de sus sistemas. En razón de lo anterior, son parte del engranaje del sistema de seguridad que la organización posee para salvaguardar sus activos. Las PSI constituyen las alarmas y compromisos compartidos en la organización, que le permiten actuar proactivamente ante situaciones que comprometan su integridad. Por tanto, deben constituir un proceso continuo y retroalimentado que observe la concientización, los métodos de acceso a la información, el monitoreo de cumplimiento y la renovación, aceptación de las directrices y estrategia de implantación, que lleven a una formulación de directivas institucionales que logren aceptación general.

Las políticas por sí mismas no constituyen una garantía para la seguridad de la organización. Ellas deben responder a intereses y necesidades organizacionales basados en la visión de negocio, que lleven a un esfuerzo conjunto de sus actores por administrar sus recursos y a reconocer en los mecanismos de seguridad informática factores que facilitan la normalización y materialización de los compromisos adquiridos con la organización.

La seguridad tiene varios estratos:

- El marco jurídico adecuado.
- Medidas técnico-administrativas, como la existencia de políticas y procedimientos o la creación de funciones, como administración de la seguridad o auditoría de sistemas de información interna.

Debe existir una definición de funciones y una separación suficiente de tareas. No tiene sentido que una misma persona autorice una transacción, la introduzca, y revise después los resultados (un diario de operaciones, por ejemplo), porque podría planificar un fraude o encubrir

cualquier anomalía; por ello deben intervenir funciones / personas diferentes y existir controles suficientes. La seguridad física, como la ubicación de los centros de procesos, las protecciones físicas, el control físico de accesos, los vigilantes, las medidas contra el fuego y el agua, y otras similares.

La llamada seguridad lógica, como el control de accesos a la información exige la identificación y autenticación del usuario, o el cifrado de soportes magnéticos intercambiados entre entidades o de respaldo interno, o de información transmitida por línea. Puede haber cifrado de la información por dispositivos físicos o a través de programas, y en casos más críticos existen los dos niveles.

### 3.7 - Riesgos

La autenticación suele realizarse mediante una contraseña, aún cuando sería más lógico - si bien los costos resultan todavía altos para la mayoría de los sistemas - que se pudiera combinar con características biométricas del usuario para impedir la suplantación. Entre éstas pueden estar: la realización de la firma con reconocimiento automático por ordenador, el análisis del fondo de ojo, la huella digital u otras.

Al margen de la seguridad, nos parece que el mayor riesgo, aún teniendo un entorno muy seguro, es que la Informática y la Tecnología de la Información en general no cubran las necesidades de la entidad; o que no estén alineadas con las finalidades de la organización.

Limitándonos a la seguridad propiamente dicha, los riesgos pueden ser múltiples. El primer paso es conocerlos y el segundo es tomar decisiones al respecto; conocerlos y no tomar decisiones no tiene sentido y debiera crearnos una situación de desasosiego.

Dado que las medidas tienen un costo, a veces, los funcionarios se preguntan cuál es el riesgo máximo que podría soportar su organización. La respuesta no es fácil porque depende de la criticidad del sector y de la entidad misma, de su dependencia respecto de la información, y del impacto que su no disponibilidad pudiera tener en la entidad. Si nos basamos en el impacto nunca debería aceptarse un riesgo que pudiera llegar a poner en peligro la propia continuidad de la entidad, pero este listón es demasiado alto.

Por debajo de ello hay daños de menores consecuencias, siendo los errores y omisiones la causa más frecuente - normalmente de poco impacto pero con frecuencia muy alta - y otros, como por ejemplo:

- el acceso indebido a los datos (a veces a través de redes),
- la cesión no autorizada de soportes magnéticos con información crítica (algunos dicen "sensible"),
- los daños por fuego, por agua (del exterior como puede ser una inundación, o por una tubería interior),
- la variación no autorizada de programas, su copia indebida, y tantos otros, persiguiendo el propio beneficio o causar un daño, a veces por venganza.

Otra figura es la del “hacker”, que intenta acceder a los sistemas sobre todo para demostrar (a veces, para demostrarse a sí mismo/a) que es capaz de hacer, al superar las barreras de protección que se hayan establecido.

Alguien podría preguntarse por qué no se citan los virus, cuando han tenido tanta incidencia. Afortunadamente, este riesgo es menor en la actualidad comparando con años atrás. Existe, de todas maneras, un riesgo constante porque de forma continua aparecen nuevas modalidades, que no son detectadas por los programas antivirus hasta que las nuevas versiones los contemplan. Un riesgo adicional es que los virus pueden llegar a afectar a los grandes sistemas, sobre todo a través de las redes, pero esto es realmente difícil - no nos atrevemos a decir que imposible- por las características y la complejidad de los grandes equipos y debido a las características de diseño de sus sistemas operativos.

En definitiva, las amenazas hechas realidad pueden llegar a afectar los datos, en las personas, en los programas, en los equipos, en la red y algunas veces, simultáneamente en varios de ellos, como puede ser un incendio.

Podríamos hacernos una pregunta realmente difícil: ¿qué es lo más crítico que debería protegerse? La respuesta de la mayoría, probablemente, sería que las personas resultan el punto más crítico y el valor de una vida humana no se puede comparar con las computadoras, las aplicaciones o los datos de cualquier entidad. Ahora bien, por otra parte, podemos determinar que los datos son aún más críticos si nos centramos en la continuidad de la entidad.

Como consecuencia de cualquier incidencia, se pueden producir unas pérdidas que pueden ser no sólo directas (comúnmente que son cubiertas por los seguros) más fácilmente, sino también indirectas, como la no recuperación de deudas al perder los datos, o no poder tomar las decisiones adecuadas en el momento oportuno por carecer de información.

### **3.8- Niveles de trabajo**

- Confidencialidad
- Integridad
- Autenticidad
- No Repudio
- Disponibilidad de los recursos y de la información
- Consistencia
- Control de Acceso
- Auditoría

#### **3.8.1 – Confidencialidad**

Consiste en proteger la información contra la lectura no autorizada explícitamente. Incluye no sólo la protección de la información en su totalidad, sino también las piezas individuales que pueden ser utilizadas para inferir otros elementos de información confidencial.

### **3.8.2 – Integridad**

Es necesario proteger la información contra la modificación sin el permiso del dueño. La información a ser protegida incluye no sólo la que está almacenada directamente en los sistemas de cómputo sino que también se deben considerar elementos menos obvios como respaldos, documentación, registros de contabilidad del sistema, tránsito en una red, etc. Esto comprende cualquier tipo de modificaciones:

- Causadas por errores de hardware y/o software.
- Causadas de forma intencional.
- Causadas de forma accidental

Cuando se trabaja con una red, se debe comprobar que los datos no fueron modificados durante su transferencia.

### **3.8.3 - Autenticidad**

En cuanto a telecomunicaciones se refiere, la autenticidad garantiza que quien dice ser "X" es realmente "X". Es decir, se deben implementar mecanismos para verificar quién está enviando la información.

### **3.8.4 - No – repudio**

Ni el origen ni el destino en un mensaje deben poder negar la transmisión. Quien envía el mensaje puede probar que, en efecto, el mensaje fue enviado y viceversa.

### **3.8.5 - Disponibilidad de los recursos y de la información**

De nada sirve la información si se encuentra intacta en el sistema pero los usuarios no pueden acceder a ella. Por tanto, se deben proteger los servicios de cómputo de manera que no se degraden o dejen de estar disponibles a los usuarios de forma no autorizada. La disponibilidad también se entiende como la capacidad de un sistema para recuperarse rápidamente en caso de algún problema.

### **3.8.6 – Consistencia**

Se trata de asegurar que el sistema siempre se comporte de la forma esperada, de tal manera que los usuarios no encuentren variantes inesperadas.

### **3.8.7 - Control de acceso a los recursos**

Consiste en controlar quién utiliza el sistema o cualquiera de los recursos que ofrece y cómo lo hace.

### **3.8.8 – Auditoría**

Consiste en contar con los mecanismos para poder determinar qué es lo que sucede en el sistema, qué es lo que hace cada uno de los usuarios y los tiempos y fechas de dichas acciones.

En cuanto a los dos últimos puntos resulta de extrema importancia, cuando se trata de los derechos de los usuarios, diferenciar entre “espiar” y “monitorear” a los mismos. La ética es algo que todo buen administrador debe conocer y poseer.

Finalmente, todos estos servicios de seguridad deben ser tomados en cuenta en el momento de elaborar las políticas y procedimientos de una organización para evitar pasar por alto cuestiones importantes como las que señalan dichos servicios. De esta manera, es posible sentar de forma concreta y clara los derechos y límites de usuarios y administradores. Sin embargo antes de realizar cualquier acción para lograr garantizar estos servicios, es necesario asegurarnos de que los usuarios conozcan sus derechos y obligaciones (es decir, las políticas), de tal forma que no se sientan agredidos por los procedimientos organizacionales.

## **3.9 - Algoritmo**

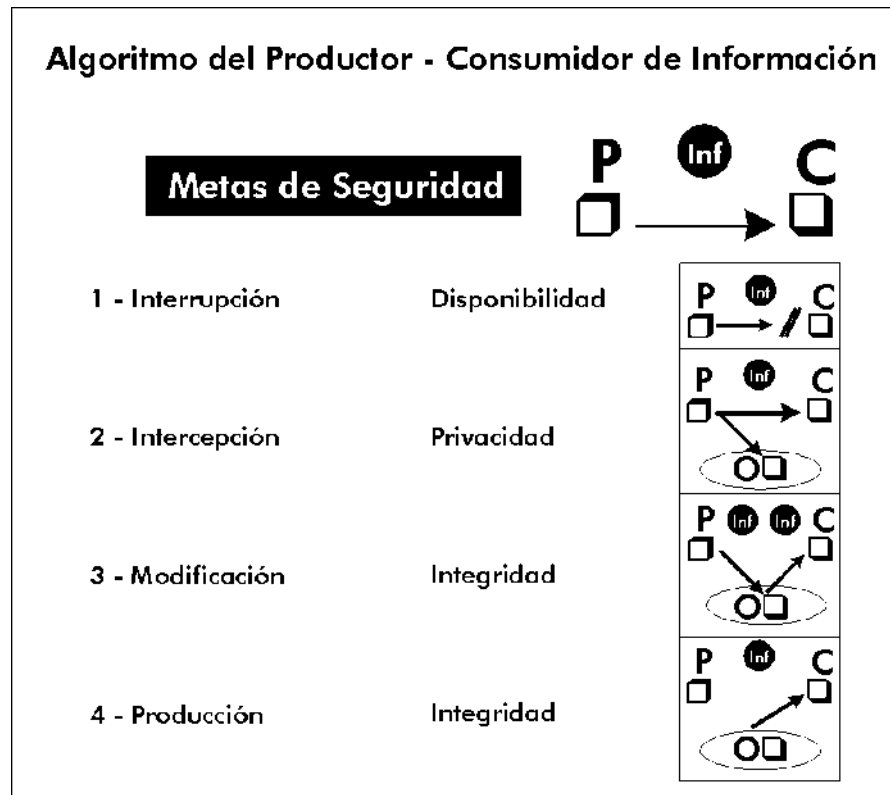
Cuando se piensa establecer una estrategia de seguridad, la pregunta que se realiza, en primera instancia, es: ¿en qué baso mi estrategia?. La respuesta a esta pregunta es bien simple. El algoritmo Productor/Consumidor.

En este algoritmo, hay dos grandes entidades: una que es la encargada de producir la información; la otra entidad es el consumidor de esta información y otra, llamada precisamente “otros”. Entre el productor y el consumidor, se define una relación que tiene como objetivo una transferencia de “algo” entre ambos, sin otra cosa que intervenga en el proceso. Si esto se logra llevar a cabo y se mantiene a lo largo del tiempo, se estará en presencia de un sistema seguro.

En la realidad, existen entidades y/o eventos que provocan alteraciones a este modelo. El estudio de la seguridad, en pocas palabras, se basa en la determinación, análisis y soluciones de las alteraciones a este modelo.

En una observación y planteo del modelo, determinamos que sólo existen cuatro tipos de alteraciones en la relación producción-consumidor (ver el gráfico del algoritmo)

Antes de pasar a explicar estos casos, habrá que definir el concepto de “recurso”.



Recurso, está definido en el diccionario Espasa Calpe como “bienes, medios de subsistencia”.

Esta es una definición muy general. De todas maneras, resulta conveniente para nuestra tarea. Podemos mencionar como recurso a cualquier cosa, ya sean bienes específicos o que permitan la subsistencia de la organización como tal.

Debido a ello, es que podemos diferenciar claramente tres tipos de recursos:

- Físicos
- Lógicos
- Servicios.

Los recursos físicos son, por ejemplo, las impresoras, los servidores de archivos, los routers, etc.

Los recursos lógicos son, por ejemplo, las bases de datos de las cuales sacamos la información que permite trabajar en la organización.

Los servicios son, por ejemplo, el servicio de correo electrónico, de página WEB, etc.

Todas las acciones correctivas que se lleven a cabo con el fin de respetar el modelo estarán orientadas a atacar uno de los cuatro casos. Explicaremos y daremos ejemplos de cada uno de ellos.

El caso número uno es el de *Interrupción*. Este caso afecta la disponibilidad del recurso (tener en cuenta la definición de recurso: físico, lógico y servicio).

Por ejemplo:

Recurso afectado	Nombre	Causa	Efecto
Servicio	Correo electrónico	Alguien dio de baja el servidor (por algún método)	No poder enviar mail
Físico	Impresora	Falta de alimentación	No imprime

El segundo caso es el de *Intercepción*, en el cual se pone en riesgo la privacidad de los datos.

Recurso afectado	Nombre	Causa	Efecto
Lógico	Datos sobre cuentas en el banco	Se ha puesto un dispositivo que permite monitorear los paquetes en la red	Conseguir datos privados sobre montos de cuentas corrientes
Servicio	Correo electrónico	Se ha implantado un programa que duplica los mensajes (mails) que salen de una sección y los envía a una dirección.	Leer información

El tercer caso, *Modificación* afecta directamente la integridad de los datos que le llegan al consumidor.

Recurso afectado	Nombre	Causa	Efecto
Lógico	Base de datos de pagos en cuentas corrientes	Se ha implantado un programa que redondea en menos los pagos y carga éstos redondeos a una cuenta corriente	Incrementar el crédito de una cuenta corriente en base al redondeo realizado en
Servicio	Servidor de página WEB	Alguien logró ingresar como WEBMASTER y ha cambiado los contenidos de la página	Los datos mostrados en la página no son los reales

El cuarto y último caso es el de la *producción* impropia de información. En éste, la información que recibe el consumidor es directamente falaz.

Recurso afectado	Nombre	Causa	Efecto
Lógico	Datos de deudores	Se ha generado una base de datos falsa, la que ante el pedido de informes, responde ella con sus datos	Hacer pasar a los deudores como que no lo son
Servicio	Servidor WEB	Alguien se ha apropiado del password del WEBMASTER y, modificando el direccionamiento, logra que se cargue otra página WEB	Redireccionar la página WEB hacia otro sitio

## ¿CÓMO ESTABLECER LOS NIVELES DE RIESGO DE LOS RECURSOS INVOLUCRADOS?

Al crear una política de seguridad de red, es importante entender que la razón para crear tal política es, en primer lugar, asegurar que los esfuerzos invertidos en la seguridad son costeables. Esto significa que se debe entender cuáles recursos de la red vale la pena proteger y que algunos recursos son más importantes que otros. También se deberá identificar la fuente de amenaza de la que se protege a los recursos. A pesar de la cantidad de publicidad sobre intrusos en una red, varias encuestas indican que para la mayoría de las organizaciones, la pérdida real que proviene de los “miembros internos” es mucho mayor (tal cual se ha explicado anteriormente)[9].

El análisis de riesgos implica determinar lo siguiente:

- Qué se necesita proteger
- De quién protegerlo
- Cómo protegerlo

Los riesgos se clasifican por el nivel de importancia y por la severidad de la pérdida. No se debe llegar a una situación donde se gasta más para proteger aquello que es menos valioso.

En el análisis de los riesgos, es necesario determinar los siguientes factores:

- Estimación del riesgo de pérdida del recurso (lo llamaremos  $R_i$ )
- Estimación de la importancia del recurso (lo llamaremos  $W_i$ )

Como un paso hacia la cuantificación del riesgo de perder un recurso, es posible asignar un valor numérico. Por ejemplo, al riesgo ( $R_i$ ) de perder un recurso, se le asigna un valor de cero a diez, donde cero, significa que no hay riesgo y diez es el riesgo más alto. De manera similar, a la importancia de un recurso ( $W_i$ ) también se le puede asignar un valor de cero a diez, donde cero significa que no tiene importancia y diez es la importancia más alta. La evaluación general del riesgo será entonces el producto del valor del riesgo y su importancia (también llamado el peso). Esto puede escribirse como:

$$WR_i = R_i * W_i$$

Donde:

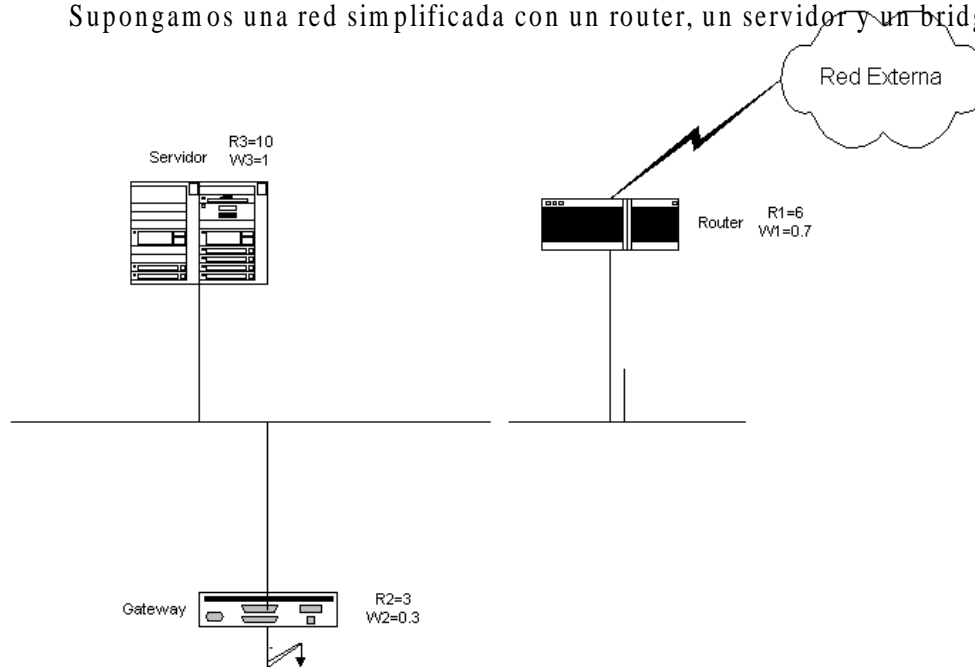
$WR_i$  : es el peso del riesgo del recurso "i" (también lo podemos llamar ponderación)

$R_i$  : es el riesgo del recurso "i"

$W_i$  : es la importancia del recurso "i"

#### 4.1 - Ejemplo práctico

Supongamos una red simplificada con un router, un servidor y un bridge.



Los administradores de la red y de sistemas han producido las estimaciones siguientes para el riesgo y la importancia de cada uno de los dispositivos que forman nuestra red:

Como se ve, a cada uno de los componentes del sistema, se le ha asignado un cierto riesgo y una cierta importancia. Hay que destacar que estos valores son totalmente subjetivos, dependen exclusivamente de quien ó quienes están realizando la evaluación.

Tenemos, entonces:

*Router:*

$$R_1 = 6$$

$$W_1 = 7$$

*Bridge:*

$$R_2 = 6$$

$$W_2 = 3$$

*Servidor:*

$$R_3 = 10$$

$$W_3 = 10$$

El cálculo de los riesgos evaluados, será, para cada dispositivo: Router:

$$WR1 = R1 * W1 = 6 * 7 = 42$$

Bridge:

$$WR2 = R2 * W2 = 6 * 3 = 1.8$$

Servidor:

$$WR3 = R3 * W3 = 10 * 10 = 100$$

La tabla que sigue a continuación, nos muestra cómo podríamos llevar a cabo esta tarea de una manera ordenada y los valores que contiene son los que hemos tratado:

Recurso del sistema		Riesgo (R <sub>i</sub> )	Importancia (W <sub>i</sub> )	Riesgo Evaluado (R <sub>i</sub> * W <sub>i</sub> )
Número	Nombre			
1	Router	6	7	42
2	Bridge	6	3	18
3	Servidor	1	10	100

Vemos que, en este caso, el recurso que debemos proteger más es el Servidor ya que su riesgo ponderado es muy alto. Por tanto, se debe comenzar por buscar las probables causas que pueden provocar problemas con los servicios brindados por él.

Hay que tener muy en cuenta que, al realizar el análisis de riesgo, se deben identificar todos los recursos (por más triviales que parezcan) cuya seguridad está en riesgo de ser quebrantada.

Ahora bien, ¿cuáles son los recursos?

Los recursos que deben ser considerados al estimar las amenazas a la seguridad son solamente seis:

*Hardware:* procesadores, tarjetas, teclados, terminales, estaciones de trabajo, computadoras personales, impresoras, unidades de disco, líneas de comunicación, cableado de la red, servidores de terminal, routers, bridges.

*Software:* programas fuente, programas objeto, utilerías, programas de diagnóstico, sistemas operativos, programas de comunicaciones.

*Datos:* durante la ejecución, almacenados en línea, archivados fuera de línea, back-up, bases de datos, en tránsito sobre medios de comunicación.

*Gente:* usuarios, personas para operar los sistemas.

*Documentación:* sobre programas, hardware, sistemas, procedimientos administrativos locales.

*Accesorios:* papel, formularios, cintas, información grabada.

Ahora, cómo protegemos nuestros recursos. Tal vez, ésta sea la

pregunta más difícil de responder, pues, según el recurso del que se trate, será el modo de protegerlo.

Primero, deberemos tener en cuenta qué es lo queremos proteger. Si se trata de los problemas ocasionados por el personal propio o de intromisiones clandestinas que puedan afectar la operatoria de la organización. (1)

Hay que tener en cuenta, que todos los estudios realizados demuestran que el 80% de los problemas proceden de los llamados “clientes internos” de la organización (los empleados o elementos que se desempeñan en la organización), y sólo el 20 % restante, proviene de elementos externos a la organización.

Una aproximación acerca de cómo proteger los recursos de los problemas originados por el cliente interno consiste en la identificación del uso correcto de los mismos por parte de éstos.

Pero primero, deberemos saber quiénes son los que van a hacer uso de los recursos. Una vez identificados los usuarios (o grupos de usuarios), se puede realizar la determinación de los recursos de que harán uso y de los permisos que tendrán. Esto es sencillo de realizar con una tabla como la siguiente:

Recurso del sistema		Identificación del usuario	Tipo de acceso	Permisos otorgados
Número	Nombre			
1	Base Datos Cuentas Corrientes	Grupo de auditores	Local	Lectura
2	Router 2500	Grupo de mantenimiento de comunicaciones	Local y remoto	Lectura y escritura

Este modelo, nos permitirá disponer para cada usuario (o grupos de usuarios), la información de qué se les está permitido hacer y qué no.

El otro problema que se presenta, es el de las intromisiones clandestinas.

Aquí, es preciso tener en cuenta el tipo de recurso a proteger. En base a ello, estará dada la política de seguridad.

Daremos, a continuación, algunos ejemplos acerca de a qué nos estamos enfrentando:

- ¿Cómo aseguramos que no están ingresando a nuestro sistema por un puerto desprotegido o mal configurado?

- ¿Cómo nos aseguramos de que no se estén usando programas propios del sistema operativo o aplicaciones para ingresar al sistema en forma clandestina?
- ¿Cómo aseguramos de que, ante un corte de energía eléctrica, el sistema seguirá funcionando?
- ¿Cómo nos aseguramos de que los medios de transmisión de información no son susceptibles de ser monitoreados?
- ¿Cómo actúa la organización frente al alejamiento de uno de sus integrantes?

La respuesta a estos interrogantes reside en la posibilidad de conseguir dicha seguridad por medio de herramientas de control y seguimiento de accesos, utilizando check-lists para comprobar puntos importantes en la configuración y/o funcionamiento de los sistemas y por medio de procedimientos que hacen frente a las distintas situaciones.

Es muy aconsejable que se disponga de una agenda con las tareas que se deben llevar a cabo regularmente, a fin de que el seguimiento de los datos obtenidos sea efectivo y se puedan realizar comparaciones válidas al contar con datos secuenciales. Esta agenda, podría ser en sí misma un procedimiento.

Damos, a continuación, un ejemplo de procedimiento de chequeo de eventos en el sistema:

*Diariamente:*

- Extraer un logístico sobre el volumen de correo transportado.
- Extraer un logístico sobre las conexiones de red levantadas en las últimas 24 horas.

*Semanalmente:*

- Extraer un logístico sobre los ingresos desde el exterior a la red
- Extraer un logístico con las conexiones externas realizadas desde nuestra red.
- Obtener un logístico sobre los downloads de archivos realizados y quién los realizó.
- Obtener gráficos sobre tráfico en la red.
- Obtener logísticos sobre conexiones realizadas en horarios no normales (desde dónde, a qué hora y con qué destino).

*Mensualmente:*

- Realizar un seguimiento de todos los archivos logísticos a fin de detectar cambios (realizados con los archivos de back-up del mes anterior).

Cabría resaltar que, en gran parte, este procedimiento puede ser automatizado por medio de programas que realicen las tareas y sólo informen de las desviaciones con respecto a las reglas dadas.

## 4.2 - Procedimiento de alta de cuenta de usuario

Cuando un elemento de la organización requiere una cuenta de operación en el sistema, debe llenar un formulario que contenga, al menos los siguientes datos:

- Nombre y apellido
- Puesto de trabajo
- Jefe inmediato superior que avale el pedido
- Descripción de los trabajos que debe realizar en el sistema
- Consentimiento de que sus actividades son susceptibles de ser auditadas en cualquier momento y de que conoce las normas de "buen uso de los recursos" (para lo cual, se le debe dar una copia de tales normas).
- Explicaciones breves, pero claras de cómo elegir su password.

Asimismo, este formulario debe tener otros elementos que conciernen a la parte de ejecución del área encargada de dar de alta la cuenta, datos como:

- Tipo de cuenta
- Fecha de caducidad
- Fecha de expiración
- Datos referentes a los permisos de acceso (por ejemplo, tipos de permisos a los diferentes directorios y/o archivos)

Si tiene o no restricciones horarias para el uso de algunos recursos y/o para el ingreso al sistema.

## 4.3 - Procedimiento de baja de cuenta de usuario

Este procedimiento es el que se lleva a cabo cuando se aleja un elemento de la organización o cuando alguien deja de trabajar por un determinado tiempo (licencia sin goce de sueldo, vacaciones, viajes prolongados, etc.). En base a la explicación anterior hay, entonces, dos tipos de alejamientos: permanente y parcial.

La definición de si se da de baja o se inhabilita es algo importante pues, si se da de baja, se deberían guardar y eliminar los archivos y directorios del usuario, mientras que si sólo se inhabilita, no pasa de esa acción. Si el alejamiento del individuo no era permanente, al volver a la organización, la sección que había informado anteriormente de la ausencia, debe comunicar su regreso, por medio de un formulario dando cuenta de tal hecho para volver a habilitar la cuenta al individuo.

## 4.4 - Procedimiento para determinar las buenas passwords

Aunque no lo parezca, la verificación de palabras claves efectivas no es algo frecuente en casi ninguna organización. El procedimiento debe explicar las normas para elegir una password:

Se debe explicitar:

- La cantidad de caracteres mínimo que debe tener,
- No tiene que tener relación directa con las características del usuario.
- Debe constar de caracteres alfanuméricos, mayúsculas, minúsculas, números y símbolos de puntuación.
- Determinar, si es posible, el seguimiento de las palabras claves (llevar registros de las palabras claves anteriores elegidas por el usuario).

Una vez que el usuario ha elegido su password, se le debe correr un “programa crackeador” para tener idea de cuán segura es, en base al tiempo que tarda en romper la palabra.

#### **4.5 - Procedimientos de verificación de accesos**

Debe explicar la forma de realizar las auditorías de los archivos logísticos de ingresos a fin de detectar actividades anómalas. También debe detectar el tiempo entre la auditoría y cómo actuar en caso de detectar desviaciones.

Normalmente, este trabajo es realizado por programas a los que se les dan normativas de qué y cómo comparar. Escanean archivos de “log” con diferentes fechas tomando en cuenta las reglas que se le han dado. Ante la detección de un desvío, generan reportes informando el mismo.

En el procedimiento debe quedar perfectamente indicado quién es el responsable del mantenimiento de estos programas y cómo se actúa cuando se generan alarmas.

#### **4.6 - Procedimiento para el chequeo del tráfico de la red**

Permite conocer el comportamiento del tráfico en la red, al detectar variaciones que pueden ser síntoma de mal uso de la misma.

El procedimiento debe indicar el/los programas que se ejecuten, con qué intervalos, con qué reglas de trabajo, quién se encarga de procesar y/o monitorear los datos generados por ellos y cómo se actúa en consecuencia.

#### **4.7 - Procedimiento para el monitoreo de los volúmenes de correo**

Este procedimiento permite conocer los volúmenes del tráfico de correo o la cantidad de “mails” en tránsito. Dicho procedimiento se encuentra realizado por programas que llevan las estadísticas, generando reportes con la información pedida. El conocimiento de esta información permite conocer, entre otras cosas, el uso de los medios de comunicación, y si el servidor está siendo objeto de un “spam”.

Como en los casos anteriores, en el procedimiento debe estar

explicitado quién es el encargado del mantenimiento y del análisis de los datos generados, y qué hacer cuando se detectan variaciones.

#### **4.8 - Procedimientos para el monitoreo de conexiones activas**

Este procedimiento se efectúa con el objeto de prevenir que algún usuario deje su terminal abierta y sea posible que alguien use su cuenta. El procedimiento es ejecutado por medio de programas que monitorean la actividad de las conexiones de usuarios. Cuando detecta que una terminal tiene cierto tiempo inactiva, cierra la conexión y genera un log con el acontecimiento.

#### **4.9 - Procedimiento de modificación de archivos**

Este procedimiento sirve para detectar la modificación no autorizada y la integridad de los archivos y, en muchos casos, permite la traza de las modificaciones realizadas. Al igual que en los casos anteriores, debe estar bien determinada la responsabilidad de quién es el encargado del seguimiento y de actuar en caso de alarmas.

#### **4.10 - Procedimientos para el resguardo de copias de seguridad**

Este procedimiento debe indicar claramente dónde se deben guardar las copias de seguridad y los pasos a seguir en caso de problemas. Para lograr esto, deben estar identificados los roles de las personas que interactúan con el área, a fin de que cada uno sepa qué hacer ante la aparición de problemas.

#### **4.11 - Procedimientos para la verificación de las máquinas de los usuarios**

Este procedimiento permitirá encontrar programas que no deberían estar en las máquinas de los usuarios y que, por su carácter, pueden traer problemas de licencias y fuente potencial de virus. El procedimiento debe explicitar los métodos que se van a utilizar para la verificación, las acciones ante los desvíos y quién/quienes lo llevarán a cabo.

#### **4.12 - Procedimientos para el monitoreo de los puertos en la red**

Este procedimiento permite saber qué puertos están habilitados en la red, y, en algunos casos, chequear la seguridad de los mismos. El procedimiento deberá describir qué programas se deben ejecutar, con qué reglas, quién estará a cargo de llevarlo a cabo y qué hacer ante las desviaciones detectadas.

### **4.13 - Procedimientos de cómo dar a publicidad las nuevas normas de seguridad**

Este tipo de procedimiento no siempre es tenido en cuenta. Sin embargo, en una organización es muy importante conocer las últimas modificaciones realizadas a los procedimientos, de tal manera que nadie pueda poner como excusa “que no conocía las modificaciones”. En él, debe describirse la forma de realizar la publicidad de las modificaciones: puede ser mediante un mailing, por exposición en transparencias, por notificación expresa, etc.; quién estará a cargo de la tarea y las atribuciones que tiene.

Es fundamental tener en cuenta este último punto ya que un porcentaje de los problemas de seguridad, según está demostrado en estudios de mercado, proviene del desconocimiento de las normas y/o modificaciones a ellas por parte de los usuarios.

### **4.14 - Procedimientos para recuperar información**

Este procedimiento sirve para reconstruir todo el sistema o parte de él, a partir de las copias de seguridad. En él, deben explicarse todos los pasos a seguir para rearmar el sistema a partir de los back-up existentes, así como cada cuánto tiempo habría que llevarlos a cabo y quiénes son los responsables de dicha tarea.

### **4.15 - Check-Lists**

Las check-lists, como su nombre lo indica, son listas con un conjunto de ítems referentes a lo que habría que chequear en el funcionamiento del sistema.

Algunos ejemplos de check-lists:

- Asegurar el entorno. ¿Qué es necesario proteger? ¿Cuáles son los riesgos?
- Determinar prioridades para la seguridad y el uso de los recursos.
- Crear planes avanzados sobre qué hacer en una emergencia.
- Trabajar para educar a los usuarios del sistema sobre las necesidades y las ventajas de la buena seguridad
- Estar atentos a los incidentes inusuales y comportamientos extraños.
- Asegurarse de que cada persona utilice su propia cuenta.
- ¿Están las copias de seguridad bien resguardadas?
- No almacenar las copias de seguridad en el mismo sitio donde se las realiza
- ¿Los permisos básicos son de sólo lectura?
- Si se realizan copias de seguridad de directorios/archivos críticos, usar chequeo de comparación para detectar modificaciones no autorizadas.
- Periódicamente rever todo los archivos de “booteo de los

sistemas y los archivos de configuración para detectar modificaciones y/o cambios en ellos.

- Tener sensores de humo y fuego en el cuarto de computadoras.
- Tener medios de extinción de fuego adecuados en el cuarto de computadoras.
- Entrenar a los usuarios sobre qué hacer cuando se disparan las alarmas.
- Instalar y limpiar regularmente filtros de aire en el cuarto de computadoras.
- Instalar UPS, filtros de línea, protectores gaseosos al menos en el cuarto de computadoras.
- Tener planes de recuperación de desastres.
- Considerar usar fibras ópticas como medio de transporte de información en la red.
- Nunca usar teclas de función programables en una terminal para almacenar información de login o password.
- Considerar realizar autolog de cuentas de usuario.
- Concientizar a los usuarios de pulsar la tecla ESCAPE antes de ingresar su login y su password, a fin de prevenir los “Caballos de Troya”.
- Considerar la generación automática de password.
- Asegurarse de que cada cuenta tenga un password.
- No crear cuentas por defecto o “guest” para alguien que está temporariamente en la organización.
- No permitir que una sola cuenta esté compartida por un grupo de gente.
- Deshabilitar las cuentas de personas que se encuentren fuera de la organización por largo tiempo.
- Deshabilitar las cuentas “dormidas” por mucho tiempo.
- Deshabilitar o resguardar físicamente las bocas de conexión de red no usadas.
- Limitar el acceso físico a cables de red, routers, bocas, repetidores y terminadores.
- Los usuarios deben tener diferentes passwords sobre diferentes segmentos de la red.
- Monitorear regularmente la actividad sobre los gateways.

## TIPOS DE ATAQUES Y VULNERABILIDADES

### 5.1 - Negación de servicio (denial of service )

En el presente apartado, se describirán los modos de ataques que podrían ocurrir más frecuentemente en las redes de información. Debido a la pérdida de dinero y de tiempo que estos ataques pueden ocasionar, se presentarán también algunas formas de prevención y de respuesta a los mismos[9].

#### 5.1.1 - ¿Qué es “Denial of service”? Descripción de ataques.

Denial of service es un tipo de ataque cuya meta fundamental es la de negar el acceso del atacado a un recurso determinado o a sus propios recursos.

Algunos ejemplos de este tipo de ataque son:

- tentativas de “floodear” (inundar) una red, evitando de esta manera el tráfico legítimo de datos en la misma;
- tentativas de interrumpir las conexiones entre dos máquinas evitando, de esta manera, el acceso a un servicio;
- tentativas de evitar que una determinada persona tenga acceso a un servicio;
- tentativas de interrumpir un servicio específico a un sistema o a un usuario;

Cabría tener en cuenta que, el uso ilegítimo de recursos puede también dar lugar a la negación de un servicio. Por ejemplo, un “hacker” puede utilizar un área del FTP anónimo como lugar para salvar archivos, consumiendo, de esta manera, espacio en el disco y generando tráfico en la red.

Como consecuencia, los ataques de negación de servicio pueden esencialmente dejar inoperativa una computadora o una red. De esta forma, toda una organización puede quedar fuera de Internet durante un tiempo determinado.

#### 5.1.2 - Modos de ataque

Algunos ataques de negación de servicio se pueden ejecutar con recursos muy limitados contra un sitio grande y sofisticado. Este tipo de ataque se denomina “ataque asimétrico”. Por ejemplo, un atacante con una vieja PC y un módem puede poner fuera de combate a máquinas rápidas y sofisticadas. Ultimamente, esto es común con ataques de los denominados “nukes” en los cuales caen instalaciones grandes, por ejemplo, de clusters Windows NT.

Hay tres tipos de ataques básicos de negación de servicios:

- a.- Consumo de recursos escasos, limitados, o no renovables
- b.- Destrucción o alteración de información de configuración
- c.- Destrucción o alteración física de los componentes de la red

### **5.1.3 - Consumo de recursos escasos, limitados, o no renovables**

Las computadoras y las redes necesitan para funcionar ciertos recursos: ancho de banda de la red, espacio de memoria y disco, tiempo de CPU, estructuras de datos, acceso otras computadoras y redes, entre otros.

#### *Conectividad*

Los ataques de Negación de servicio se ejecutan, con frecuencia, contra la conectividad de la red. La meta del hacker es evitar que las computadoras se comuniquen en la red.

Un ejemplo de este tipo de ataque es el “SYN flood”:

En este tipo de ataque, el hacker comienza el proceso de establecer una conexión TCP a la máquina de la víctima, pero lo hace de manera tal que evita que la conexión se complete. En este tiempo, la máquina del atacado ha reservado uno entre un número limitado de las estructuras de datos requeridas para terminar la conexión inminente. El resultado es que las conexiones legítimas se rechazan mientras que la máquina del atacado se queda esperando para terminar esas falsas conexiones “medio abiertas”.

Debe tenerse en cuenta que este tipo de ataque no depende del ancho de banda que disponga el atacante. En este caso, el hacker está consumiendo las estructuras de datos del kernel, implicadas en establecer una conexión TCP. Un hacker con una simple conexión dial-up puede realizar este ataque contra una poderosa Workstation (este último es un buen ejemplo de un ataque asimétrico).

#### *Aprovechamiento de los recursos del otro*

Un hacker también puede utilizar los recursos que usted dispone contra usted mismo, de maneras inesperadas. Por ejemplo, el caso de Negación de servicio UDP. En este ataque, el hacker utiliza los paquetes “falsificados” de UDP para conectar el servicio de generación de eco en una máquina con el servicio de chargen en otra máquina.

El resultado es, que los dos servicios consumen todo el ancho de banda de red entre ellos. Así, la conectividad para todas las máquinas en la misma red desde cualquiera de las máquinas atacadas se ve afectada.

#### *Consumo de ancho de banda*

Un hacker puede, también, consumir todo el ancho de banda disponible en su red generando una gran cantidad de paquetes dirigidos a la misma. Típicamente, estos paquetes son de generación de eco de ICMP (ping), pero pueden ser cualquier otra cosa. Además, el hacker no

necesita operar desde una sola máquina; él puede poder coordinar varias máquinas en diversas redes para alcanzar el mismo efecto.

#### *Consumo de otros recursos*

Además del ancho de banda de la red, los hackers pueden consumir otros recursos que su sistema necesite para funcionar. Por ejemplo, en muchos sistemas, un número limitado de las estructuras de datos en el kernel está disponible para almacenar información de procesos (identificadores, entradas en tablas de procesos, slots, etc.).

Un hacker puede consumir estas estructuras de datos escribiendo un programa o un script que no haga nada pero que cree en varias ocasiones copias de sí mismo. Muchos sistemas operativos modernos, aunque no la totalidad de ellos, tienen recursos para protegerse contra este problema. Además, aunque las tablas de procesos no se llenen, se consume CPU por la gran cantidad de procesos y conmutación entre los mismos.

Un hacker puede también consumir su espacio en disco de otras maneras, por ejemplo:

- Generar miles de mails (Spam, Bombing. Para ampliar este tema, consultar el próximo).
- Generar intencionalmente errores que deben ser logueados. En este tipo de ataque, podemos citar también la utilización indebida del syslog en unix. Es decir, utilizar el proceso syslog de la víctima para que registre eventos de otra máquina, llenando el espacio en disco con el archivo de syslog.
- Colocar archivos en su disco, utilizando ftp anónimo.

En general, se puede utilizar cualquier cosa que permita que los datos sean escritos en su disco para ejecutar un ataque de negación de servicio si no hay límites en la cantidad de datos que se pueden escribir (quotas).

No obstante, muchos sitios tienen esquemas de “lockout” de cuenta después de un cierto número de logins fallados. Un setup típico bloquea el login después de 3 o 5 tentativas falladas. Un hacker puede utilizar este esquema para evitar que los usuarios legítimos entren. En algunos casos, incluso las cuentas privilegiadas, tales como root o administrator, pueden ser víctimas de este tipo de ataque.

Un hacker puede hacer caer su sistema o ponerlo inestable, enviando datos inesperados. Un ejemplo de tal ataque es el “ping flood” o Pings de tamaño demasiado grande. Si su sistema está experimentando caídas frecuentes sin causa evidente, podría deberse a este tipo de ataque.

Hay otros componentes que pueden ser vulnerables a la negación de servicio y que deben vigilar se. Estos incluyen:

- Impresoras
- Unidades de cinta
- Conexiones de red
- Otros recursos limitados importantes para la operación de su sistema.

#### **5.1.4 - Destrucción o alteración de la información de configuración**

Una computadora incorrectamente configurada puede no funcionar bien o directamente no arrancar. Un hacker puede alterar o destruir la información de configuración de su sistema operativo, evitando de esta forma la utilización de una computadora o red.

Veamos algunos ejemplos:

Si un hacker puede cambiar la información de ruteo de sus routers, su red puede ser deshabilitada.

Si un hacker puede modificar la registry en una máquina Windows NT, ciertas funciones pueden ser imposibles de utilizar, o directamente el sistema puede no volver a bootear.

#### **5.1.5 - Destrucción o alteración física de los componentes de la red**

Es muy importante la seguridad física de la red. Se debe resguardar contra el acceso no autorizado a las computadoras, los routers, los racks de cableado de red, los segmentos del backbone de la red, y cualquier otro componente crítico de la red.

#### **5.1.6 - Prevención y respuesta**

Tal como se ha expresado anteriormente, los ataques de Negación de servicio pueden dar lugar a pérdidas significativas de tiempo y dinero para muchas organizaciones, por lo que se recomiendan una serie de medidas:

- Colocar access lists en los routers. Esto reducirá su exposición a ciertos ataques de negación de servicio
- Instalar patches al sistema operativo contra flooding de TCP SYN. Esta acción permitirá reducir sustancialmente su exposición a estos ataques aunque no pueda eliminar el riesgo en forma definitiva.
- Invalidar cualquier servicio de red innecesario o no utilizado. Esto puede limitar la capacidad de un hacker de aprovecharse de esos servicios para ejecutar un ataque de negación de servicio. Por ejemplo: chargen, Echo, etc.
- Si el sistema operativo lo permite, implementar sistemas de cuotas. Por ejemplo, si el sistema operativo soporta "disk Quotas" impleméntelo para todos los logins. Si el sistema operativo soporta partición o volúmenes, separe lo crítico de lo que no lo es.
- Observar el funcionamiento del sistema y establecer valores base para la actividad ordinaria. Utilice estos valores para calibrar niveles inusuales de la actividad del disco, del uso de la CPU, o del tráfico de red.
- Incluir como parte de la rutina, el examen de la seguridad física.
- Considerar, entre otras cosas, los servidores, routers, terminales desatendidas, ports de acceso de red y los gabinetes de cableado.

- Utilizar Tripwire o una herramienta similar para detectar cambios en la información de configuración u otros archivos.
- Tratar de utilizar configuraciones de red redundantes y fault-tolerant.

## 5.2 - Cracking de passwords

En este apartado, se presentarán una serie de consideraciones referidas al “cracking de passwords” basadas en UNIX<sup>1</sup>.

El objetivo inicial consiste en entrar al server. Para ello, se procede como si se tratase de una máquina remota (telnet). Pero, debido a que se permite el acceso a múltiples usuarios, los sistemas UNIX nos solicitarán un nombre de identificación acompañado de una clave. Dicho nombre darse de alta en el sistema para que se pueda acceder.

Cuando un usuario desea entrar en una máquina, el sistema solicitará:

Un login de acceso o nombre de usuario. Si el login es incorrecto, el sistema no lo notificará para impedirle conocer qué accesos se encuentran dados de alta.

Una password o palabra clave. Si la password coincide con la que tiene asignada el login que se emplea, el sistema permitirá el acceso.

### 5.2.1 - El archivo “/etc/passwd”: descripción

Los usuarios que se encuentran dados de alta en el sistema, así como las passwords que emplean, se hallan localizados en el archivo: /etc/passwd (para la mayoría de los sistemas operativos basados en UNIX).

Lamentablemente para algunos, y afortunadamente para otros, este archivo es el punto más débil del sistema. Está compuesto de líneas o registros en las cuales cada línea se divide en siete campos con dos puntos (:).

Veamos un ejemplo:

```
kaiser:j99sE3vc23:105:100:El Kaiser:/usr/var1:/bin/ksh
```

campo 1: kaiser

Es el “username”, nombre de usuario, “login” que se emplea para acceder al sistema.

campo 2: j99sE3vc23

Es la password encriptada.

---

<sup>1</sup> Los servidores conectados a la Red necesitan de un sistema operativo que permita gestionar la cantidad de procesos y usuarios que pueden encontrarse simultáneamente en el lugar. Los sistemas operativos más empleados son los basados en UNIX y Windows NT.

Si el campo contiene un asterisco (\*), indica que la cuenta no se puede utilizar. Si todos los campos poseen el asterisco u otro signo significa que las claves están en un archivo "shadow". En caso de que esto no ocurra, significa que la cuenta no posee ninguna contraseña de acceso.

Dado su sistema de encriptación, una vez que el usuario introduce la password, el sistema operativo lo encripta y verifica con el que se encuentra en este archivo. En caso de que sean iguales, la password introducida es correcta.

Si los dos últimos caracteres de este campo van precedidos de una coma, esto indica la fecha en que expira el empleo de esta contraseña.

campo 3: 105

Es el UID, número de usuario de sistema. Puede ser de 0 a 60000. Por lo general, se comienza a partir del 100. El UID con valor 0 indica que la persona tiene nivel de "super usuario", esto es, aquel que tiene todos los privilegios de acceso en el sistema.

campo 4: 100

Es el GID, número del grupo al que pertenece el usuario. Puede ser de 0 a 60000.

El 0 también se corresponde con el grupo de "super usuario" o root. Los usuarios del mismo grupo tienen el mismo nivel de privilegios para aquellos elementos que pertenezcan a su dominio.

campo 5: El Kaiser

Es el "comment", es decir, datos personales del usuario (nombre completo, teléfono, etc.).

campo 6: usr/var1

Es el "home directory", directorio donde se ubica al usuario. Su directorio de trabajo.

campo 7: /bin/ksh

Es el "shell" o intérprete de comandos empleado por el usuario. Depende de cuál emplee, podrá realizar determinadas acciones.

Unix encripta las contraseñas mediante un mecanismo del kernel, crypt. Este es un algoritmo basado en el estándar de encriptación de datos (DES) desarrollado por el Instituto Nacional de Estándares y Tecnología (NIST).

El estándar DES transforma la información de texto plano en datos encriptados, texto cifrado, mediante el uso de un algoritmo especial y valor semilla llamado clave.

No debe confundirse el comando crypt(3) con el crypt(1). Este último es

mucho más inseguro y esta basado en el dispositivo Enigma, utilizado por las fuerzas armadas alemanas en la Segunda Guerra Mundial.

En crypt (3) el texto plano se encripta en un grupo de ceros. Posteriormente el texto cifrado resultante es encriptado de nuevo con la password del usuario como clave. Este proceso se repite 25 veces. Una vez finalizado los 64 bits se dividen en 11 caracteres y se guardan en el archivo `/etc/passwd` o se guardan en el archivo `shadow`.

También suele utilizarse lo que en criptología se denomina "grano de sal", dato de variabilidad, etc... Se trata de un valor de 12 bits que utiliza para modificar el resultado de la función DES. Es decir, un valor que puede variar de 0 a 4095. Así, para cada contraseña posible existen 4096 formas de encriptación y almacenamiento.

Cuando recurrimos al programa `/bin/passwd` para introducir una nueva contraseña, dicho programa utiliza un "grano de sal" basado en la hora del sistema. Esta última es usada para la función de cálculo de la contraseña encriptada. Esta sal es guardada junto con la contraseña en el archivo `/etc/passwd` (en el caso de que no este instalado el soporte de `shadow password`). De esta forma, los dos primeros caracteres de una contraseña son en realidad el "grano de sal":

E67hfr83cEr23

E6 son el "grano de sal". Cuando ingresemos nuevamente al sistema y nos pida nuestra password el programa encriptará la palabra que le demos utilizando ese mismo "grano de sal". Posteriormente, comparará el resultado de esta encriptación con la password almacenada para comprobar si coinciden. Es decir, en ningún momento se desencripta la password. Lo que, en realidad, realiza el sistema es encriptar la palabra que le suministramos y comparar con el original.

### 5.2.2 - Descubrir una password

Una vez encriptada una password, no se puede desencriptar. Sin embargo, esto no garantiza la seguridad de la password, puesto que no significa que la password no se pueda averiguar.

El mecanismo que se utiliza para descubrir (no desencriptar) las passwords consiste en efectuar encriptaciones de palabras (posibles passwords) y comparar estas encriptaciones con el original.

¿De que depende el éxito?

El éxito depende de la calidad del diccionario (archivo que contiene un conjunto de posibles passwords), del programa que se utilice, del CPU y, por supuesto, de nuestra paciencia.

Los programas buscadores de contraseñas son fácilmente diseñables.

Si mediante un "bug" se obtiene el archivo `/etc/passwd`, se puede iniciar un ataque de diccionario contra el mismo obteniéndose, de este modo, las passwords.

Otro tipo de ataque es el de "fuerza bruta", que consiste simplemente en realizar todas las combinaciones posibles de caracteres hasta hallar la

password.

En el siguiente cuadro podemos ver el tiempo de búsqueda de una password de acuerdo a la longitud y tipo de caracteres utilizados. Se supone una velocidad de búsqueda de 100.000 passwords por segundo.

Long. En caracteres	26 letras (minúsculas)	36 letras y dígitos	52 (mayúsculas y minúsculas)	96 Todos los caracteres
6	50 minutos	6 horas	2.2 días	3 meses
7	22 horas	9 días	4 meses	23 años
8	24 días	10.5 meses	17 años	2287 años
9	21 meses	32.	881 años	219.000 años
10	45 años	1159 años	45.838 años	21 millones de años

Como puede apreciarse, resulta importante utilizar más de 8 caracteres y cuantos más símbolos intervengan, menos probabilidades habrán de encontrar la password.

### 5.3 - E-mail bombing y spamming

En este apartado, se presentarán algunas de las dificultades que pueden surgir como consecuencia de la utilización de los servicios de mail.

#### 5.3.1 - Descripción

El e-mail bombing consiste en enviar muchas veces un mensaje idéntico a una misma dirección, saturando el mailbox del destinatario.

El spamming, que es una variante del e-mail bombing, se refiere a enviar el e-mail a centenares o millares de usuarios e, inclusive, a listas de interés. El Spamming puede resultar aún más perjudicial si los destinatarios contestan el mail, haciendo que todos reciban la respuesta.

Puede, además, ocurrir inocentemente como resultado de enviar un mensaje a la lista y no darse cuenta de que la lista lo distribuye a millares de usuarios, o como resultado de mala configuración de un autorespondedor, por ejemplo el "vacation".

El e-mail bombing/spamming se puede combinar con el e-mail spoofing - que altera la identidad de la cuenta que envía el mail -, logrando que sea más difícil determinar quién está enviando realmente el mail.

### 5.3.2 - Detalles técnicos

Cuando se proveen los servicios de e-mail los usuarios son, lógicamente, vulnerables al e-mail bombing y spamming.

En efecto, el e-mail spamming es casi imposible de prevenir. Un usuario con una dirección válida de mail puede realizar " Spam " a cualquier otra dirección de mail, newsgroup, o sistema de BBS.

Cuando gran cantidad de mails son dirigidos a un solo sitio, éste puede sufrir "denial of service" por pérdida de conectividad, caerse el sistema o producirse fallas en el servicio debido a:

- sobrecarga de conexiones de red;
- utilización de todos los recursos de sistema disponibles;
- llenado del disco como resultado de postings múltiples y de entradas en el "syslog".

### 5.3.3. - ¿Cómo proceder?

#### *Detección*

Si un sistema aparece repentinamente lento (el e-mail es lento o no parece ser enviado o recibido), la razón puede ser que su mailer está intentando procesar una excesiva cantidad de mensajes. Esto puede comprobarse a través del "log" de sistema.

#### *Reacción*

Es importante:

- Identificar la fuente del e-mail bomb/spam y configurar el router para evitar el acceso de los paquetes entrantes de esa dirección. Puede colocar un "access list" en el port 25 (SMTP) del tipo "established" para esa dirección.
- Observar los "headers" del e-mail para determinar su origen verdadero.
- Ponerse en contacto con el sitio que usted identificó en su revisión con el propósito de alertarlos de la actividad del spammer.
- Asegurarse de tener la versión más actualizada del "daemon" de mail (por ejemplo sendmail) y aumente el grado de "debug" o "log" que posea el proceso, para detectar o alertar estas actividades. Tenga la precaución de vigilar el tamaño del archivo de log, que puede crecer considerablemente, si se está bajo un e-mail-bombing.

#### *Prevención*

Desafortunadamente, hasta el momento, no hay manera de prevenir el bombardeo de e-mail o spamming y es imposible predecir el origen del ataque siguiente. Es trivial obtener acceso a listas de interés o acceder a información que contenga grandes volúmenes de direcciones de e-mail, las que proporcionan al atacante direcciones de

destino para el spam.

Pueden desarrollarse herramientas internas, que pueden ayudar a reconocer y a responder al e-mail bombing/spamming reduciendo, de esta manera, el impacto de tal actividad. Tales herramientas deben aumentar las capacidades de log y alertar de mensajes que vienen de un mismo lugar en un corto período de tiempo. Asimismo, deberían ser capaces de rechazar esos mensajes, o descartarlos.

Si un sitio utiliza un número pequeño de servidores de e-mail, podría configurarse un "firewall" para asegurarse de que las conexiones de "smtp" fuera de su firewall puedan hacerse solamente a sus "hubs" de mail y a ninguno de los otros equipos. Aunque esta operación no prevendrá un ataque, reduce al mínimo el número de las máquinas disponibles para un ataque basado en SMTP. De este modo, se puede controlar el tráfico entrante SMTP y filtrarlo de manera acorde.

## **5.4 - Problemas de seguridad en el FTP**

### **5.4.1 - El comando PORT**

En los últimos años, se ha incrementado el debate en torno a los problemas relacionados con el comando PORT del protocolo del FTP. Estos problemas se basan en el uso erróneo de dicho comando.

### **5.4.2. - El Protocolo FTP**

Para entender estos ataques, es necesario tener una comprensión básica del protocolo FTP.

Un cliente abre una conexión al port de control de ftp (21) de un FTP SERVER. De este modo, para que el servidor sea capaz luego de enviar datos de nuevo a la máquina del cliente, una segunda conexión (de datos) debe abrirse entre el servidor y el cliente.

Para hacer esta segunda conexión, el cliente envía un comando PORT al servidor. Este comando incluye parámetros que indican al servidor cuál IP ADDRESS conectar y qué port abrir en aquella dirección.

El servidor luego abre aquella conexión, siendo la fuente de la conexión el port 20 del servidor y el destino el port identificado en los parámetros del comando PORT.

El comando PORT se utiliza generalmente sólo en el " modo activo " del ftp (por default). No se suele utilizar en modo pasivo (PASV). Debe notarse que los servidores de ftp generalmente implementan ambos modos en ejecución, y el cliente especifica qué método utilizar.

### **5.4.3. - El ataque "Ftp bounce"**

Conforme con el protocolo FTP, el comando PORT hace que la máquina que lo origina especifique una máquina de destino y un port arbitrarios para la conexión de datos. Sin embargo, esto también significa

que un hacker puede abrir una conexión a un port del hacker eligiendo una máquina que puede no ser el cliente original.

Hacer esta conexión a una máquina arbitraria es hacer un ataque “ftp bounce”.

Con fines ilustrativos, se presentan seguidamente varios ejemplos de cómo los hackers pueden utilizar el “ftp bounce”.

#### “Scanning” de ports

Un hacker que desea realizar una port scan contra un sitio puede hacerlo de un server FTP de un tercero, que actúa como un “puente” para el scan. El sitio de la víctima ve la exploración como procedente del server FTP más que de la fuente verdadera (el cliente FTP).

Bajo algunas circunstancias, esta técnica ofrece al hacker más ventajas que ocultar la fuente verdadera de la prueba. Cuando el sitio previsto de la víctima está en la misma subnet que el server FTP server, o cuando no filtra tráfico del server FTP, el hacker puede utilizar la máquina del servidor como la fuente del port scan más que la máquina del cliente, desviando de esta manera los controles de acceso que de otra manera se aplicarían.

#### “Bypass” de dispositivos básicos de filtrado de paquetes.

Un hacker puede realizar un “bypass” de un firewall en ciertas configuraciones de red.

Por ejemplo, supongamos que un sitio tiene su servidor de FTP anónimo detrás del firewall. Usando la técnica de port scan, un hacker determina que un web server interno en ese sitio está disponible en el acceso 8080, un port normalmente bloqueado por un firewall.

Conectándose al server FTP público del sitio, el hacker inicia otra conexión entre el server FTP y un port arbitrario, en una máquina no pública del sitio (por ejemplo el web server interno en el port 8080). Como resultado, el hacker establece una conexión a una máquina que sería protegida de otra manera por el firewall.

#### ***5.4.4 - Bypass de dispositivos de filtrado dinámicos***

Otro problema se refiere a los sitios que tienen firewalls que utilizan filtros dinámicos para protegerse. Los sitios están abiertos al ataque porque el firewall confía en la información que recibe.

En este ejemplo, el sitio de la víctima contiene todos sus sistemas detrás de un firewall que utiliza los filtros dinámicos. Una persona en el sitio de la víctima hojea las páginas de la Web y baja un Java applet construido por el hacker. Sin el conocimiento de esa persona, el Java applet abre una conexión de salida de ftp a la máquina del hacker. El applet entonces publica un comando PORT de ftp, ordenando a la máquina del servidor abrir una conexión a, por ejemplo, el port telnet que de otra manera se encontraba protegido detrás del firewall.

Como el firewall de filtros dinámicos examina los paquetes de salida para determinar si alguna acción se requiere de su parte, observa el

comando PORT y permite una conexión entrante del server web remoto al port del telnet en la máquina de la víctima. Esta conexión normalmente no es permitida por el firewall; fue permitida en este caso porque el comando PORT fue realizado por el cliente.

### 5.4.5 - Soluciones

Los ataques de los ejemplos demuestran el componente base de la vulnerabilidad: los contenidos del comando PORT del ftp no son tan dignos de confianza mientras están bajo control de un potencial atacante. El ejemplo del “ftp bounce” demuestra qué sucede cuando un servidor confía en la información. El ejemplo del filtro dinámico demuestra qué sucede cuando un firewall confía en la información.

#### *Software del Ftp server*

La mejor solución al problema del “ftp bounce” desde la perspectiva de la seguridad es asegurarse de que el software del server FTP no puede establecer conexiones a máquinas arbitrarias. Sin embargo, los sitios que confían en el comportamiento “RFC-compliant” pueden encontrar que el implementar esta solución afectará las aplicaciones que ellos utilizan. Por lo tanto, muchos vendedores ofrecen soluciones que permiten al sitio dar servicio de ftp adaptado a las necesidades del cliente. Las implementaciones del vendedor caen en tres grupos:

- 1) conformidad estricta con funciones del RFC: el comando PORT se puede utilizar para conectar directamente con una máquina de una tercera persona, y ésta es la única funcionalidad permitida. Algunos vendedores que eligen mantener conformidad estricta, han tratado este problema modificando el resto de los servicios de red para rechazar las conexiones que se originaban en el port de datos del ftp (port 20).
- 2) supresión estricta del comando PORT: el comando PORT puede ser utilizado para conectar con el cliente de origen, y ésta es la única funcionalidad permitida.
- 3) comportamiento variable del comando PORT: el comando PORT se puede utilizar en las dos formas descritas, siendo una la forma por default. El cambiar entre ellas se logra generalmente con un parámetro en la línea de comando. Se debe tener cuidado de verificar cuál es el valor por default.

Asimismo, se debe tener conciencia sobre la categoría en que se halla el software del server. La recomendación es utilizar la opción 2, o la opción 3 con la supresión habilitada.

#### *Configuración de Red*

Hay algunas cosas a tener presente al configurar las “fronteras” de la red, esto es, los routers con access-lists y los firewalls.

Los sitios deben asegurarse de que se diseñe cuidadosamente la topología de red de modo que los límites eficaces del tráfico existan

entre los sistemas que ofrecen niveles distintos del servicio. Por ejemplo, un sitio tiene típicamente un servicio de FTP Anonymous, servicio del Web, y un hub entrante de correo electrónico. Una buena práctica de seguridad consiste en separar las máquinas que proporcionan estos servicios externos de las que realizan servicios internos. Es importante tener límites “fuertes” en la red, preferiblemente firewalls, entre estos dos conjuntos de máquinas.

Por ejemplo, los sitios que tienen un server FTP que permite el comando PORT para establecer conexiones a las máquinas de un tercero deben bloquear el tráfico entre el server FTP y las máquinas que ofrecen servicios que confían en el hostname o la dirección IP para la autenticación. Los ejemplos de tales servicios son rlogin, rsh y NFS. Mientras que un firewall o un filtering router debe prevenir siempre el acceso externo directo a tales servicios, debe también filtrar el tráfico de un server FTP interno que se comporte de esta manera. Esto advierte al server FTP que está siendo utilizado como una máquina de relay para atacar protocolos con mecanismos débiles de autenticación basados en el hostname o la dirección IP.

Los sitios que usan firewall de filtrado dinámico de paquetes dinámico necesitan tomar medidas adicionales para asegurarse de que los comandos PORT de terceros sean bloqueados por el firewall.

#### **5.4.6 - Otros problemas del FTP**

El FTP y los programas que lo implementan son reales problemas para los encargados de seguridad de los sistemas. Veamos una lista parcial de los mismos:

- El protocolo, como hemos visto, usa dos conexiones TCP, complicando el trabajo de controlarlo a través de un firewall. En la mayoría de los casos un control de una conexión saliente requiere una conexión entrante de datos.
- El demonio ftpd corre inicialmente como root, ya que normalmente procesa un login a determinada cuenta, incluyendo el procesamiento de la password. Peor aun, no puede dejar su privilegio despues del login, el protocolo requiere conexión al port 20 el cual esté en el rango privilegiado.
- Históricamente, han habido bugs en la implementación del demonio, lo cual ha producido grandes problemas de seguridad

Por otra parte, el FTP anónimo se ha convertido en un standard de internet para distribuir software, documentos, etc. No hay duda que es un servicio útil, pero debe ser administrado con sumo cuidado.

La primera regla es que ningún archivo o directorio en el area de FTP anónimo debe ser poseida por el login ftp , ya que el FTP anónimo corre con esa identificación de usuario.

La siguiente regla es evitar colocar un archivo real /etc/passwd en el area de FTP anónimo. Hay que crear aquí un /etc/passwd “dummy”, con cuentas inexistentes y sin passwords reales encriptadas. En muchos caso se ha colocado aquí el /etc/passwd real, dando servido al hacker las passwords encriptadas

para así hacer un ataque de diccionario.

Crear o no un directorio público de acceso read/write es tema de controversia. No hay duda que es útil hacerlo, pero se puede abusar fácil de ello. Uno puede encontrarse con su server convertido en repositorio de software pirata, por ejemplo. Este repositorio puede ser temporario o permanente, en el primer caso, hackers pueden usar su sitio como lugar de tránsito, consumiendo sus recursos.

## 5.5 - TFTP

El Trivial File Transport Protocol (TFTP) es un mecanismo sencillo de file transfer basado en UDP. Este protocolo no tiene autenticación, constituyendo un potencial problema de seguridad. Es usado frecuentemente para bootear estaciones de trabajo X11, o para bootear routers desde sistemas unix, por ejemplo.

El servicio TFTP, correctamente configurado, restringe el file transfer a uno o dos directorios, típicamente `/usr/local/boot` o `/etc/tftpboot`, según la variante de UNIX utilizada.

Hasta no hace demasiado tiempo, la mayoría de los vendedores liberaban su software con acceso irrestricto al TFTP. Esto hacia el trabajo de los hackers sencillo:

```
$ tftp victima.com.ar
tftp> get /etc/passwd
/tmp/passwd Received 4670
bytes in 0.8 seconds tftp> quit
$ crack </tmp/passwd
```

Un ataque de diccionario contra el `/etc/passwd` da normalmente con el 25% de las passwords.

Se recomienda **NO HABILITAR** el tftp a menos que sea estrictamente necesario. Si se lo hace, verificar que este correctamente configurado, para enviar solo los archivos correctos a solo los clientes autorizados.

## 5.6 - TELNET

TELNET provee acceso de terminal a un sistema. El protocolo incluye provisiones para soportar varios seteos de terminal como ser raw mode, eco de caracteres, etc. Generalmente, el demonio de telnet llama al programa login para autenticar al usuario e iniciar la sesión. El mismo provee un nombre de cuenta y una password para el login.

Una sesión de telnet puede ocurrir entre dos máquinas de la misma organización o confiables, en ese caso se puede utilizar un secure telnet para encriptar la sesión completa, protegiendo la password y la sesión completa.

Pero en la mayoría de los casos, la mayoría de las sesiones de telnet

vienen de sistemas no confiables. Es decir, no podemos confiar ni en el sistema operativo que hace telnet al nuestro, ni en las redes que intervienen en el proceso. La password y la sesión entera son fácilmente visible para los ojos de un espía, típicamente usando sniffers.

Una técnica común de hackeo es “pinchar” el programa cliente de telnet, logrando que registre los nombres de usuario y password, e inclusive la sesión entera.

De todas formas, si la red está bajo “sniffing”, es extremadamente sencillo obtener las passwords que circulan por sesiones de telnet. La mejor defensa para este tipo de ataque es el esquema de password de única vez.

Una de las implementaciones de este esquema consiste en que el usuario disponga de un dispositivo programado mediante una clave secreta. El sistema que acepta el login envía un “challenge”, que el usuario digita en su dispositivo. Esto le devuelve la password adecuada para el código “challenge” enviado. Pero esa password que circula por la red es válida solo para esa sesión, el hacker, si observa la sesión, deberá descifrar cual es el algoritmo utilizado para que en base al “challenge” variable y una clave secreta que no circula por la red se obtenga la password de única vez.

### 5.7 - Los comandos “r”

Los comandos “r” provienen del sistema de autenticación del UNIX BSD. Un usuario puede realizar un rlogin a una máquina remota sin ingresar password si el criterio de autenticación es el correcto. Estos criterios consisten en:

- La conexión debe originarse desde un port TCP privilegiado. En sistemas como PC's con Win95, por ejemplo, estas restricciones no existen con lo cual no tienen mucho sentido. Como corolario, rlogin y rsh deben ser permitidos sólo desde máquinas donde esta restricción exista.
- El usuario y la máquina cliente deben estar listados en la máquina server como socios autenticados. ( Típicamente /etc/hosts.equiv o en el directorio home del usuario, en el archivo .rhosts )
- La máquina cliente y su dirección IP deben coincidir, estando listadas en el server.

Desde el punto de vista del usuario, este esquema es muy interesante. El usuario no es molestado con prompts de passwords en logins que utiliza frecuentemente. Pero desde el punto de vista del hacker, los comandos “r” ofrecen dos ventajas: una manera de entrar a un sistema, y una vez dentro, una forma de ganar acceso a máquinas de confianza de la primera máquina hackeada.

El principal objetivo del hacker es colocar una entrada apropiada en /etc/hosts.equiv o .rhosts. Para ello utilizan FTP, UUCP, TFTP u otros medios. Por ejemplo, pueden utilizar FTP para dejar .rhosts en /usr/ftp. o UUCP, para dejarlo en /usr/spool/uucppublic. Obviamente, uno debe verificar la estructura de permisos de la máquina server para prohibir

eso.

Una vez adquirido el acceso no autorizado, muchas otras computadoras son accesibles. El hacker accede a `/etc/hosts.equiv` de la máquina atacada, y de ahí puede seguir su cadena de accesos, obteniendo más archivos `/etc/passwd`.

Notemos que la implementación de comandos “r” presenta un problema adicional: Parte de la seguridad del sistema puede residir en decisiones del usuario y no del administrador. En efecto, el usuario puede hacer que su archivo `.rhosts` sea de lectura y escritura para todos los otros usuarios. Algunas implementaciones de `rlogin` y `rsh` solucionan esto: si el usuario no lo hace, un cron se ocupa que los archivos `.rhosts` estén con sus permisos en orden.

Dado las debilidades del sistema de autenticación de los comandos “r” que hemos visto, no se recomienda que estos servicios estén disponibles en sistemas accesibles directamente en internet.

Aquí hay un punto delicado. La alternativa usual a emplear `rlogin` es usar `telnet`, que como hemos visto transmite por la red una password, mientras que `rlogin` no lo hace. Las alternativas y los riesgos deben ser cuidadosamente evaluados.

## **5.8 - Seguridad en NetBIOS**

Los sistemas operativos de Microsoft utilizan para comunicarse entre sí el protocolo NetBIOS (Network Basic Input Output System), desarrollado originalmente por IBM. Este protocolo a su vez debe ir sobre otro de inferior nivel que puede ser uno de los siguientes: NetBEUI, IPX/SPX, TCP/IP; es por ello que hablaremos de NetBIOS sobre TCP/IP o NetBIOS sobre NetBEUI. Otras aplicaciones y servicios acceden a la red utilizando directamente IPX/SPX o TCP/IP pero sin utilizar NetBIOS. A la implementación de NetBIOS sobre TCP/IP se la conoce como NBT. NetBIOS nos permite compartir archivos e impresoras en la red Microsoft.

### **5.8.1 - Qué hacer**

Los ports usados por el servicio de NetBIOS/TCP ( NBT ) deben ser IMPERIOSAMENTE filtrados en el router que vincula la LAN con Internet. Además de permitir que un usuario cualquiera de la red acceda a shares de discos de la instalación, la implementación de NBT particularmente en Windows 95 contiene bugs que lo hace vulnerable a ataques del tipo “WinNUKE”, como el OOB ( Out of Band) bug. Si bien estos ataques no comprometen la integridad de los datos, producen la caída del equipo o de al menos el stack de protocolo, dejando a la máquina aislada.

## DESCRIPCIÓN DE ALGUNAS HERRAMIENTAS DE CONTROL Y SEGUIMIENTO DE ACCESOS

En este apartado se encuentran aquellas herramientas que nos permitirán tener una información - mediante archivos de trazas o logísticos - de todos los intentos de conexión que se han producido sobre nuestro sistema o sobre otro que nosotros hayamos señalado, así como intentos de ataque de forma sistemática a puertos tanto de TCP como de UDP (herramientas de tipo SATAN)[9].

Este tipo de herramientas nos permite tener un control sobre todos los paquetes que entran por la interfaz de red de la máquina: IP (TCP, UDP) e ICMP, o analizando paquetes a nivel de aplicaciones (TELNET, FTP, SMTP, LOGIN, SHELL, etc.). Estas herramientas pueden ser utilizadas junto con otras que nos permitan definir desde qué máquinas permitimos ciertas conexiones y cuales se prohíben. Algunas de las herramientas descritas en este apartado no necesitan estar instaladas en la máquina que se quiere controlar, ya que se puede poner en una máquina cuya interfaz de red funcione en modo promiscuo, permitiendo seleccionar la dirección IP o máquina que queremos auditar.

Algunas de las herramientas descritas en este apartado pueden tener un doble uso. Es decir, nos permiten protegernos ante posibles ataques, pero también podrían ser utilizadas para intentar comprometer los sistemas. Por eso es importante que el uso de estas herramientas esté restringido - en la manera que se pueda - para que no todo el mundo esté utilizándolas de forma aleatoria y nos oculten realmente un ataque. También podrán ser utilizadas para realizar seguimientos en la red cuando creamos que alguna de nuestras máquinas ha sido comprometida.

Las herramientas que permiten este tipo de operatividad son: tcp-wrapper, netlog, argus, tcpdump, SATAN, ISS, courtney, gabriel, nocol, tcplist.

### 6.1 - TCP-WRAPPERS

El tcp-wrappers es un software de dominio público desarrollado por Wietse Venema (Universidad de Eindhoven, Holanda). Su función principal es: proteger a los sistemas de conexiones no deseadas a determinados servicios de red, permitiendo a su vez ejecutar determinados comandos ante determinadas acciones de forma automática.

Con este paquete podemos monitorear y filtrar peticiones entrantes a distintos servicios TCP-IP, como: SYSTAT, FINGER, FTP, RLOGIN, RSH, REXEC, TFTP, TALK. El software está formado por un pequeño programa que se instala en el "/etc/inetd.conf". Una vez instalado, se pueden controlar los accesos mediante el uso de reglas y dejar una traza de todos los intentos de conexión tanto admitidos como rechazados (por servicios, e indicando la máquina que hace el intento de conexión).

Veremos, en primer lugar, el tema de las trazas que genera este software.

El programa utiliza el syslogd (puerto 514 udp) para mandar esa

información; por defecto utilizará la salida de mail, la cual estará indicada en el archivo de configuración de syslogd (/etc/syslog.conf) con la línea mail.debug. Esto se puede cambiar en los fuentes del programa y se puede re-dirigir a otro lugar mediante el uso de las variables de usuario que deja libres el syslogd (LOCAL\_0,...LOCAL\_7, estas variables vienen definidas en el archivo /usr/include/syslog.h). Una vez modificados las fuentes, se deberá indicar al syslogd donde debe dejar la información de esa variable local.

En referencia al control de conexiones a distintos servicios, el software utiliza dos archivos de información (hosts.allow, hosts.deny) situados en el directorio "/etc". Es en estos archivos donde se definirán las reglas que deben utilizarse para el filtrado de los paquetes. El filtrado se puede realizar teniendo en cuenta tanto a máquinas como a servicios o una mezcla de ambos. En el caso de las máquinas hay varias formas de hacerlo. Por ejemplo se le puede indicar que sólo se conecten las máquinas pertenecientes al mismo dominio (esto se puede ampliar a los que tengan el mismo rango de direcciones IP, para evitar que máquinas no definidas en el DNS no puedan conectarse), o sólo aquellas cuyos nombres sean especificados de forma explícita.

Veremos, a continuación, un ejemplo que consiste en permitir las conexiones sólo de aquellas máquinas de mi dominio. Para ellos tendríamos que disponer de lo siguiente:

```
hosts.deny
    ALL:
    ALL
hosts.allow
    ALL: LOCAL,
    sfp.gov.ar
```

La secuencia es la siguiente: en el primer archivo denegamos todas las conexiones; mientras que en el segundo, permitimos las conexiones de las máquinas definidas en mi mismo dominio.

Una utilidad que puede ser interesante a la hora de tener información de conexiones de forma automática es el uso de comandos en estos archivos. Es decir, podemos decirle al programa que cuando se produzcan ciertas conexiones ejecute un comando.

Veamos un ejemplo:

```
hosts.deny
    ALL: ALL (/usr/ucb/finger -l @ % h | /usr/ucb/mail -s % d% h e-mail) &
hosts.allow
    ALL: LOCAL, uc3m.es
```

Según este ejemplo, cuando se produzca una conexión no deseada, de forma automática se ejecutará un finger a la máquina que origine esa conexión y el resultado del mismo, se mandará vía correo electrónico al usuario especificado (el administrador de la máquina o el responsable de seguridad de la organización), indicando en el "subject" del mensaje el al cual se servicio intento la conexión y la máquina que lo originó, y como cuerpo el resultado del finger sobre esa máquina.

Además de estas reglas podemos incluir protocolos específicos de control, veamos un pequeño ejemplo de esta utilidad:

```
hosts.deny
  ALL:
  ALL
hosts.allow
  in.ftpd: LOCAL,
  sfp.gov.ar
```

Según este ejemplo, sólo permitimos conexiones ftp desde nuestro dominio, prohibiendo las demás conexiones que estén filtradas por el programa.

Estos ejemplos son muy básicos, pero el grado de complejidad de las reglas puede aumentar incluyendo distintos protocolos y listas de máquinas por protocolos.

Como mencionábamos anteriormente, este tipo de programa genera trazas de las conexiones establecidas. Veremos, a continuación, unas recomendaciones referentes a las trazas que pueden extenderse a otro tipo de utilidades.

Es aconsejable disponer de una o varias máquinas para centralizar las trazas que creamos convenientes.

Describiremos ahora una posible organización para tener información de todas las conexiones que se producen en nuestro grupo de máquinas.

Podemos clasificar nuestras máquinas por sistema operativo o por funciones que realizan. A cada uno de estos grupos se le asigna una variable en el syslog (como veíamos anteriormente), y se envía (vía syslog) a una o varias máquinas (cuya finalidad es que tengan todas las conexiones que se produzcan en tiempo real). Disponer de varias máquinas de este tipo es muy útil ya que los hackers suelen correr programas del tipo "zap", que borran las huellas en el sistema asaltado, y este tipo de herramienta (tcp-wrapper) que deja las trazas en un archivo tipo texto, sería muy fácil su modificación editando el archivo con un editor de texto, tal como el "vi" en sistemas UNIX para eliminar las huellas.

De hecho, esto puede constituir una pista de que un sistema ha sido asaltado. Es decir, que los archivos de trazas relativos a una máquina sean distintos en la máquina que lo originan y en las máquinas que lo centralizan. Debemos tener en cuenta que las máquinas que centralizan esta información deben estar muy protegidas ante los posibles ataques.

Para concluir, podemos decir que el tcp-wrappers es una simple pero efectiva herramienta para controlar y monitorear la actividad de la red en nuestra máquina, y nos permite un control sobre las conexiones que se efectúan en nuestra red.

Veamos un pequeño ejemplo del archivo de trazas que genera este software.

```
May 29 13:21:22 lince.uc3m.es in.ftpd[237]: connect from
acme.uc3m.es
May 29 13:52:00 lince.uc3m.es in.ftpd[557]: connect from
acme.uc3m.es
May 29 13:54:21 lince.uc3m.es in.telnetd[561]: connect from
```

```

acme.uc3m.es
May 29 14:50:20 lince.uc3m.es in.ftpd[8228]: refused connect from
acme.uc3m.es
May 29 14:51:12 lince.uc3m.es in.ftpd[8232]: connect from
acme.uc3m.es
May 29 14:57:33 lince.uc3m.es in.ftpd[8275]: connect from
acme.uc3m.es
May 29 15:09:25 lince.uc3m.es in.telnetd[8631]: connect from
acme.uc3m.es

May 29 15:35:34 lince.uc3m.es in.ftpd[8729]: connect from
acme.uc3m.es
May 29 15:43:17 lince.uc3m.es in.ftpd[8754]: connect from
acme.uc3m.es
Jul 25 13:47:45 lince.uc3m.es in.telnetd[338]: refused connect
from acme.uc3m.es Jul 25 13:48:16 lince.uc3m.es
in.telnetd[351]: refused connect from acme.uc3m.es Aug 7
10:20:27 lince.uc3m.es in.telnetd[3457]: connect from
acme.uc3m.es
Sep 22 12:09:29 lince.uc3m.es in.telnetd[19795]: connect from
acme.uc3m.es
Oct 2 09:43:03 lince.uc3m.es in.telnet[10836]:
refused connect from arwen.uc3m.es
Oct 5 07:32:15 lince.uc3m.es in.telnet[2554]: refused connect from
elrond.uc3m.es
Oct 10 17:51:18 lince.uc3m.es in.telnet[6959]:
refused connect from arwen.uc3m.es

```

## 6.2. - NETLOG

Este software de dominio público diseñado por la Universidad de Texas, es una herramienta que genera trazas referentes a servicios basados en IP (TCP, UDP) e ICMP, así como tráfico en la red (los programas pueden ejecutarse en modo promiscuo) que pudiera ser "sospechoso" y que indicara un posible ataque a una máquina (por la naturaleza de ese tráfico).

El paquete está formado por el siguiente conjunto de programas:

### 6.2.1. - Tcplogger

Este programa escucha todos los servicios sobre TCP, dejando una traza de cada servicio en un archivo de trazas, indicando la hora, la máquina origen y el puerto de esa conexión.

Veamos un pequeño ejemplo de un archivo originado por este programa:

```

04/25/95 14:23:50 6C016A00 arapaima.uc3m.es 1153 -> acme telnet
04/25/95 14:29:50 4D444600 elrond 3865 -> acme smtp
04/25/95 14:35:30 39665600 svin09.win.tue.nl ftp-data -> acme 2527
04/25/95 14:53:26 18DB4800 ordago 3268 -> acme smtp
04/25/95 14:54:49 58ECA600 elrond 3880 -> acme smtp

```

04/25/95	14:59:58	399B7801 siuX	1529 -> acme	telnet
04/25/95	15:27:27	4562B200 sun.rediris.es	1617 -> acme	domain
04/25/95	16:15:20	DE069664 pc_jj	1048 -> acme	telnet
04/25/95	16:31:54	35C98800 nazgul	3215 -> acme	domain
04/25/95	16:32:29	D87DE00 sauron	2038 -> acme	domain
04/25/95	16:33:23	F760200 a16-unix	1200 -> acme	domain
04/25/95	16:49:35	707E1A04 info.tamu.edu	ftp-data ->	2536
07/05/95	14:24:18	34874400 elrond	4636 -> acme	smtp
07/05/95	15:02:44	FBB5800 arapaima.uc3m.es	ftp-data -> acme	1545
07/05/95	16:19:48	53C64A00 grande	2271 -> acme	domain
07/05/95	17:36:40	4D09AE00 elrond	1112 -> acme	smtp
07/05/95	17:55:53	2D816A00 chico.rediris.es	4499 -> acme	domain
07/05/95	18:23:50	3DF64E01 selene.uc3m.es	1443 -> acme	domain
07/05/95	18:57:21	A613A00 elrond	1149 -> acme	smtp
09/13/95	14:44:51	4C5FC201 saruman	1023 -> acme	printer
09/13/95	14:45:02	4C60BC01 saruman	1023 -> acme	printer
09/13/95	17:03:02	7603EA00 sauron	1230 -> acme	domain
09/13/95	17:04:37	95720601 fivos	1825 -> acme	domain
09/13/95	17:08:28	18315C00 nazgul	2835 -> acme	domain
09/13/95	17:22:12	786C2000 sauron	1252 -> acme	domain

### 6.2.2. - UDPLOGGER

Es semejante al anterior, pero para los servicios sobre UDP. Un ejemplo del archivo de trazas:

10/23/95	11:25:04	0 d.root-servers.net	domain -> acme	domain
10/23/95	11:25:05	0 elrond	1659 -> acme	domain
10/23/95	11:25:05	0 elrond	1660 -> acme	domain
10/23/95	11:25:05	0 srl.dec.com	domain -> acme	domain
10/23/95	11:25:06	0 acme	4083 -> acme	domain
10/23/95	11:25:06	0 acme	4084 -> acme	domain
10/23/95	11:25:06	0 darkstar.isi.edu	domain -> acme	domain
10/23/95	11:25:06	0 acme	4087 -> acme	domain
10/23/95	11:25:06	0 acme	4088 -> acme	domain
10/23/95	11:25:07	0 slaw.arl.mil	domain -> acme	domain
10/23/95	11:25:09	0 gw.home.vix.com	domain -> acme	domain
10/23/95	11:25:10	0 ns-too.ripe.net	domain -> acme	domain
10/23/95	11:25:10	0 relav.bt.net	domain -> acme	
10/23/95	11:25:32	0 ns.mci.net	domain -> acme	
10/23/95	11:25:32	0 nazgul	1030 -> acme	
10/23/95	11:25:33	0 ordago	3661 -> acme	
10/23/95	11:25:33	0 0.0.0.0	68 -> 255.255.255.255	67
10/23/95	11:25:51	0 zenon	4435 -> acme	
10/23/95	11:25:51	0 zenon	4436 -> acme	
10/23/95	11:25:54	0 vgr.arl.mil	domain -> acme	domain

Los archivos que generan estas dos herramientas pueden ser útiles también para detectar ataques de tipo SATAN o ISS, ya que en los archivos de trazas se aprecian intentos de conexión muy cortos en el tiempo a puertos (tcp o udp) de forma consecutiva.

### 6.2.3. - ICMPLOGGER

Se encarga de trazar el tráfico de icmp. Veamos un ejemplo del archivo de trazas:

10/23/95	11:24:08	0 elrond	-> acme	portunreach
10/23/95	11:25:05	0 ES-	-> acme	hostunreach
10/23/95	11:25:05	0 elrond	-> acme	portunreach
10/23/95	11:25:39	0 elrond	-> acme	portunreach
10/23/95	11:26:25	0 163.117.138.60	-> acme	portunreach

```

10/23/95 11:26:26 0 pc-11-58 -> acme portunreach
10/23/95 11:26:45 0 arpa-gw.hpc.org -> acme hostunreach
10/23/95 11:27:17 0 ES- -> acme hostunreach
10/23/95 11:27:18 0 192.157.65.82 -> acme hostunreach
10/23/95 11:27:41 0 elrond -> acme portunreach
10/23/95 11:28:16 0 elrond -> acme portunreach
10/23/95 11:28:16 0 arpa-gw.hpc.org -> acme hostunreach
10/23/95 11:28:22 0 elrond -> acme portunreach
10/23/95 11:28:26 0 192.157.65.82 -> acme hostunreach
10/23/95 11:28:51 0 192.157.65.82 -> acme hostunreach
10/23/95 11:29:05 0 192.157.65.82 -> acme hostunreach
10/23/95 11:29:46 0 arpa-gw.hpc.org -> acme hostunreach

```

Estos programas pueden guardar su información en ASCII o en formato binario. En este segundo caso, el programa dispone de una herramienta (extract) que permite consultar los archivos de trazas dándole patrones de búsqueda, como puede ser el tráfico desde una red concreta, los intentos de conexión a puertos específicos, etc.

#### 6.2.4. - ETHERSCAN

Es una herramienta que monitorea la red buscando ciertos protocolos con actividad inusual, como puedan ser conexiones tftp - en este caso, si se han realizado con éxito nos indica qué archivos se han llevado -, comandos en el puerto de sendmail (25 tcp) como vrfy, expn, algunos comandos de rpc como rpcinfo, peticiones al servidor de NIS (algunas herramientas utilizan este tipo de servidores para obtener el archivo de password, ej: ypx), peticiones al demonio de mountd, etc. Etherscan se ejecuta en modo promiscuo en la máquina utilizando (al igual que las anteriores) el NIT (Network Interface Tap de SunOs 4.1.x), y también el "Packet Filtering Interface" para realizar esas capturas.

Veamos al igual que en los casos anteriores un ejemplo de archivo de trazas:

```

04/25/95 14:32:29 [rpc] pc_12B14B.uc3m.es.1500 acme RPC lookup for:
pcnfsd
04/25/95 14:32:29 [rpc] pc_12B14B.uc3m.es.1501 acme RPC lookup for:
pcnfsd
04/25/95 16:05:57 [rpc] tony.1500 acme RPC lookup for: ypserv
04/25/95 16:06:01 [rpc] tony.1502 acme RPC lookup for: ypserv
05/05/95 12:26:52 [tftp] router4.61892 acme Attempt to write
`/tftpboot/L1'.
05/05/95 12:26:56 [tftp] router4.61892 acme Attempt to write
`/tftpboot/L1'.
05/08/95 09:50:56 [smtp] arapaima.uc3m.es.1033 acme vrfy jose@acme
05/09/95 17:53:34 [rpc] paco.1501 acme RPC lookup for: pcnfsd
05/10/95 10:38:16 [smtp] saruman.1339 selene.uc3m.es unknown cmd:
hello selene
05/11/95 16:26:00 [rpc] balleste.1500 acme RPC lookup for: pcnfsd
05/11/95 17:30:26 [smtp] bruno.cs.colorado.edu.4671 elrond EXPN rivera
05/11/95 19:58:22 [smtp] mudhoney.micro.umn.edu.1808 elrond EXPN
cloquell
05/17/95 14:47:44 [smtp] elrond.2725 tidos.tid.es vrfy jason

```

```
05/17/95 15:27:31 [tftp] master.etsit.upm.es.1918 elrond Attempt to
read`etc/passwd'.
05/19/95 09:22:17 [rpc] paco.1501 acme RPC lookup for: ypserv
05/19/95 09:32:31 [smtp] tornasol.2748 acme vrfy jose
05/19/95 09:32:53 [smtp] tornasol.2748 acme vrfy jose@di
05/19/95 09:33:10 [smtp] tornasol.2748 acme vrfy jose@kk
05/26/95 09:29:13 [rpc] pc_12B15.uc3m.es.1500 acme RPC lookup for:
pcnfsd
05/26/95 09:29:13 [rpc] pc_12B15.uc3m.es.1501 acme RPC lookup for:
pcnfsd
09/26/95 09:32:23 [smtp] arapaima.uc3m.es.1063 elrond vrfy
postmaster@uc3m.es
09/26/95 10:02:00 [rpc] paco.1500 acme RPC lookup for: ypserv
09/26/95 10:02:00 [rpc] paco.1500 acme RPC lookup for: ypserv
```

### 6.2.5. - NSTAT

Esta herramienta que originariamente fue diseñada para obtener estadísticas de uso de varios protocolos, se puede utilizar para detectar cambios en los patrones de uso de la red, que nos puedan hacer sospechar que algo raro está pasando en la misma.

Esta herramienta viene acompañada por dos utilidades que nos permiten analizar la salida que origina nstat, a saber: nsum, nload. La primera de ellas, nos da información de ciertos periodos de tiempo. La segunda, es un programa awk que produce una salida que puede ser vista de forma gráfica por herramientas como xvgr.

Para concluir este apartado, podemos decir que esta herramienta es muy útil para detectar ciertos tipos de ataques, tal como hemos reflejado anteriormente (con etherscan), así como dar una idea de qué tipo de protocolos están viajando por la red.

Además, tiene la ventaja de que al estar en modo promiscuo, con sólo tenerlo en una máquina del segmento se puede tener monitoreado todo el segmento en el que esté conectado.

### 6.3. - ARGUS

Es una herramienta de dominio público que permite auditar el tráfico IP que se produce en nuestra red, mostrándonos todas las conexiones del tipo indicado que descubre.

Este programa se ejecuta como un demonio, escucha directamente la interfaz de red de la máquina y su salida es mandada bien a un archivo de trazas o a otra máquina para allí ser leída. En la captura de paquetes IP se le puede especificar condiciones de filtrado como protocolos específicos, nombres de máquinas, etc.

A la hora de leer esa información disponemos de una herramienta que incluye el software (llamado ra) y que nos permite también realizar filtros de visualización. Una característica de esta herramienta es la posibilidad de filtrar paquetes de acuerdo a las listas de acceso de los routers CISCO. Es

posible por tanto decirle que nos capture aquellos paquetes que no cumplen las reglas de la lista de acceso definida para esa interfaz del router. Como en el caso anterior (netlog) es posible ejecutar el comando en modo promiscuo (si lo que queremos es auditar todo nuestro segmento). Este programa divide las transacciones en cuatro grupos: TCP, UDP/DNS, MBONE, ICMP.

Algunos ejemplos de captura pueden ser:

```
argus -w NombreArchivoTraza &
```

En este ejemplo le indicamos que nos capture todas las transacciones que se producen en nuestra subred y que lo almacene en un archivo.

```
argus -w ArchivoSalida ip and not icmp &
```

Todo el tráfico ip pero no el icmp.

Como decíamos antes, el ra es el programa para leer la información generada por argus.

Veamos algunos ejemplos de utilización:

```
ra -r ArchivoSalida tcp and host galileo
```

Vemos todo el tráfico tcp (tanto de entrada como salida) en la máquina galileo.

```
ra -C lista_acceso dst net 163.117.1.0
```

Vemos en tiempo real todas las transacciones a la red 163.117.1.0 que violan la lista de acceso de esa interfaz del router.

Observemos, a continuación, un pequeño ejemplo del archivo de trazas generado por esta utilidad:

```

Mon 10/23      Ip   router4      <-  255.255.255.255
Mon 10/23      Ip   router4      <-  255.255.255.255
Mon 10/23      udp  acme.4075    ->  acme.domai TIM
Mon 10/23      udp  acme.4076    ->  acme.domai TIM
Mon 10/23      udp  acme.4079    ->  acme.domai TIM
Mon 10/23      udp  leland.Stanford.domai ->  acme.domai TIM
Mon 10/23      udp  acme.4081    ->  acme.domai TIM
Mon 10/23      udp  acme.4082    ->  acme.domai TIM
Mon 10/23      udp  julieta.2137 ->  acme.domai TIM
Mon 10/23      udp  julieta.2138 ->  acme.domai TIM
Mon 10/23      udp  karenina.2135 ->  acme.domai TIM
Mon 10/23      udp  karenina.2136 ->  acme.domai TIM
Mon 10/23      udp  karenina.2137 ->  acme.domai TIM
Mon 10/23      udp  sparky.arl.mil.domai ->  acme.domai TIM
Mon 10/23      udp  serv2.cl.msu.ed.domai ->  acme.domai TIM
Mon 10/23      udp  wor-srv.wam.umd.domai ->  acme.domai TIM
Mon 10/23      udp  a09-unix.1359 ->  acme.domai TIM
Mon 10/23      udp  admii.arl.mil.domai ->  acme.domai TIM
Mon 10/23      udp  sahara.upf.es.domai ->  acme.domai CON
Mon 10/23      udp  sun.rediris.es.domai ->  acme.domai TIM

```

Para terminar, podemos decir que este software está disponible para SunOs 4.1.x, Solaris 2.3 y SGI IRIX5.2

#### **6.4. - TCPCDUMP**

Es un software de dominio público que imprime las cabeceras de los paquetes que pasan por una interfaz de red. Este programa es posible ejecutarlo en modo promiscuo con lo que tendremos las cabeceras de los paquetes que viajan por la red.

Tanto en la captura como en la visualización de la información, es posible aplicar filtros por protocolo (TCP, UDP, IP, ARP, RARP...), puertos (en este caso el puerto puede ser un número o un nombre especificado en el archivo/etc/services), direcciones fuente, direcciones destino, direcciones de red, así como realizar filtros con operadores (=, <, >, !=, and, not, ...). En la última versión, es posible ver también los paquetes de datos.

#### **6.5. - SATAN (Security Administrator Tool for Analyzing Networks)**

Es un software de dominio público creado por Dan Farmer que chequea máquinas conectadas en red y genera información sobre el tipo de máquina, qué servicios da cada máquina y avisa de algunos fallos de seguridad que tengan dichas máquinas.

Una de las ventajas de SATAN frente a otros paquetes, es que utiliza una interfaz de WWW (como Mosaic, Netscape,...), va creando una base de datos de todas las máquinas chequeadas y las va relacionando entre ellas (de forma que si encuentra una máquina insegura, y chequea otra máquina que está relacionada con ésta, automáticamente esta segunda quedará marcada también como insegura).

Además, tiene la posibilidad de poder chequear las máquinas con tres niveles ("light", normal y "heavy"). Una vez realizado el chequeo de la máquina se genera una salida en formato html, y en el caso de encontrar fallos, da una pequeña explicación sobre el fallo en concreto. Cuando existe algún documento sobre ese fallo recogido en el CERT (advisory) tiene un enlace a ese documento, para que sobre la marcha pueda ser consultado. Asimismo, en el caso de que el fallo de seguridad sea debido a versiones antiguas de software da la posibilidad (mediante un enlace) de instalar una versión nueva de ese software.

Algunos de los servicios chequeados por SATAN son: finger, NFS, NIS, ftp, DNS, rexd, así como tipo de sistema operativo, versión de sendmail, etc. La base de datos generada por SATAN puede ser luego consultada por varios campos: tipo de sistema operativo, tipo de servicio (servidores de NIS, ftp, NFS, X, etc).

SATAN ha sido diseñado como una herramienta de seguridad para ayudar a administradores de sistemas y redes, pero también puede ser utilizada para atacar a sistemas y descubrir la topología de la red de una organización. SATAN es capaz de chequear máquinas por subredes, con lo que quedan al descubierto todas las máquinas que se encuentran conectadas

en dicha subred.

Para poder compilar y ejecutar SATAN basta con poseer la versión 5 de perl y un visualizador de WWW.

Para terminar, algunos de los fallos de seguridad que SATAN es capaz de detectar son:

- Acceso vía rexec
- Vulnerabilidad en el sendmail
- Acceso vía tftp
- Accesos vía rsh
- Acceso a servidores X no restringido
- Exportar sistemas de archivos no restringido
- Acceso a archivos de password vía NIS

## **6.6. - ISS (Internet Security Scanner)**

Es una herramienta de la cual existe versión de dominio público que chequea una serie de servicios para comprobar el nivel de seguridad que tiene esa máquina. ISS es capaz de chequear una dirección IP o un rango de direcciones IP (en este caso se indican dos direcciones IP e ISS chequeará todas las máquinas dentro de ese rango).

El programa viene acompañado de dos utilidades que son ypx y strobe. La primera, nos permite la transferencia de mapas NIS a través de la red y la segunda, chequea y describe todos los puertos TCP que tiene la máquina que chequeamos. Como podemos ver, con la primera herramienta es posible la transferencia de los archivos de "password" en aquellas máquinas que hayan sido configuradas como servidores de NIS.

ISS se puede ejecutar con varias opciones y la salida se deja en un archivo. Además, si ha podido traerse el archivo de "password" de la máquina chequeada, creará un archivo aparte con la dirección IP de la máquina

## **6.7. - Courtney**

Este software de dominio público sirve para identificar la máquina origen que intenta realizar ataques mediante herramientas de tipo SATAN.

El programa es un script perl que trabaja conjuntamente con tcpdump. Courtney recibe entradas desde tcpdump y controla la presencia de peticiones a nuevos servicios del stack TCP/IP (las herramientas de este tipo realizan ataques, chequeando de forma ordenada todos los puertos TCP y UDP que tiene el sistema, para poder ver qué servicios tiene instalados dicha máquina). Si se detecta que se está produciendo un continuo chequeo de estos puertos en un breve intervalo de tiempo, Courtney da un aviso. Este aviso se manda vía syslog.

Courtney puede generar dos tipos de alarmas dependiendo del ataque que se esté produciendo (normal o "heavy", las herramientas como SATAN dispone de distintos grados de chequeo de la máquina).

Esta herramienta necesita el intérprete de PERL y el tcpdump.

### **6.8. - Gabriel**

Software desarrollado por "Los Altos Technologies Inc" que permite detectar "ataques" como los generados por SATAN.

Gabriel identifica el posible ataque y de forma inmediata lo notifica al administrador o responsable de seguridad. La notificación se puede realizar de varias formas (e-mail, cu, archivo de trazas). Este programa existe, en este momento, para SunOs 4.1.x y Solaris, y está formado por un cliente y un servidor. El cliente se instala en cualquier máquina de la red, recoge la información que se está produciendo y la envía al servidor vía syslog. Estos clientes además envían de forma regular información al servidor para indicarle que están en funcionamiento.

En el caso de SunOs 4.1.x (Solaris 1), Gabriel utiliza el programa etherfind para realizar su trabajo. Una característica interesante de este software es que no necesita programas adicionales (como en el caso anterior PERL y tcpdump). El software viene con los ejecutables para SunOs 4.1.x y Solaris (cliente y servidor) así como un programa para realizar un test de funcionamiento.

Veamos un ejemplo de una alerta generada por el programa ante un ataque con SATAN. Además de este archivo, se genera un mensaje de correo alertando del ataque.

```
Mon 07/24/95 14:15:01 restrained attacks
from acme Tue 07/25/95 10:15:01
restrained attacks from acme Tue
07/25/95 14:00:01 restrained attacks from
acme
```

### **6.9. - TCPLIST**

Es un pequeño programa de dominio público que nos informa acerca de todas las conexiones TCP desde o hacia la máquina donde lo estamos ejecutando.

### **6.10. - NOCOL (Network Operations Center On-Line)**

Es un conjunto de programas de monitoreo de sistemas y redes. El software es un conjunto de agentes que recogen información y escriben la salida en un formato que se puede, luego, procesar. Cada dato procesado recibe el nombre de evento y cada evento tiene asociado una gravedad.

Existen cuatro niveles de gravedad: CRITICAL, ERROR, WARNING, INFO. Cada uno de estos niveles es controlado de forma independiente por cada agente. Existe un conjunto de herramientas que nos permite ver toda la información generada por los agentes y que puede

ser filtrada dependiendo de la gravedad del evento.

Entre las cosas que pueden ser controladas por este software tenemos:

- Monitor de ICMP (usando ping o multiping)
- Carga en la red (ancho de banda)
- Monitor de puertos TCP.
- Monitor de SNMP y SNMP traps.
- Monitor de servidor de Nombres.
- Monitor de rpc.
- Chequeo del bootpd

## HERRAMIENTAS QUE CHEQUEAN LA INTEGRIDAD DEL SISTEMA

Veremos, a continuación, una serie de herramientas que nos ayudarán a proteger nuestro sistema. Para conseguirlo, tenemos dos tipos de herramientas. Las primeras, se basan en chequeos a los archivos. Las segundas, nos alertan de posibles modificaciones de archivos y de programas "sospechosos" que puedan estar ejecutándose en la máquina de forma camuflada[9].

Veremos, en primer lugar, las que chequean la integridad de los sistemas de archivos.

### 7.1. - COPS (Computer Oracle and Password System)

Cops es un conjunto de programas diseñado por la Universidad de Purdue que chequea ciertos aspectos del sistema operativo UNIX relacionados con la seguridad.

Existen dos versiones de este paquete: una versión escrita en "sh" y "C" y otra versión escrita en "perl", aunque su funcionalidad es similar. Este programa es fácil de instalar y configurar y se ejecuta en gran cantidad de plataformas UNIX.

En el primer caso, necesitaremos un compilador de lenguaje C y un shell estándar (sh). En el segundo, nos bastará con tener instalado el interprete de perl (versión 3.18 o superior). Entre las funcionalidades que tiene Cops podemos destacar.

- Chequeo de modos y permisos de los archivos, directorios y dispositivos
- Passwords pobres. En el caso que tengamos una herramienta como crack, podemos comentar la línea de chequeo de passwords.
- Chequeo de contenido, formato y seguridad de los archivos de "password" y "group".
- Chequeo de programas con root-SUID.
- Permisos de escritura sobre algunos archivos de usuario como ".profile" y ".cshrc"
- Configuración de ftp "anonymous".
- Chequeo de algunos archivos del sistema como "hosts.equiv", montajes de NFS sin restricciones, "ftusers", etc.

Veamos un ejemplo del archivo creado por este programa:

```
ATTENTION:
Security Report for Tue Apr 11 13:33:33 WET DST 1995
from host acme
Warning! Root does not own the following file(s):
/dev /usr/etc
Warning! NFS file system exported with no restrictions!
Warning! NFS file system exported with
no restrictions! Warning! NFS file system
exported with no restrictions! Warning!
```

```

/dev/fd0 is _World_ writable!
Warning! /dev/fd0 is _World_ readable!
Warning! /etc/ethers is _World_
writable! Warning! /etc/motd is
_World_ writable! Warning!
/etc/utmp is _World_ writable!
Warning! /usr/adm/snm is _World_
writable!
Warning! File /etc/motd (in /etc/rc.local.orig) is _World_ writable!
Warning! User uucp's home directory /var/spool/uucppublic
is mode 03777! Warning! Password file, line 12, user
sysdiag has uid = 0 and is not root
      sysdiag:*:0:1:Old                               System
Diagnostic:/usr/diag/sysdiag:/usr/diag/sysdiag/sysdiag
Warning! Password file, line 13, user sundiag has uid = 0 and is
not root sundiag:*:0:1:System
      Diagnostic:/usr/diag/sundiag:/usr/diag/sundiag/sundiag
Warning! YPassword file, line 2, user sundiag has uid = 0 and is
not root sundiag:*:0:1:System
      Diagnostic:/usr/diag/sundiag:/usr/diag/sundiag/sundiag
Warning! YPassword file, line 3, user sysdiag has uid = 0 and is not
root sysdiag:*:0:1:Old                               System
Diagnostic:/usr/diag/sysdiag:/usr/diag/sysdiag/sysdiag
Warning! /etc/ftpusers should exist!
Warning! Anon-ftp directory pub is World Writable!

```

## 7.2. - Tiger

Es un software desarrollado por la Universidad de Texas que está formado por un conjunto de shell scripts y código C que chequean el sistema para detectar problemas de seguridad de forma parecida a COPS.

Una vez chequeado el sistema, se genera un archivo con toda la información recogida por el programa. Tiger dispone de una herramienta (tigexp) que recibe como parámetro dicho archivo y da una serie de explicaciones adicionales de cada línea que generó el programa anterior. El programa viene con un archivo de configuración donde es posible informarle qué tipo de chequeo se quiere realizar. Podemos comentar las operaciones más lentas y ejecutar éstas de forma menos continuada, mientras que las más rápidas pueden ser ejecutadas más frecuentemente.

Entre la información que chequea el programa tenemos:

- Configuración del sistema.
- Sistemas de archivos.
- Archivos de configuración de usuario.
- Chequeo de caminos de búsqueda.
- Chequeos de cuentas.
- Chequeos de alias.
- Comprueba la configuración de ftp "anonymous".
- Chequeo scripts de cron.

- NFS.
- Chequeo de servicios en el archivo `/etc/inetd.conf`
- Chequeo de algunos archivos de usuario (`.netrc`, `.rhosts`, `.profile`, etc)
- Comprobación archivos binarios (firmas). Para poder chequear éstos es necesario disponer de un archivo de firmas.

### 7.3. - Crack

Este paquete de dominio público realizado por Alex Muffet permite chequear el archivo de contraseñas de UNIX y encontrar passwords triviales o poco seguras.

Para ello, usa el algoritmo de cifrado (DES) utilizado por el sistema UNIX y va comprobando a partir de reglas y de diccionarios las passwords que se encuentran en el archivo de contraseñas, creando un archivo con todos los usuarios y palabras descubiertas. Se realiza una serie de pasadas sobre el archivo de contraseñas, aplicando la secuencia de reglas que se especifique. Estas reglas se encuentran en dos archivos (`gecos.rules` y `dicts.rules`) y pueden ser modificadas utilizando un lenguaje bastante simple. Para una mayor efectividad pueden utilizarse diccionarios complementarios (existen en gran diversidad servidores ftp) en diferentes idiomas y sobre diversos temas.

Experiencias realizadas en la Universidad Carlos III de Madrid sobre diversas máquinas han arrojado resultados de 16% de passwords triviales en máquinas donde no se tenía ninguna norma a la hora de poner contraseñas de usuario.

Es una buena norma pasar de forma periódica el crack para detectar contraseñas poco seguras, además de tener una serie de normas sobre passwords, tanto en su contenido como en la periodicidad con que deben ser cambiadas.

### 7.4. - Tripwire

Este software de dominio público desarrollado por el Departamento de Informática de la Universidad de Purdue, es una herramienta que comprueba la integridad de los sistemas de archivos y ayuda al administrador a monitorizar éstos frente a modificaciones no autorizadas.

Esta herramienta avisa al administrador de cualquier cambio o alteración de archivos en la máquina (incluido binarios). El programa crea una base de datos con un identificador por cada archivo analizado y puede comparar, en cualquier momento, el actual con el registrado en la base de datos, avisando ante cualquier alteración, eliminación o inclusión de un nuevo archivo en el sistema de archivos.

La base de datos está compuesta por una serie de datos como la fecha de la última modificación, propietario, permisos, etc. con todo ello se crea una firma para cada archivo en la base de datos.

Esta herramienta debería ser ejecutada después de la instalación de la máquina con el objeto de tener una "foto" de los sistemas de archivos en ese momento y puede ser actualizada cada vez que añadimos algo nuevo. Dispone de un archivo de configuración que permite decidir qué parte del sistema de archivos va a ser introducida en la base de datos para su posterior comprobación.

## 7.5.- SPAR

Software de dominio público diseñado por CSTC (Computer Security Technology Center) realiza una auditoría de los procesos del sistema, mucho más flexible y potente que el comando `lastcomm` de UNIX.

El programa lee la información recogida por el sistema y puede ser consultada con una gran variedad de filtros como usuario, grupo, dispositivo, admitiendo también operadores (=, >, <, >=, &&...).

Por defecto, el programa obtiene la información del archivo `/var/adm/pacct`. No obstante, se le puede indicar otro archivo. La información puede ser mostrada en ASCII o en binario para su posterior proceso con `spar`.

## 7.6.- lsof (List Open Files)

Este programa de dominio público creado por Vic Abell, nos muestra todos los archivos abiertos por el sistema, entendiendo por archivo abierto: un archivo regular, un directorio, un archivo de bloque, archivo de carácter, un archivo de red (socket, archivo NFS).

El programa admite varios parámetros que nos permiten filtrar información, dependiendo qué tipo de procesos queramos ver en ese instante. Este software está disponible para una gran variedad de plataformas: Aix 3.2.3, HP-UX 7.x y 8.x, IRIX 5.1.1, SunOs 4.1.x, Ultrix 2.2 y 4.2, Solaris 2.3, NetBSD ...

Veamos a continuación un pequeño extracto de una salida de este programa:

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODENAME
init		root	cwd	VDI	7, 0	1536	2 / (/dev/sd0a)
init		root	cwd	VDI	7, 0	1536	2 / (/dev/sd0a)
init		root	txt	VRE	7, 0	65536	1252 / (/dev/sd0a)
	2	root	cwd	VDIR	7, 0	1536	2 / (/dev/sd0a)
portmap	57	root	cwd	VDI	7, 0	1536	2 / (/dev/sd0a)
portmap	57	root	txt	VRE	7, 6	24576	6115 /usr (/dev/sd0g)
vserv	62	root	12u	unix	0xff64ee0c	0x0	>(none)
ypbind	66	root	8uW	VRE	7, 0	14	1279 / (/dev/sd0a)
rpc.yppud	68	root	cwd	VDIR	7, 0	512	3670 /var/yp
rpc.yppud	68	root	txt	VRE	7, 6	16384	6204 /usr (/dev/sd0g)
rpc.yppud	68	root	txt	VRE	7, 6	24576	7334 /usr (/dev/sd0g)
named	83	root	cwd	VDIR	7, 0	512	2484 /etc/namedb
named	83	root	txt	VRE	7, 0	106496	130 / (/dev/sd0a)
named	83	root	7u	inet	0xff64988c	0x0	UDP *:domain
named	83	root	8r	VRE	7, 0	2268	30 / (/dev/sd0a)
named	83	root	9u	inet	0xff64958c	0x0	UDP *:681
syslogd	99	root	22w	VRE	7, 14	102657	73779 /usr/local

syslogd	99	root	23w	VRE	7, 14	102657	73779 /usr/local
				G			(/dev/sd1g)
syslogd	99	root	24w	VREG	7, 14	280321	276653 /usr/local
							(/dev/sd1g)
sendmail	107	root	cwd	VDI	7, 0	51	2468 /var/spool/mqueue
sendmail	107	root	txt	VR	7, 6	172032	7302 /usr (/dev/sd0g)

### 7.7. - CPM (Check Promiscuous Mode)

Este pequeño programa realizado por la Universidad de Carnegie Mellon, chequea la interfaz de red de la máquina descubriendo si está siendo utilizada en modo promiscuo (escuchando todo el tráfico de la red).

Está herramienta es muy útil, porque nos alerta de la posible existencia de un "sniffer" (olfateador) que intente capturar información en nuestra red como puedan ser las passwords. Este programa debería ser ejecutado de forma periódica para detectar lo antes posible el estado promiscuo en la placa de red. Una forma útil de utilizarlo es mandarnos el resultado vía correo electrónico.

Es importante tener en cuenta que muchos de los programas descritos en este documento, pueden poner la placa en modo promiscuo con lo que deberemos asegurarnos que no son nuestros programas los que producen esa alerta. Generalmente los programas tipo "sniffer" suelen estar ejecutándose como procesos camuflados en el sistema.

### 7.8. - IFSTATUS

Software de dominio público creado por Dave Curry, permite, al igual que el anterior, descubrir si un interfaz de red está siendo utilizada en modo promiscuo para capturar información en la red. Sirven todas las recomendaciones mencionadas anteriormente.

Veamos un ejemplo del mensaje que genera ésta aplicación, cuando encuentra una interfaz de red ejecutada en modo promiscuo:

```
Checking interface le0... flags = 0x163
WARNING: ACME INTERFACE le0 IS IN
PROMISCUOUS MODE. Checking interface
le0... flags = 0x49
```

### 7.9. - OSH (Operator Shell)

Creado por Mike Neuman, este software de dominio público es una shell restringida con "setuid root", que permite indicar al administrador mediante un archivo de datos qué comandos puede ejecutar cada usuario.

El archivo de permisos está formado por nombres de usuario y una lista de los comandos que se permite a cada uno de ellos. También es posible especificar comandos comunes a todos ellos. Este shell deja

una auditoría de todos los comandos ejecutados por el usuario, indicando si pudo o no ejecutarlos. Dispone, además, de un editor (vi) restringido.

Este programa es de gran utilidad para aquellas máquinas que dispongan de una gran cantidad de usuarios y no necesiten ejecutar muchos comandos, o para dar privilegios a determinados usuarios “especiales” que tengan algún comando que en circunstancias normales no podrían con un shell normal.

Veamos un ejemplo del logístico creado por el programa:

```
LOGIN: acme ran osh at Wed Jun 7 12:09:09 1995
acme (6/7/95 12:09:11)pwd          -
acme (6/7/95 12:09:13)ls          +
acme (6/7/95 12:09:16)ls -la      +
acme (6/7/95 12:09:20)elm         -
acme (6/7/95 12:09:23)quit        -
acme (6/7/95 12:09:27)exit        -
acme (6/7/95 12:09:30)logout      -
acme (6/7/95 12:09:33)exit        -
logout: acme left osh at Wed Jun 7 12:09:34 1995
```

## 7.10. - NOSHELL

Este programa permite al administrador obtener información adicional sobre intentos de conexión a cuentas canceladas en una máquina.

Para utilizarlo basta sustituir el shell del usuario en el archivo `/etc/passwd` por éste programa. A partir de ahí, cada intento de conexión generará un mensaje (vía e-mail o syslog) indicando: usuario remoto, nombre de la computadora remota, dirección IP, día y hora del intento de login y tty utilizado para la conexión.

Todas estas herramientas se pueden bajar de [lince.uc3m.es](http://lince.uc3m.es) o de cualquier `sunsite`.

## 7.11. - TRINUX

Trinux contiene las últimas versiones de las más populares herramientas de seguridad en redes y es usado para mapear y monitorear redes TCP/IP.

El paquete es muy interesante pues, básicamente, se compone varios discos, con los cuales se bootea la máquina que se va a dedicar a realizar el trabajo y corre enteramente en RAM.

Las aplicaciones que trae, principalmente, son:

- mail -soporte simple de correo saliente usando `smail`.
- netbase - utilitarios estándar de redes, tales como `ifconfig`, `arp`, `ping`, etc.
- netmap - herramientas de escaneo de red, tal como `fyodor's`, `strobe`, `nmap` y `netcat`.
- netmon - herramientas de monitoreo y sniffers, tal como `sniffit`, `tcpdump` y `iptraf`
- perlbase - base del lenguaje Perl.

- perli386 - archivos del sistema Perl.
- perlmods - módulos de Perl.
- pcmcia - soportes de módulos de kernel y scripts para laptop
- snmp - herramientas seleccionadas desde CMU SNMP.
- web - cliente Lynx.
- win32 - herramientas de seguridad para Windows95/NT.

Obtenible en [www.trinux.org](http://www.trinux.org)

## BIBLIOGRAFÍA

1. CHAPMAN, B; ZWICKY, E. (1997). Construya Firewalls para Internet
2. FRISCH, A. (1995) Essential System Administration. O'Reilly & Associates.
3. GARFINKEL, S; SPAFFORD, G. (1995). Practical Unix Security. O'Reilly & Associates.
4. RFC 1244
5. KARANJIT, S. (1996) Internet Firewalls and Network Security. New Riders Publishing MINASU, M; ANDERSON, C; CREEGAN, E. (1996) Windows NT Server 4. Sybex
6. STREBE, M. (1997) NT NETWORK SECURITY. Sybex
7. A. S. Tanenbaum. Sistemas Operativos Distribuidos. Prentice Hall Hispanoamericana, S.A., México, 1996.
8. Magíster David Luis La Red Martínez. Sistemas Operativos. Estudio Sigma S.R.L., Buenos Aires, Argentina, Julio de 2004.
9. **ArCERT** Coordinación de Emergencia en Redes Teleinformáticas de la Administración Pública Argentina. Subsecretaría de Tecnologías Informáticas. Secretaría de la Función Pública

[http://www.arcert.gov.ar/webs/manual/manual\\_de\\_seguridad.pdf](http://www.arcert.gov.ar/webs/manual/manual_de_seguridad.pdf)