Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

TESIS DE LICENCIATURA

ESTUDIO DE CARACTERÍSTICAS BÁSICAS DEL MODELO DE SEGURIDAD DE SISTEMAS OPERATIVOS EN UN AMBIENTE DE REDES DE COMPUTADORAS

Director

Lic. Roberto Bevilacqua

Tesistas

Ma. Eugenia Arroyo
Alejandro D. Limbrunner

Buenos Aires, Argentina
1999

Indice

Consideraciones sobre el trabajo

Organización

El presente trabajo consta de dos tomos correspondientes al estudio en sí mismo y a los apéndices de referencia, respectivamente.

Luego de una breve introducción general, el estudio se organiza en dos partes que exponen los análisis comparativos teórico y práctico de distintos sistemas operativos modernos con respecto a las cuestiones de seguridad contempladas en el proyecto.

La primera parte del trabajo presenta un estudio teórico de dichas cuestiones en los sistemas operativos considerados. Asimismo, se exponen los resultados de la comparación de cada uno de estos aspectos en los distintos sistemas operativos.

De acuerdo a las características propias de estos sistemas operativos, en la segunda parte del trabajo se utilizan herramientas que permiten establecer la comparación de los aspectos de seguridad equiparables.

Por último, en el final del primer tomo se consignan las conclusiones obtenidas en el estudio realizado y las sugerencias de las líneas de trabajo a futuro.

En tanto, los primeros cinco apéndices (A a E) del segundo tomo corresponden a la descripción de la implementación y otras referencias de los aspectos estudiados en cada uno de los sistemas operativos.

El apéndice F expone los resultados obtenidos en los testeos realizados. Estos resultados constituyeron la base del análisis comparativo práctico.

En el anexo G se incluyen algunas notas sobre el nivel de seguridad del software consignando referencias del *Orange Book*, National Computer Security Center (NCSC), Departamento de Defensa de los Estados Unidos.

Sistemas Operativos Estudiados

El criterio de selección de los sistemas operativos se fundamentó en la difusión actual y la disponibilidad de los mismos. De este modo, el estudio se basó sobre las versiones correspondientes a servidores en redes de computadoras de UNIX (Solaris y Linux) y Windows NT.

El mayor inconveniente en el caso de UNIX es que existen más de cien versiones en vigencia. Sin embargo, sólo un pequeño porcentaje de ellas comprende la mayor parte del parque informático dedicado a este sistema operativo. Dicho conjunto reducido puede dividirse en dos grupos según la naturaleza libre o comercial de dichas versiones.

Con la intención de cubrir ambos enfoques se optó por trabajar con Solaris, del campo comercial, y Linux, como versión libre. En el primer caso, se utilizó la versión Intel 2.6 con los parches recomendados al 1º de enero de 1999. En cuanto a Linux, se contó con la distribución Slackware 3.6 con la actualización del kernel y la aplicación de los parches pertinentes.

Windows NT, en cambio, no presenta ningún tipo de problemática ya que existe un único proveedor. Se utilizó la versión 4.0 con Service Pack 4 y los parches correspondientes al 1º de enero de 1999.

Introducción

Un poco de historia

Los sistemas de información han evolucionado en las últimas décadas desde sistemas basados en grandes servidores centralizados y aislados, a sistemas corporativos de computación descentralizados. En estos últimos tanto los datos como los equipos están dispersos por toda la organización, características que, en principio, los tornan más inseguros que los primeros.

En los años ochenta, el advenimiento de la computadora personal permitió la masificación del uso de computadoras y, por ende, del auge de los usuarios. Hizo posible almacenar enormes cantidades de información en los equipos distribuidos en vez de en computadoras centralizadas. Como consecuencia también se agudizó la problemática de la protección de la información, la seguridad y el robo de datos y de escuchas. [She97]

La situación se hizo más sensible con el surgimiento de las redes locales y la interconexión de las mismas en redes institucionales. A los problemas de seguridad tradicionales se sumaron otras brechas propias de este cambio tecnológico. Estos riesgos adquirieron ribetes particulares con la aparición de las figuras de usuarios remotos, portátiles, y la popularización del uso de redes al gran público, como Internet. [She97]

Bajo estas condiciones, resulta imprescindible maximizar el sistema de seguridad en todos los niveles. Una cuestión sumamente relevante es la seguridad de los sistemas operativos. Sin embargo, no existe un sistema operativo que pueda cubrir todos los riesgos de seguridad [She97]. La solución involucra otros aspectos que comprenden puntos muy variados como seguridad física, lógica, políticas bien diseñadas, educación y cooperación de los usuarios, entre otros.

Aunque la seguridad de los servidores puede ser administrada desde una posición central, la seguridad de los usuarios es difícil de gestionar. Con frecuencia los usuarios administran su propia seguridad. Dado que la información se distribuye por la organización, resulta más vulnerable al ser expuesta en las localizaciones de los equipos distribuidos.

Las amenazas a los sistemas de información son tanto naturales como intencionadas. Los datos de dichos sistemas son vulnerables ya sea a desastres naturales (inundaciones, incendios, inconvenientes con la alimentación eléctrica, fallas de equipos, etc.), como a la corrupción generada por gente maliciosa y a las acciones con consecuencias dañinas debidas al desconocimiento o a la ignorancia.

La naturaleza de los riesgos para la información originados por seres humanos puede ser premeditada o accidental. Las acciones intencionales se clasifican por el tipo de ataque en pasivas y activas. El primero implica únicamente la recopilación de información tales como escuchas (sniffing) o pinchazos. El último caso involucra la modificación de los datos almacenados o transmitidos, proceso que puede ser disfrazado bajo la figura de un accidente. [Gar96][She97]

La fuente de los ataques puede ser interna o externa. En principio la idea sería establecer una fortaleza inexpugnable para el amenazante mundo exterior. Sin embargo, los sondeos revelan que la mayoría de los expertos de seguridad consideran que los propios empleados constituyen la mayor amenaza para los sistemas de información. [Gar96]

De hecho, las estimaciones obtenidas por dichas encuestas indican que más del 70% de los ataques perpetrados sobre las respectivas organizaciones fueron provocados desde el interior de las mismas. [Network World Magazine - www.nwfusion.com] Los empleados están

familiarizados con la red, conocen la información vital a la que pueden acceder mediante sus cuentas o las de aquellos que estén autorizados. Existen otras fuentes de riesgo potenciales tales como socios temporales o cualquier tipo de intercambio de datos con otras instituciones. [She97]

Este tipo de situaciones ya plantea necesidades que van más allá del modelo de seguridad de los sistemas operativos mismos e involucran el establecimiento de políticas estrictas de seguridad, la concientización y el entrenamiento riguroso no sólo de los responsables del área informática sino de todo el plantel directivo. La tarea del administrador de sistemas consiste en estar actualizado en las cuestiones candentes y volcar esta información para generar un entorno.

Definiciones

Los términos *seguridad*, *protección*, *privacidad*, y demás conceptos relacionados, pueden tener múltiples interpretaciones. Por tal motivo, resulta conveniente precisar su significado en el contexto de los sistemas informáticos.

"La seguridad de la información consiste en la práctica de proteger los recursos y los datos de un sistema de computadoras y redes, incluyendo la información guardada en dispositivos de almacenamiento y en su transmisión." [She97]

Esta definición implica que los bienes de la tecnología informática que requieren protección comprenden tanto a las computadoras y las redes, sus componentes de hardware y software, así como también la información almacenada, procesada o transmitida mediante dichos sistemas.

El propósito de la seguridad informática apunta a proveer un amplio espectro de protección para la organización, proyectado en forma de políticas y procedimientos así como en la aplicación de las medidas tecnológicas que permitan asegurar todos los recursos informáticos de la misma.

De este modo, la seguridad de un sistema de computadoras también incluye encontrar soluciones técnicas a problemas que no son técnicos. Siempre que se desee hacer más seguro un sistema, resulta imprescindible realizar un análisis detallado de lo que se desea proteger, establecer los recursos humanos y económicos que se está dispuesto a gastar para lograrlo y, finalmente, definir políticas, normas y procedimientos para aseverar que la seguridad del sistema no sea comprometida. Asimismo, es necesario llevar un control de auditoría del sistema en cuestión. [Gar96]

Por otra parte, en su amplia definición existen muchas formas de seguridad que se deben contemplar. Aunque todos estos aspectos son relevantes, las características del entorno de trabajo determinan la importancia relativa de cada uno de ellos dentro del contexto en cuestión. A continuación se definen algunos de estos tipos [Gar96]:

- Confidencialidad. Proteger la información de ser leída o copiada por un individuo que no ha sido expresamente autorizado a hacerlo por el dueño de la misma. Esto se refiere no sólo a custodiar la totalidad de la información sino también a proteger pequeñas porciones de la misma que puedan ser utilizadas para inferir datos.

- Integridad. Proteger la información (incluyendo los programas) de ser borrados o alterados de cualquier manera sin el permiso del dueño de dicha información. Esto incluye información de auditoría, copias de resguardo, alteración de fechas de creación y modificación de archivos y documentación.

- Disponibilidad. Proteger los servicios de forma que no sean degradados o inhabilitados sin autorización. Este aspecto apunta a asegurar que en el momento que un usuario autorizado necesite acceder al servidor, pueda hacerlo sin inconvenientes.

- Consistencia. Asegurar que el sistema se comporta como los usuarios autorizados lo esperan. Está relacionado con asegurar la correctitud de los datos y los programas en uso.

- Control. Regular el acceso al sistema y el uso que se hace del mismo. Controlar la presencia de usuarios no autorizados y alteraciones en el sistema.

- Auditoría. Llevar un registro de las acciones y modificaciones realizadas en el sistema por los administradores o los usuarios. Ante el surgimiento de un evento de seguridad es imperativo conocer como se produjo para evitar que se repita.

Actualmente existe una tendencia a no utilizar los términos "seguro" o "inseguro" para referirse a un sistema de computación, sino adoptar la idea de niveles de confianza, recalcando de esta forma la imposibilidad de alcanzar la seguridad absoluta. La única manera de alcanzar un sistema seguro es generando un sistema que sea suficientemente confiable. [Gar96]

Si bien los conceptos *seguridad* y *protección* pueden ser utilizados como sinónimos, se especificará el significado en este contexto de modo de evitar confusiones.

Tal como se planteó, *seguridad* se refiere a la problemática en su totalidad, con las múltiples facetas implicadas. En tanto, *protección* se relaciona con los mecanismos específicos apreciados por un sistema operativo con el fin de salvaguardar los recursos del equipo. [Tan92]

El concepto de *privacidad* se presta a confusión con la *confidencialidad* ya nombrada. Si bien la idea es similar, el primero es más amplio y abarca no sólo la confidencialidad de la información almacenada en los sistemas de cómputos, sino también la de los datos que pudieran estar fuera de los mismos.

Análisis de Riesgos

El primer paso para mejorar la seguridad de un sistema es poner en claro estos tres puntos [Gar96]:

- Objeto que se trata de proteger.
- De quién se trata de proteger.
- Cuánto tiempo, esfuerzo y dinero se está dispuesto a gastar para obtener una protección adecuada.

Un análisis de riesgos simple debe constar de tres tareas básicas:

- Identificar las áreas a proteger.
- Determinar los peligros.
- Calcular los riesgos.

Estas tareas pueden ser demasiado para un hogar o para una empresa pequeña y demasiado pobres para una universidad o una gran corporación.

Si bien hay muchas maneras de realizar este análisis, se recomienda hacer un listado de los ítems a proteger. Es necesario para ello conocer la instalación, los equipos involucrados, los sistemas operativos en uso, las leyes locales y las condiciones de contratación de seguros.

Algunos de los ítems a proteger son tangibles (computadoras, copias de resguardo, manuales, datos, registros de auditoría) y otros son intangibles (privacidad de los usuarios, imagen y reputación pública, capacidad de operación). Posiblemente no todos sean de igual interés y, además, factibles de proteger mediante una política de seguridad informática.

También resulta necesario determinar los peligros. Algunos de estos pueden ser naturales (terremotos, inundaciones, epidemias). Otros pueden tener que ver con el personal involucrado (renuncia o enfermedad prolongada de individuos clave, daños provocados por empleados o ex-empleados) o de servicios (corte de energía, de vínculos de comunicación). Finalmente, otros pueden estar relacionados con la pérdida o daño provocado a la información (robo de equipos, errores en los programas, introducción de un virus).

Por supuesto es necesario prever la probabilidad que estos peligros ocurran. En una ciudad como Buenos Aires la probabilidad de un terremoto es casi nula, pero no debe subestimarse la posibilidad de una inundación o corte de energía.

Luego se debe realizar un análisis del costo provocado por cada uno de los peligros previamente determinados. En el mismo se debe incluir el costo de reposición de bienes, de imposibilidad de operación o de pérdida de personal. Finalmente se debe calcular el costo de la prevención de cada uno de los peligros.

Políticas de Seguridad Informática

Una política de seguridad define los bienes que se consideran importantes y especifica los pasos a tener en cuenta para salvaguardar dichos bienes. Hay diferentes tipos de políticas, desde una guía de unas pocas hojas en papel, hasta políticas separadas para cada uno de los servicios de información en uso.

Una política de seguridad de información debe en principio, dejar en claro qué se protege y por qué, en segundo lugar especificar los roles y responsabilidades relacionados y, por último proveer una guía para interpretar y resolver los conflictos que pudieran ocurrir. [Gar96]

En muchas ocasiones es necesaria la formulación de estándares de seguridad a los que deben adherir todos los miembros de la organización. En general, los encargados de seguridad de los sistemas, además de aplicar los aspectos tecnológicos que reflejen estas normas, son responsables de comprobar y exigir el cumplimiento de las políticas en cuestión. También es posible desarrollar guías con recomendaciones y sugerencias a seguir, pero que puedan dejarse de lado de ser necesario.

Seguridad Física

Un tema sobre el que no se trata habitualmente es la seguridad física de las computadoras y demás equipos que forman parte de los sistemas. Los daños ocasionados por un incendio, un robo o un terremoto pueden ser de igual o mayor importancia que los provocados por un terrorista informático que logra ingresar a los sistemas. [Gar96]

Un problema grande al tratar de definir los posibles peligros y las políticas para enfrentarlos, es que las necesidades de cada instalación son extremadamente disímiles. Las zonas geográficas susceptibles a terremotos o inundaciones tienen políticas de seguridad física radicalmente distintas a otras zonas sin estos problemas.

Otro problema en esta definición es que este tipo de seguridad no puede ser incluida como parte del sistema operativo, aunque el mismo puede proveer herramientas para planificar situaciones de contingencia generales.

Los peligros físicos que atañen a la seguridad de los sistemas de computadoras pueden provenir de distintas fuentes, ambientales, accidentales y de acceso, vandalismo, actos de guerra, terrorismo o robo [Gar96]. Algunos ejemplos de estos peligros se exponen al final de la sección.

También puede ser necesario considerar el impacto de la perdida de las comunicaciones telefónicas o de red, el cierre de empresas proveedoras, la falta de personal masiva debido a una epidemia y la muerte o la incapacidad de personal crítico.

En la mayor parte de los casos basta tener un buen esquema de copias de resguardo para que los daños sean mínimos. Las reproducciones deben ser conservadas en un ambiente seguro y alejado de la instalación principal. Además, las copias de resguardo deben ser guardadas en cajas de seguridad y transportados por personal autorizado. El robo de las cintas de resguardo es tan grave como el de las computadoras que contienen la información resguardada.

Sitios que son susceptibles a inundaciones, terremotos u otros desastres naturales, así como blancos militares o terroristas, posiblemente consideren tener instalaciones en otra ciudad conectada por un medio de alta velocidad y mantener la información en forma replicada en ambos sitios. En el caso de las personas involucradas, siempre es necesario que haya más de un individuo realizando las tareas críticas.

Algunos Peligros Físicos

- Fuego. Las computadoras no resisten el fuego y en caso de incendio, muchas veces el agua utilizada para apagar el fuego causa tanto o más daño que el fuego mismo. El uso de gases como halón o dióxido de carbono para extinguir el fuego en centros de cómputos tiene la desventaja que es asfixiante y, además, ayuda a promover el efecto invernadero.
- Humo. El humo es un abrasivo potente y suele dañar las cabezas de los discos. Incluso el humo de cigarrillos es dañino.
- Polvo. El polvo es otro agente que afecta los discos. Además, muchas veces el polvo es conductor eléctrico por lo que puede provocar cortocircuitos si se acumula en gran cantidad.
- Terremotos. Mientras que algunas construcciones pueden ser destruidas, muchas seguirán en pie. Es importante asegurar que las computadoras ubicadas en sitios donde pueden ocurrir terremotos, no estén colocadas en lugares donde puedan caer o moverse.
- Explosiones. Es posible que una explosión afecte al edificio. En este caso es importante minimizar las vibraciones a las que se someten los equipos de computadoras.
- Temperaturas extremas. Las computadoras funcionan bien entre el rango 10- 32ºC y pueden funcionar erróneamente o incluso dañarse a temperaturas muy bajas o muy elevadas.
- Insectos y otros animales. A veces los insectos ingresan al interior de la computadora y pueden causar daños a la fuente de poder. Las telas de araña juntan polvo y las ratas y otros roedores pueden deteriorar cables de conexión.

- Ruido eléctrico. Motores, ventiladores, equipos de aire acondicionado u otras computadoras pueden generar ruido eléctrico que puede causar problemas intermitentes con ciertos sistemas. Es importante mantener las instalaciones aisladas con filtros eléctricos y conectadas a tierra.
- Rayos y relámpagos. Un rayo que golpea un edificio puede dañar equipos conectados a la red eléctrica y genera un gran campo magnético que puede alterar información almacenada en medios magnéticos. Es importante desenchufar todo equipo que no sea crítico durante una tormenta eléctrica.
- Vibraciones. Las vibraciones ambientales, provocadas por un tren o por caminar sobre el piso elevado de un centro de cómputos puede provocar que los circuitos impresos se salgan de sus lugares o que las cabezas de un disco pierdan su alineación.
- Humedad. Si bien la humedad previene los problemas ocasionados por las descargas de electricidad estática, también puede producir condensación en los circuitos y provocar cortocircuitos.
- Agua. El agua puede destruir una computadora. El riesgo primario es un cortocircuito. El agua suele provenir de lluvias o inundaciones pero puede igualmente salir de un extintor de incendios fallado o un toilette.
- Comidas y bebidas. Muchos equipos de computadoras son dañados por bebidas o comidas que caen dentro del mismo. Además, una comida puede engrasar los dedos de una persona que puede arruinar cintas y otros medios magnéticos al tocarlos.
- Control de acceso. Se debe asegurar que no es posible la entrada a lugares que contengan servidores o información sensitiva a través de conductos de aire o pisos elevados. Las paredes de vidrio son fáciles de romper y, además, se puede obtener información valiosa simplemente mirando a través de ellas.
- Actos de vandalismo. Los mismos pueden ser provocados por un empleado disconforme o por una revuelta popular. Por los agujeros de ventilación de los equipos se pueden ingresar elementos líquidos o metálicos que pueden causar gran daño. Los cables y conexiones de red, especialmente si están en el exterior del edificio, pueden ser fácilmente objeto de vándalos.
- Actos de guerra o terrorismo. Si la instalación puede ser un blanco militar o se encuentra en una región de turbulencia política, es posible requerir de una mayor protección estructural de las instalaciones.
- Robo. Debido a que algunas computadoras y partes de las mismas son pequeñas, las mismas son fácilmente objeto de robos. En caso en que la computadora sea robada, es importante que la información se encuentre cifrada para que permanezca inaccesible a los ladrones.

PARTE I

Análisis Teórico

Modelo de Seguridad

I. Sistemas Operativos

Una computadora, mediante el software que tiene instalado, ofrece la posibilidad de almacenar, procesar y recuperar información, entre otras múltiples posibles actividades. Este software puede clasificarse en dos categorías básicas: los programas del sistema, que administran la operación del equipo mismo, y las aplicaciones, que resuelven los problemas de los usuarios.

El *sistema operativo* es el programa del sistema más importante ya que controla los recursos de la computadora y brinda la base sobre la que se desarrollan las aplicaciones.

Las computadoras constan de una amplia variedad de dispositivos, tales como procesadores, memoria, discos, terminales, adaptadores de red, impresoras, etc. El sistema operativo es el responsable de administrar, controlada y ordenadamente, dichos dispositivos entre aquellos programas que compiten por estos recursos.

Asimismo, desde el punto de vista del usuario, el sistema operativo representa una máquina virtual o extendida que facilita el desarrollo de programas de aplicación sobre el hardware subyacente.

La estructura de los sistemas operativos puede ser dividida en tres partes básicas [Gar96]:

- Núcleo (kernel). El núcleo del sistema se carga al momento de encender la máquina y es el responsable de proveer una capa de abstracción entre el hardware y los programas de usuario. Además, se incluyen en este nivel, aunque algunas implementaciones de sistemas operativos los separen, a los módulos de administración de memoria, de procesos y de archivos que son los básicos en todos los sistemas operativos.

- Programas Utilitarios. Estos programas estándar permiten acceder a funciones del sistema operativo. Suelen servir a un conjunto de funciones aunque hay algunos que son más generales.

- Bases de Datos. El sistema almacena la información de configuración en archivos con diversos formatos Dichos datos son utilizados por una gran variedad de programas del sistema.

Para conocer detalles específicos de la arquitectura de los sistemas operativos considerados en el presente trabajo, consultar el Apéndice A.

II. Seguridad del Sistema Operativo

Dado el papel preponderante del sistema operativo en el funcionamiento de un equipo, la estrategia de seguridad implementada por el mismo es crucial en la definición de la seguridad del sistema como un todo.

Si bien los *firewalls* (protectores de entrada a redes), los dispositivos de cifrado y otros componentes discretos juegan un rol vital en la infraestructura de seguridad, un punto crítico es la elección del sistema operativo de los servidores. [Mic98]

En un sistema operativo multitarea, los programas comparten una gama de recursos del sistema incluyendo la memoria del equipo, dispositivos de E/S, archivos y procesadores. Un sistema operativo confiable debe asegurar que las aplicaciones no accedan a dichos recursos sin la autorización apropiada o monopolicen alguno de ellos a expensas de otras aplicaciones. (esto es protección)

Esta cuestión se torna aún más crítica en servidores de archivos o bases de datos para proteger esta información de cualquier acceso indebido a través de la red.

En ambientes de red con múltiples usuarios con acceso a los mismos recursos físicos, resulta vital controlar dicho acceso y evitar acciones prohibidas. Los sistemas operativos y los usuarios deben ser capaces de asegurar archivos, memoria y configuraciones de visualizaciones o modificaciones no autorizadas.

El objetivo del sistema de seguridad es proteger todos los componentes del sistema, incluyendo hardware, software y datos almacenados en el sistema. La seguridad de los sistemas operativos ofrece una línea inicial de defensa fundamentada en el enfoque de cuentas y claves.

Sin embargo, el modelo de seguridad de los sistemas operativos también comprende mecanismos no tan obvios para proteger a los usuarios y al mismo sistema operativo de acciones indebidas, ya sean intencionales o accidentales. [Rus98c]

Es importante destacar que las características relativas a la seguridad deben ser abordadas en el diseño del sistema operativo. Toda cuestión que no sea contemplada como parte del sistema operativo mismo constituye una vulnerabilidad intrínseca del mismo. [Lew98]

Cabe reiterar que el espectro de seguridad comprende otras facetas que no pueden ser cubiertas por el sistema operativo por sí solo. Es necesario considerar la cuestión de la seguridad en todos los niveles del sistema, desde el aspecto físico hasta la capa de aplicación.

Niveles de Seguridad

El National Computer Security Center (NCSC) se fundó a los fines de ayudar a los usuarios a proteger la información de su propiedad. El primer objetivo del NCSC apuntó a la generación de un documento que contuviese estándares técnicos y criterios para ser utilizados en la evaluación de sistemas de computadoras. El NCSC también estableció un proceso a través del cual los fabricantes de software pueden someter a prueba y calificación sus productos con respecto a la cuestión de seguridad.

El proceso de calificación del software es sumamente extenso. Se basa en un conjunto de normas asentadas en el *Orange Book* de la NCSC. Los documentos adicionales que cubren este proceso se denominan colectivamente como *Rainbow Series*. En particular, la publicación *Trusted Network Interpretation*, también conocida como *Red Book*, constituye la interpretación del *Orange Book* para la aplicación de esta calificación a las cuestiones relacionadas con ambientes de red.

Las calificaciones de seguridad de NCSC contemplan un rango de cuatro divisiones entre *A*, nivel de máxima seguridad, y *D*, grado más bajo de la escala. Cada una de estas divisiones refleja el grado de confianza que el sistema brinda para proteger información clasificada o sensible.

A continuación, se delinean brevemente los puntos más importantes requeridos por cada uno de los niveles según las definiciones del *Orange Book*:

- División D - Mínima Protección. Esta división se reserva para aquellos sistemas evaluados que han fallado en cumplir los requisitos de un nivel más elevado.

- División C - Protección Discrecional. En líneas generales, este nivel requiere control de acceso discrecional y ciertas capacidades de auditoría sobre los usuarios y las acciones de los mismos.

- División B - Protección Obligatoria. Uno de los puntos fundamentales de esta división apunta a evaluar los mecanismos provistos por el sistema para preservar la integridad de la información sensible y su utilización para implementar un conjunto de reglas de control de acceso obligatorio.

- División A - Protección Verificada. Este nivel se caracteriza por la utilización de métodos formales de verificación de seguridad, de modo de asegurar que los controles de seguridad obligatorios y discrecionales implementados en el sistema pueden proteger efectivamente información clasificada o cualquier otros datos almacenados o procesados por el sistema. Se necesita una extensa documentación del sistema para demostrar que el mismo cumple los requisitos de seguridad en todos los aspectos del diseño, el desarrollo y la implementación.

Además, las bandas de calificación *B* y *C* se organizan en una jerarquía de clases que se diferencian por los mecanismos de protección que cada una de ellas requiere. Por ejemplo, dentro de la división *C* el software puede calificarse como *C1* o *C2*, siendo esta última la clase de mayor seguridad.

El Apéndice G incluye algunas referencias del *Orange Book* que explican las características de cada una de las clases y detallan los requisitos específicos para cumplimentar cada nivel.

En el caso de los sistemas operativos, los cuestiones más importantes para lograr la calificación *C2* son: acceso y control discrecional; identificación y autenticación; auditoría; reutilización de objetos [Rus98c]. Un sistema operativo debe ser capaz de definir y controlar el acceso de sus usuarios, proporcionar una vía por la cual los usuarios puedan identificarse a sí mismos de manera unívoca, ofrecer facilidades para realizar auditorías de los sucesos y las acciones relativas a la seguridad y evitar que un proceso pueda acceder a información de otro proceso.

También es posible que un sistema cumpla sólo algunos requisitos de un determinado nivel superior obteniendo la calificación correspondiente a esta funcionalidad.

III. Implementación del Modelo de Seguridad

a. Características Comunes

La estrategia de seguridad de los sistemas operativos apunta a administrar los recursos compartidos de modo que estrictamente las personas adecuadas tengan acceso a los datos y dispositivos según se les autorice en el momento que corresponda.

Esta tarea se traduce en un conjunto de servicios de seguridad que aplican los controles sobre quién puede ver, modificar o eliminar información. Asimismo, existen un número de servicios adicionales que facilitan la implementación y la administración de dichos servicios de seguridad.

Windows NT y UNIX contemplan estas características de seguridad en las especificaciones de diseño de los sistemas operativos.

Como la mayoría de los sistemas operativos en ambientes de red, el modelo de seguridad se apoya en una estructura integrada por cuentas de usuario, grupos, permisos y derechos. Todo usuario que accede al servidor debe identificarse mediante el nombre unívoco que tiene asignado en el momento de iniciar la sesión de trabajo.

Cabe aclarar que la administración de usuarios y los conceptos afines de los sistemas operativos considerados en el presente trabajo se explican en el Apéndice B.

En principio, el proceso de autenticación se basa en el secreto compartido entre el usuario y el servidor en cuestión, es decir, la palabra clave del usuario. Sin embargo, es posible incorporar otros medios de validación.

El grupo reúne a un grupo de usuarios según algún criterio particular. Todo usuario puede pertenecer a uno o más grupos. Generalmente, la implementación de este concepto facilita la administración de permisos y derechos.

Los permisos reflejan el nivel de acceso que los distintos usuarios y grupos poseen sobre los recursos. Por otra parte, los derechos indican las acciones que los usuarios o grupos pueden ejecutar.

b. Particularidades de Windows NT

El modelo de seguridad comprende elementos para controlar quién accede a los recursos y en qué condiciones, y para proyectar un plan de auditoría de sucesos. Este esquema de seguridad se aplica tanto a Windows NT Server como Workstation, con excepción de la base de cuentas de usuario que, en el primer caso puede abarcar a un dominio entero, mientras que en el último es únicamente local. (ver definiciones en Apéndice B.1)

La seguridad de Windows NT puede ser bastante complicada, especialmente cuando se refiere al sistema operativo. A pesar de su complejidad, apunta a cubrir dos tareas simples:

- Restringir el acceso a objetos tales como recursos del sistema, archivos y dispositivos.
- Proporcionar los servicios de auditoría que permitan la operación de seguimiento sobre cada objeto.

En estos conceptos se fundamenta la seguridad de Windows NT. La idea es verificar que el usuario es aquel que dice ser en el inicio de sesión de trabajo y, posteriormente, autorizarle el acceso a los recursos apropiados.

Windows NT es un sistema operativo orientado a objetos y su seguridad se construye desde el nivel más bajo de la estructura de objetos. Esta característica determina que sea mucho más fácil de proteger que otros sistemas operativos. [Mic96]

Con el fin de lograr un sistema operativo seguro, resulta indispensable contar con la capacidad de controlar el acceso a recursos tales como archivos, memoria y dispositivos. Los diseñadores de Windows NT abordaron esta cuestión centralizando el acceso a estos recursos mediante la representación de los mismos como objetos. Asimismo, otro tipo de recursos más abstractos (procesos, *threads*, ventanas y demás) también se visualizan como objetos. [Mic96]

Un *objeto* constituye un concepto de programación bajo el cual los datos y las funciones necesarias para manipular dichos datos están combinados en una estructura de programación. Unicamente las funciones incluidas en el objeto pueden operar en forma directa sobre los datos. Cualquier otra referencia debe invocar a estas funciones.

Dado que todos los recursos en Windows NT están representados internamente como objetos, es posible implementar un único mecanismo de seguridad para controlar el acceso a todos los objetos de Windows NT. La utilización de objetos y un mecanismo común de seguridad contribuye a definir un esquema de seguridad más robusto y consistente que el uso de distintos mecanismos específicos para cada recurso. [She97]

El concepto de objetos es vital en la concepción de la seguridad en Windows NT, ya que del único modo en que los programas pueden acceder a un objeto Windows NT, y en consecuencia al recurso asociado con el objeto, es solicitando al mismo sistema operativo que realice la operación sobre el objeto por ellos.

Los programas no operan sobre los objetos en forma directa. Solamente el sistema operativo cuenta con esta capacidad. Esta propiedad es fundamental para considerar a Windows NT como seguro: resulta relativamente fácil verificar el acceso a cada objeto para comprobar si el programa invocante está autorizado a ejecutar su solicitud.

De este modo, bajo Windows NT la seguridad de los recursos redunda realmente en la seguridad de los objetos, que involucra las siguientes tareas [She97]:

- **Control de Acceso.** El primer requerimiento de seguridad consiste en controlar todos los accesos a todos los objetos de modo de evitar usos indebidos. Esta característica de Windows NT fortalece la capacidad del sistema operativo de impedir que las aplicaciones obtengan accesos no autorizados sobre recursos de otras aplicaciones o del sistema operativo con o sin intención. Este control se implementa de los siguientes modos:
  - Windows NT controla el acceso a los objetos definiendo listas de control de acceso (LCA) asociadas a cada uno de ellos. Un usuario que puede acceder a un objeto tiene *permisos* sobre ese objeto.
  - Windows NT controla las acciones que los usuarios pueden ejecutar mientras trabajan en el sistema mediante el establecimiento de *derechos*.

  Así, estos controles habilitan a los administradores para determinar específicamente quiénes pueden trabajar, qué pueden hacer y dónde pueden hacerlo. También permiten que el sistema operativo proteja los objetos de accesos al azar o maliciosos.

- **Auditoría.** Debido a que Windows NT utiliza un único mecanismo para controlar el acceso a objetos, puede llevar un registro de las acciones exitosas y fallidas de los usuarios. El administrador del sistema determina qué objetos y acciones rastrear. Estos registros de auditoría son sumamente útiles a la hora de analizar problemas de seguridad.

- **Monopolización de Objetos.** Bajo Windows NT cada usuario tiene cuotas de uso de recursos tales como memoria, espacio en disco, tiempo de uso de procesador y demás. Esta característica impide la monopolización de algún recurso por parte de un usuario. La implementación del concepto de cuotas de uso de recursos en Windows NT se basa en el mismo esquema de seguridad utilizado para el control de acceso y la auditoría de objetos. Cada objeto cuenta con cargas del recurso asociadas con él, las cuales son contrastadas con los umbrales del usuario que intenta acceder al objeto. Windows NT se asegura que el usuario no supere el límite permitido.

Más allá de las cuestiones de diseño del sistema de seguridad, cabe destacar que el sistema operativo Windows NT aprovecha los mecanismos de seguridad provistos por los microprocesadores modernos en cuanto al manejo de la memoria y al acceso a los dispositivos.

Windows NT ejecuta en memoria protegida. Sólo el sistema operativo tiene acceso al código y a los datos de Windows NT. Cualquier intento de acceder a la memoria del sistema operativo es rechazado y el proceso involucrado es eliminado. Sin embargo, Windows NT cuenta con la posibilidad de acceder a toda la memoria del sistema, incluyendo la correspondiente a otros programas. [Rus98a]

Windows NT suministra una porción de memoria de uso exclusivo para cada proceso. De este modo, un programa sólo está autorizado a trabajar en su espacio de memoria y cualquier intento de acceso no autorizado a la memoria asignada a otros procesos es denegado.

Por último, ningún programa puede acceder directamente a un dispositivo. Toda petición es remitida a Windows NT que se encarga de administrar el dispositivo y de verificar la validez de la solicitud.

Arquitectura

El acceso a los objetos es únicamente provisto a usuarios autorizados por lo cual deben identificarse en el proceso de registro al iniciar la sesión de trabajo (consultar Apéndice B.1). Cada usuario recibe un número de identificación único y un *token* que se renueva en cada sesión para prevenir posibles ataques de *hackers*. Mediante este *token* es posible rastrear las actividades a lo largo de una sesión y registrarlas en archivos de auditoría. Los administradores pueden determinar quién y qué se audita y tener permisos exclusivos para visualizar los archivos correspondientes. [Mic96]

En Windows NT todo componente de hardware, software o datos, constituye un objeto cuyo acceso es estrictamente controlado por el sistema de seguridad. Este modelo de seguridad involucra los siguientes componentes [Mic96]:

- ***Proceso de Inicio de Sesión*** *(Logon Process)*. Un proceso realiza tres tipos de logon. Si un usuario inicia una sesión de trabajo, el proceso toma las credenciales del usuario (nombre y clave) y las verifica con el administrador de cuentas de seguridad (Security Account Manager, SAM). Si un usuario en una sesión de trabajo intenta acceder a recursos en otro sistema, el proceso comprueba al usuario en dicho sistema. También posibilita este proceso de verificación entre dominios.

- ***Autoridad Local de Seguridad*** *(Local Security Authority, LSA)*. Este componente es el punto central del sistema de seguridad que administra y coordina los logons, el acceso a objetos y otros eventos relacionados con la seguridad del sistema. La LSA se articula con el SAM y el monitor de referencia de seguridad (Security Reference Monitor, SRM). Además, está enlazado con una base de políticas de seguridad y un archivo de auditoría.

- ***Administrador de Cuentas de Seguridad*** *(Security Account Manager, SAM)*. Este componente administra la base de cuentas de usuarios. La LSA contacta al SAM cuando necesita verificar los permisos de un usuario para acceder a un objeto.

- ***Monitor de Referencia de Seguridad*** *(Security Reference Monitor, SRM)*. El SRM es el software que ejecuta en modo protegido que chequea si un usuario tiene permisos para acceder a un objeto o derechos para realizar alguna acción. Ejecuta la autenticación de acceso y el plan de generación de auditoría definida por la LSA.

Este conjunto de componentes conforma el subsistema de seguridad constituyendo un subsistema integrado ya que afecta a todo el sistema operativo. [She97]

Dominios

El concepto de dominio está íntimamente relacionado con la estructura administrativa de Windows NT. Constituye una agrupación lógica de servidores de red y otros equipos que comparten una información común de seguridad y cuentas de usuarios administrada por una autoridad central. (ver detalles en Apéndice B.1)

Cada dominio tiene su conjunto de políticas de seguridad y permite controlar el estado de las cuentas existentes por defecto o propias del sistema. Algunos puntos que se pueden delinear en cuanto a estas políticas son:

- Restricciones sobre las claves, tales como fecha de expiración o mínima longitud requerida.
- Eventos que provoquen el cierre temporal (lockout) de cuentas de usuario, como cantidad de intentos fallidos en el proceso de logon.

- Des/Habilitación de la cuenta de visitante.
- Políticas de auditoría con el fin de controlar el tipo de eventos generados en el dominio.
- Creación de grupos específicos con derechos especiales para manipular partes del sistema.

Debido a que cada dominio maneja un conjunto de cuentas de usuario propias, el particionamiento de una red en dominios determina una barrera de seguridad inmediata. Los usuarios de un dominio no pueden utilizar los recursos de otro dominio a menos que se establezcan las *relaciones de confianza* (ver definición en Apéndice B.1) correspondientes entre dichos dominios.

Aún así, el administrador de cada dominio debe otorgar los permisos requeridos para que usuarios de otros dominios accedan al propio. Todos los recaudos contemplados en relación con las relaciones de confianza y la concesión de permisos refuerzan la seguridad de la red.

Los dominios representan entidades administrativas seguras. Cuando un usuario ingresa a su cuenta, se halla en el dominio al que pertenece y puede acceder a los recursos del dominio para los cuales posea permisos. Los dominios pueden suministrar una barrera inicial de seguridad entre diferentes sectores de una organización si se encuentran adecuadamente configurados. [She97]

Sin embargo, al crear relaciones de confianza indiscriminadamente o permitir que usuarios y grupos accedan a los recursos en otros dominios, el plan de seguridad se debilita [She97]. No existen garantías de que los otros dominios implementen la seguridad apropiada a las circunstancias. La regla general indica aplicar la menor cantidad de relaciones de confianza. Y de ser necesaria, establecerla en un único sentido.

Esta cuestión se combina con el tema de los servicios prestados por un servidor. Hay que tomar particulares recaudos de las relaciones que se definen en un dominio en el cual se ejecutan servicios de naturaleza crítica que pudieran permitir que alguien los dañe.

En ambientes de trabajo Windows NT cuando un equipo inicia una sesión en la red, el servicio de inicio de sesión de la computadora cliente genera un canal de comunicación seguro con el correspondiente servicio *Servidor*. Este canal se considera seguro cuando las máquinas de ambos extremos se han identificado a sí mismas correctamente. Dicha comprobación se lleva a cabo por medio de las cuentas de equipos. Una vez establecido el canal seguro, puede comenzar la comunicación entre ambos equipos. Para mantener la seguridad durante la sesión de comunicación, resulta necesario configurar las cuentas de confianza internas entre las estaciones de trabajo y el servidor. [Mic97]

Planes de Seguridad de Dominio

La configuración del plan de seguridad de Windows NT ofrece tres alternativas distintas de seguridad respecto de las acciones de los usuarios o del equipo mismo que se aplican al dominio como un todo. [Mic97]

- El Plan de Cuentas controla cómo utilizan las contraseñas las cuentas de usuario y los eventuales bloqueos de dichas cuentas. Todas las opciones de configuración de este plan son críticas para activar una seguridad consistente, tales como longitud mínima de la palabra clave, duraciones mínima y máxima de ésta y posibilidad de reiteración de la misma.
- El Plan de Auditoría comprueba qué tipos de sucesos se graban en el registro de seguridad.
- El Plan de Relaciones de Confianza verifica en qué dominios se confía y cuáles son los dominios que confían.

Además, el Plan de Derechos de Usuario controla el acceso a los derechos otorgados a las cuentas de grupo y usuario. Estos derechos se aplican en el nivel de dominio y afectan a toda la seguridad del dominio.

Usuarios y Grupos

Como la mayoría de los sistemas operativos en ambientes de red, Windows NT se apoya en la estructura integrada por cuentas de usuario, derechos de acceso, permisos y logons seguros (ver detalles en Apéndice B.1 y C.1.a). Este esquema se controla mediante el sistema de seguridad ya citado.

- **Usuarios.** Todos los usuarios necesitan una cuenta a menos que ingresen al sistema como invitados. Existen dos tipos de cuenta de usuario: global y local.
  - Global. La cuenta de usuario global se genera en el entorno de Windows NT.
  - Local. La cuenta de usuario local se origina en el ambiente de un servidor específico.

- **Grupos.** La figura de grupo facilita la administración. En líneas generales, los administradores crean los grupos, agregan las cuentas de usuario, y asignan derechos y permisos al grupo en su conjunto. Así como en el caso de las cuentas de usuario, es posible definir grupos globales y locales.
  - Global. El grupo global permite manejar grupos en un dominio. También puede utilizarse para exportar grupos de usuarios a otros dominios.
  - Local. El grupo local manipula usuarios e importa grupos globales de otros dominios.

Las cuentas de usuarios contienen información sobre los usuarios, tales como nombre completo, nombre de usuario, contraseña, localización del directorio de inicio de sesión, información sobre cuándo y cómo puede iniciar una sesión, y las configuraciones personales de su ambiente de trabajo si utiliza Windows 95, 98 o NT.

Las propiedades de la cuenta de un usuario permiten establecer el estado de la misma (bloqueada, bloqueo por intentos fallidos, fecha de caducidad), el tratamiento de la palabra clave (cambio en próximo inicio de sesión, sin posibilidad de modificación por parte del usuario, sin expiración). También los días y el horario de conexión y las estaciones de trabajo autorizadas para iniciar la sesión.

La cuenta predefinida *Administrador* posibilita la administración del servidor al ser instalado. Es una cuenta de grandes privilegios porque proporciona acceso completo al sistema o dominio. En la misma situación se encuentra el grupo integrado *Administradores Locales*.

Perfil de Usuario

El *perfil de usuario* constituye la definición de un entorno de trabajo configurado y cargado en el sistema para un usuario o grupos de usuarios. Este contexto incluye esquemas de *desktop*, colores, conexiones de red e impresoras, teclas abreviadas, entre otras configuraciones.

Existen perfiles de usuario obligatorios impuestos por el administrador y que no son modificables por los usuarios. Esta característica es muy importante para la cuestión de seguridad ya que inhibe a los usuarios de ejecutar ciertas acciones ilegales. [She97]

Derechos

Los usuarios y los grupos tienen derechos que determinan las acciones que pueden llevar a cabo en el sistema. En la versión actual de Windows NT, el sistema define el conjunto de derechos de usuarios y grupos y no es posible modificarlo.

Si bien se pueden asignar derechos a cuentas de usuarios individuales, generalmente y de forma más eficiente se otorgan a grupos. Los grupos predefinidos tienen conjuntos de

derechos ya asignados. Vale destacar que la mayoría de estos privilegios son otorgados únicamente para usuarios con funciones administrativas.

Permisos

Los usuarios y los grupos necesitan permisos para trabajar con los objetos del sistema. Los permisos son otorgados por el administrador del sistema o por el dueño de objetos como archivos o directorios.

El conjunto de permisos de un recurso depende del tipo de objeto. Cabe recordar que mientras los derechos se aplican a todo el sistema, ya sea en el dominio o en forma local, los permisos se aplican a objetos específicos.

Con frecuencia los derechos se imponen a los permisos de los objetos. Por ejemplo, un operador con derecho a realizar copias de seguridad tiene el permiso de hacer copias incluso de aquellos archivos a los que el propietario ha denegado el acceso a todos los usuarios.

c. Particularidades de UNIX

El objetivo del modelo de seguridad en UNIX es proteger la integridad del sistema. Si bien estas cuestiones básicas en un sistema operativo multiproceso, multitarea y multiusuario han sido resueltas en los diseños originales, los diferentes servicios y la conectividad a redes han impuesto nuevos riesgos que han sido tenidos en cuenta en las implementaciones posteriores.

El modelo de seguridad de los recursos del sistema permite controlar quien tiene acceso (y de que tipo) a cada uno de los recursos, además de poseer un sistema de auditoría.

Cada usuario está identificado unívocamente dentro del sistema por un número de identificación (UID). Este número es utilizado para controlar el acceso a los recursos del sistema. Tanto los archivos como los procesos poseen un número de identificador del dueño. Este número es contrastado con el UID para validar los accesos a los mismos. De manera similar, el UID es utilizado por el sistema de auditoría para registrar las acciones del usuario.

Dado que en UNIX los dispositivos internos, externos e incluso la memoria pueden ser accedidos como objetos en el sistema de archivos, el modelo de seguridad de UNIX se basa fuertemente en el esquema de seguridad del sistema de archivos. Cada objeto del sistema de archivos posee una máscara de bits que define los permisos de acceso al mismo. Estos bits definen qué tipo de acceso tiene cada uno de los usuarios del sistema ya sean el dueño del recurso, los usuarios que pertenezcan al grupo del recurso o cualquier otro usuario.

Todo el esquema de control de acceso de los procesos de usuario a los recursos del sistema es manejado por el núcleo del sistema operativo de forma que no existe el acceso directo a los objetos. De la misma manera, el núcleo es quién controla los mensajes del sistema de auditoría, que se almacenan en el mismo sistema de archivos.

El mecanismo de auditoría de UNIX es bastante más complejo que el de Windows NT. En primer lugar existe un proceso encargado de manejar los mensajes de error del sistema. Estos mensajes incluyen ciertas acciones que pueden relacionarse con la seguridad del sistema como un intento fallido de acceso a la cuenta de administrador. Por otra parte cada subsistema y servicio provisto maneja habitualmente sus propios mensajes de error y auditoría.

Por último y opcionalmente, los diferentes sistemas UNIX, en particular los comerciales, proveen algún sistema de auditoria más completo para cumplir con los estándares C2, y que permiten auditar con el detalle de cada acción de cada usuario y conservan los registros en

forma cifrada para evitar su alteración. Estos sistemas de auditoría no utilizan un estándar por lo que su uso se ve limitado a conjuntos de servidores homogéneos.

Arquitectura

Los servicios del modelo de seguridad de UNIX son provistos por distintos procesos. Estos realizan llamadas al núcleo del sistema para todas las funciones de validación de permisos y derechos. Este esquema facilita la modificación de ciertos aspectos del sistema sin comprometer la seguridad del mismo que es manejada íntegramente por el núcleo.

El acceso al servidor es exclusivo de los usuarios autorizados. Los mismos deberán autenticarse mediante el proceso de login. Cada usuario posee un nombre de usuario unívoco (username) que deberá ingresar para identificarse en el sistema y a su vez deberá ingresar una palabra clave (password) que sólo él conoce, como medio de autenticación. [Gar96]

Cada usuario tiene asociado un número de identificación de usuario (User ID o UID) como medio de identificación unívoca dentro del sistema. Este número de identificación de usuario es utilizado por el sistema para validar cada una de las acciones que el usuario desee realizar. Asimismo, cada objeto del sistema de archivos tiene un dueño que no es más que un UID. Lo mismo ocurre con los procesos en ejecución. [Gar96]

El núcleo del sistema compara, ante cada intento de acceso de un usuario a un objeto o proceso del sistema, el UID del usuario con el correspondiente al objeto accedido. De esta manera el sistema puede controlar los accesos de manera sencilla y efectiva.

El mismo UID es utilizado por el subsistema de auditoría para llevar registro de las acciones de cada usuario. El mismo permite identificar al usuario y, junto con la información generada por el núcleo, registrar la acción realizada, ya sea que haya tenido autorización o que el sistema la haya cancelado por falta de permisos.

Usuarios

Los usuarios que deseen acceder a un servidor deberán tener una cuenta que los habilite, además cada usuario deberá tener una clave personal que lo autentifique ante el sistema. Es preciso indicar que el sistema de identificación y autenticación se puede extender para utilizar otros medios de validación como ser claves de utilización única, llaves o lectores biométricos.

Las cuentas de usuario pueden estar definidas localmente en un servidor o pueden estar definidas dentro del dominio. Además, cada usuario puede pertenecer a uno o más grupos que determinan los accesos que tendrá dicho usuario a los diferentes recursos del sistema.

El sistema posee un usuario especial que tiene acceso casi irrestricto a todos los recursos. También hay un reducido grupo de usuarios que controlan servicios del sistema y que no están identificados directamente con un ser humano. Habitualmente estos usuarios tienen sus cuentas bloqueadas de forma tal que no es posible ingresar al sistema utilizando estas cuentas.

El supervisor

Cada sistema UNIX viene con un usuario especial que tiene un UID de 0. Este usuario es conocido como el supervisor, superusuario o administrador y tiene usualmente el nombre de usuario *root*. La cuenta del supervisor es utilizada dentro del sistema para realizar muchas tareas desde permitir el ingreso de usuarios hasta llevar la información de auditoría del sistema.

Por esta razón, el supervisor tiene un control casi absoluto del sistema operativo. Casi todas las restricciones de seguridad no se aplican a los procesos que ejecutan por el supervisor y muchos de los controles y avisos son inhabilitados.

Dado que lo importante dentro del sistema UNIX no es el nombre del usuario sino el número de identificación. Lo que hace a la cuenta *root* especial es que tiene asignado el número de usuario 0. Pero cualquier otro usuario que tenga el UID 0, tendrá exactamente los mismos derechos que el superusuario.

Debido a que muchos de los controles que se aplican a los usuarios comunes, no se realizan sobre la cuenta del supervisor, un ligero error de tipeo puede arruinar el sistema. Es importante que la persona que realice las tareas de administración, no utilice la cuenta *root* como su cuenta personal. En lugar de esto, sólo deberá utilizar dicha cuenta para realizar estrictamente las tareas para las que requiera accesos especiales.

Muchas versiones de UNIX impiden el ingreso directo al sistema del supervisor, a menos que lo haga desde la consola del equipo. Esto permite aumentar en gran medida la seguridad del sistema ya que es necesario acceder a la consola del equipo, que posiblemente esté dentro de un centro de cómputos con cierta seguridad física.

El supervisor es el mayor problema de seguridad del UNIX pues el supervisor tiene acceso casi irrestricto. Una vez que un atacante obtiene acceso como supervisor, el mismo puede hacer virtualmente lo que desee con el sistema. Esto explica el por qué la mayor parte de los atacantes que logran acceder a sistemas UNIX tratan de obtener privilegios de supervisor. Una vez que la cuenta del supervisor ha sido comprometida, el sistema completo está en peligro.

Grupos

Es posible definir grupos de usuarios dentro del sistema. Estos grupos pueden estar definidos en un servidor localmente o en el dominio (ver Dominios). Cada recurso tiene derechos de acceso definidos para los grupos.

Derechos

Los usuarios tienen derechos sobre ciertos recursos del sistema. Por ejemplo pueden realizar cualquier manejo de archivos dentro de su directorio HOME, o cambiar su clave de identificación o la configuración de sus sesiones de trabajo.

Permisos

Cada recurso tiene definidos tres grupos de permisos. Estos privilegios determinan el nivel de acceso que tendrán los diferentes usuarios a dicho recurso. Los permisos posibles son lectura, escritura, ejecución y listado, donde ejecución se aplica únicamente a archivos y listado únicamente a directorios.

Los permisos se aplican al dueño del recurso, al grupo al cual el recurso esté asociado y al resto de los usuarios del sistema. También es posible definir permisos para que un usuario pueda acceder a recursos como si fuera otro usuario o como si perteneciera a otro grupo, mediante este mecanismo es posible hacer que un usuario tenga permisos especiales sobre ciertos recursos.

Algunas versiones de UNIX tienen implementado un sistema de listas de control de accesos por lo que es posible mejorar la granularidad de la seguridad de estos sistemas.

Dominios

Es posible crear dominios de servidores donde los usuarios y recursos definidos en el mismo estén disponibles en cualquiera de los servidores del dominio (ver detalles en Apéndice B.2). El sistema operativo UNIX utiliza los estándares NIS (Network Information System) y NIS+ para la definición de los dominios.

Cabe destacar que prácticamente todas las versiones de UNIX soportan NIS. Sin embargo, no sucede lo mismo con NIS+ dado que su implementación es más novedosa y compleja.

Este sistema sigue el modelo cliente-servidor donde se define un servidor NIS, cero o más servidores NIS esclavos y uno o más clientes. Permite definir usuarios en todo el dominio así como compartir todo tipo de recursos. Los servidores clientes del dominio se validan contra el servidor NIS. [Ste91]

NIS nunca fue diseñado como un servicio seguro sino más bien como una herramienta de administración. Originalmente los problemas de seguridad fueron notorios y la mayoría de ellos fue arreglada con el tiempo. De todas formas el mayor peligro de NIS hoy día es que es posible obtener información de configuración de los equipos, listados de usuarios y *passwords* cifradas, listados de máquinas y direcciones IP y otros datos, simplemente conectando una máquina a un dominio NIS cualquiera.

Es por ello que se debe tratar de evitar que los nombres de los dominios NIS sean conocidos fuera de la red. Además, muchas implementaciones de NIS modernas permiten especificar desde que redes se aceptan pedidos de mapas NIS, lo que hace imposible capturar la información de los mapas sin acceso directo a las redes afectadas.

A pesar de estas mejoras, NIS no ofrece las características de seguridad necesarias para alcanzar la clase C2 (consultar Apéndice G). Por tal motivo, se diseñó una versión nueva que cumple dichos requerimientos.

En este sentido, NIS+ protege la estructura de dominios y tablas mediante un proceso de autorización y autenticación. En primer lugar, cada componente en el dominio especifica el tipo de operaciones que acepta y de quién. En segundo lugar NIS+ trata de autenticar cada pedido de acceso al dominio. Una vez identificado el origen del pedido, chequea que el mismo esté autorizado para la operación particular que requiere. [Ram94]

La entidad que realiza un pedido al servidor NIS+ desde un cliente es llamada *principal*, que puede ser un usuario, un proceso o una máquina. Los *principals* son identificados por sus credenciales. NIS+ utiliza dos tipos de credenciales, locales y DES que son utilizadas para autenticar al *principal* y determinar si el mismo tiene autorización de realizar una operación determinada.

Los permisos que se manejan dentro del entorno NIS+ permiten controlar cuatro tipos de acciones sobre los objetos: leer, modificar, crear y destruir. Cada comunicación de un cliente a un servidor es, de hecho, un pedido para realizar una de estas acciones.

Los objetos de NIS+ especifican sus permisos de acceso como parte de su definición. Además, los permisos no están especificados para cada *principal* sino más bien a cuatro clases de *principals*, el dueño, el grupo, el resto de los *principals* y el usuario especial *NOBODY* que representa a los clientes que no son *principals*.

Los servidores pueden operar en tres niveles de seguridad. Estos niveles especifican los tipos de credencial que un *principal* debe entregar para ser autenticado. El nivel de seguridad 0 es usado durante la configuración inicial y el testeo del ambiente. En este nivel se garantiza el acceso de cualquier *principal* a todos los objetos del dominio.

El nivel de seguridad 1 está diseñado para testeo sin el uso de la autenticación DES. El nivel 2 es el que se utiliza en el sistema en producción. Este se asume por defecto y sólo autentica pedidos que utilizan credenciales DES.

La credencial local de un *principal* es simplemente su número de identificación de usuario y es por ello que no es utilizada en el sistema en nivel de seguridad 2. La credencial DES es más compleja y está compuesta por el *secure RPC netname* y un campo de verificación. Además, se utiliza generalmente la palabra clave cifrada del cliente para generar un par de claves de cifrado de clave pública siguiendo el esquema de Diffie-Hellman que será utilizado subsecuentemente para las comunicaciones entre el cliente y el servidor.

IV. Análisis Comparativo

Tanto en Windows NT como en las versiones modernas UNIX, la seguridad es una cuestión primigenia en el diseño del sistema operativo. Las características básicas de seguridad están integradas al sistema operativo desde su concepción. Este punto es sumamente importante ya que, tal como se mencionó anteriormente, todo aquello que no forme parte del sistema operativo constituye una vulnerabilidad del mismo.

Los conceptos que maneja el modelo de seguridad son similares en lo que respecta a los componentes y las funcionalidades provistas. Ambos sistemas operativos proveen las funciones de seguridad básicas de un sistema operativo moderno para servidores, tales como autenticación, control de acceso, auditoría, entre otras. Sin embargo, existen diferencias en la implementación de estos servicios de seguridad en cuanto al modo de operar y a las facilidades que proveen.

Todas estas cuestiones son cubiertas con mayor nivel de detalle en los próximos capítulos.

Asimismo, este conjunto de servicios se fundamenta en una estructura semejante de cuentas de usuario, grupos, permisos y derechos. Los puntos distintivos se asientan sobre las características y la flexibilidad que poseen estos objetos en el contexto de cada uno de los sistemas operativos.

Los usuarios siempre deben poseer una cuenta del sistema. Dichos usuarios pueden reunirse en distintos grupos pero tienen asignado al menos uno como primario. Básicamente, el concepto de grupo apunta a proveer una facilidad administrativa. Sin embargo, en entornos UNIX, esta figura también posibilita manejar el acceso a los recursos.

En Windows NT existen dos categorías de usuarios y grupos que permiten manejar el concepto de globalidad de dominios. Este procedimiento permite exportar usuarios y grupos a otros dominios siempre bajo relaciones de confianza determinadas.

En ambos sistemas operativos existe una figura de superusuario con privilegios casi absolutos encargada de la administración del sistema en su totalidad. Muchos usuarios pueden ser

definidos con tales características con la consecuente degradación de la seguridad del sistema operativo.

Tanto en Windows NT como en UNIX, se contempla la posibilidad de configurar perfiles de usuario que determinan las características de su ambiente de trabajo. A diferencia de los usuarios de UNIX que manejan el perfil enteramente a su gusto, en Windows NT existen perfiles obligatorios impuestos por el administrador del sistema e inmodificables por los usuarios. Además, Windows NT ofrece la posibilidad de definir políticas de usuario como facilidad administrativa para implementar restricciones sobre la configuración de los ambientes de trabajo de los usuarios.

Mientras que el concepto de dominio es fundamental dentro de la arquitectura de Windows NT, bajo UNIX este aspecto es opcional. Si bien en principio dicho concepto en NT representa una entidad administrativa, también brinda una barrera inicial de seguridad al permitir establecer políticas que afectan a la totalidad del dominio en cuestión. En cambio, los dominios en UNIX constituyen puramente una facilidad administrativa. Es más, el sistema NIS introdujo graves problemas de seguridad, posteriormente resueltos con la aparición de NIS+.

Sin embargo, es importante no confiar en la sensación de seguridad que parecieran brindar los dominios de Windows NT. Tal como se ha visto, los dominios son grupos de computadoras que comparten una misma política de seguridad y la misma base de datos de cuentas de usuario. Este hecho permite lograr una separación lógica de usuarios y de los sistemas de información que refleje la estructura de departamentos, divisiones o sucursales de la organización. De este modo, los dominios sirven principalmente como una herramienta de administración que simplifica el manejo de las cuentas de los usuarios.

Los permisos se aplican a cada objeto en particular. La flexibilidad que ofrece NT es mucho mayor que la que presenta el enfoque de UNIX, tanto en lo que se refiere a tipo de permisos como a su asignación. Aun así, el esquema más sencillo de UNIX es igualmente funcional.

Tanto en Windows NT como en UNIX, existe un conjunto de acciones que pueden ser restringidas para un grupo de usuarios, más allá de los administradores del sistema. En ambos casos estos derechos se imponen a los permisos que posea un usuario. A modo de ejemplo, un usuario puede tener permiso para ejecutar un archivo pero si no tiene derecho para llevar a cabo tal acción, cualquier intento de ejecución resulta inocuo.

En cuanto al nivel de seguridad de la NCSA, el modelo de seguridad de Windows NT versión 3.5 cumple el nivel de seguridad C2 de acuerdo con la definición del Departamento de Defensa de E.U.A. además de algunas funcionalidades de nivel B2. La versión 4.0 está en proceso de evaluación. Sin embargo, al respetar los fundamentos del modelo de seguridad de la versión anterior se espera obtener la misma calificación.

Las distintas versiones de UNIX difieren en el nivel de seguridad del sistema operativo. Mientras que las versiones libres generalmente están por debajo del nivel C2, la mayoría de las comerciales alcanzan dicha calificación. Además, existen versiones especiales (*Trusted Solaris* entre otras) originalmente destinadas a los ámbitos gubernamentales y militares, cuyos modelos de seguridad cumplen un importante número de requisitos de nivel B.

Autenticación, Autorización y Auditoría

Los sistemas operativos modernos en ambientes de red deben administrar la asignación de recursos de modo que únicamente los usuarios adecuadamente validados tengan acceso a los mismos y de acuerdo con los permisos que se les hayan otorgado en el momento que corresponde. [Mic98]

Asimismo, es importante mantener un registro de este tipo de eventos para evaluar los sucesos ocurridos con relación a estos servicios y detectar eventuales actividades ilegales o sospechosas.

Sin embargo, no todos los servicios de seguridad son implementados en el sistema operativo. Ciertas funciones de seguridad se insertan más apropiadamente en otros niveles como en el contexto de la aplicación de usuario, de modo de poder controlar las variables que introduce la misma.

Autenticación

I. Conceptos Básicos

La primera cuestión que debe resolver el sistema operativo radica en determinar si la persona que está sentada delante de la computadora local o remota es quién dice ser. La capacidad de proveer un alto nivel de certeza a esta respuesta es una de las propiedades de seguridad más significativas entre aquellas que un sistema operativo debe ofrecer. [Gar96]

Pese a que se suele confundir el concepto de autenticación con el de identificación, los mismos son muy distintos. Mientras que la identidad de una persona es pública, la forma en que establece su identidad (autenticación) requiere que la misma provea alguna información que sólo él y el autenticador conozcan.

El mecanismo de autenticación apunta a relacionar la identidad del sistema, utilizada para rastrear las actividades en el sistema, con una identidad en la vida real.

El principal medio de autenticación de los usuarios frente a los sistemas de computación son las palabras clave (passwords), sin embargo, el uso de las mismas para acceder a los sistemas es objeto de abuso (publicación de contraseñas, préstamo de cuentas, etc.) por lo que no resulta en un método suficientemente seguro para identificar a los usuarios.

El proceso de autenticación no se limita a los usuarios ni al proceso de login (ingreso al sistema) sino que, en un sistema de computación distribuido, es fácil identificar al menos tres tipos diferentes de autenticación; la autenticación de la identidad del usuario, la autenticación del origen de un cierto mensaje y la autenticación del contenido de un mensaje.

En la mayor parte de los casos, estos dos últimos tipos de autenticación utilizan técnicas criptográficas y se encuentran implementados en el nivel de aplicación, lo que representa un grado de debilidad.

Mecanismos de Autenticación de Usuarios

Hay tres formas básicas de identificar a un individuo: por algo que él sabe, por algo que él posee o por algo que él es [Tan92]. Cualquiera de estas alternativas puede servir para

identificar unívocamente a un individuo pero combinadas proveen una autenticación mucho más fuerte que cualquiera de ellas por separado.

El sistema más difundido de utilizar palabras clave para autenticar a un usuario se basa en el secreto compartido entre el usuario y la máquina que son los únicos que conocen dicha contraseña (en teoría). Sin embargo, este sistema ha probado tener muchos problemas en cuanto a la facilidad de descubrir claves por parte de terceras personas. Si bien el cambio periódico de contraseñas mejora esta situación, también ayuda al olvido de las mismas por parte de los usuarios.

En ambientes donde la seguridad es de mayor importancia, el uso de elementos a manera de llaves que el usuario debe poseer para el ingreso a los sistemas esta más difundido. Este sistema también tiene grandes inconvenientes si es utilizado como único medio de autenticación ya que las llaves pueden ser perdidas o robadas. Es por ello que se utilizan habitualmente junto con contraseñas u otros datos que el usuario debe conocer.

Una variante con respecto a la autenticación por medio de algo que el usuario es, consiste en considerar la información sobre la locación del usuario, tales como dirección del adaptador de red o código identificador de llamada [Mic98]. La desventaja de este enfoque es la poca flexibilidad ante cambios. Por ejemplo, si la placa de red se daña y es reemplazada, resulta necesario actualizar los datos del usuario para permitir la correcta validación del mismo.

Varias tecnologías están surgiendo que utilizan características físicas de las personas como medio de autenticación. La gran ventaja de las mismas es que no requieren que el usuario conozca algo especial ni que lleve consigo más que su persona. Los métodos más usuales son la lectura de huellas dactilares, patrones de voz, forma de la mano o patrones de la retina del ojo. Estos sistemas tienen aún problemas pero se están haciendo grandes avances al respecto, habiendo muchos sistemas de este tipo en funcionamiento.

Generalmente, los sistemas de autenticación fuerte requieren la aplicación de al menos dos de estos mecanismos en forma simultánea. Por ejemplo, el acceso a un cajero automático implica la tenencia de una tarjeta y el conocimiento de un código personal.

Asimismo, la confidencialidad de la información utilizada para autenticar es extremadamente importante. Si los datos correspondientes al nombre de usuario y la clave se transmiten abiertamente por la red, la autenticación confiable es imposible. De forma similar, si el código personal del banco es asentado en la tarjeta correspondiente, se diluye el concepto de autenticación fuerte.

El servicio de autenticación de un sistema operativo puede ser evaluado en base a los mecanismos que soporta, la robustez de estos enfoques y la integración de la información de autenticación en el conjunto de las operaciones de seguridad.

II. Implementación del Servicio de Autenticación

a. Características Comunes

Los sistemas operativos Windows NT y UNIX requieren la autenticación del usuario antes de acceder a algún recurso del sistema. Todo usuario que accede al servidor debe identificarse mediante el nombre unívoco que tiene asignado en el momento de iniciar la sesión de trabajo.

Por defecto, el proceso de autenticación se basa en el secreto compartido entre el usuario y el servidor en cuestión, es decir, la palabra clave del usuario. Sin embargo, es posible incorporar otros medios de validación adicionales de modo de obtener una autenticación fuerte.

Tanto Windows NT como algunas versiones de UNIX proveen facilidades para configurar la política sobre palabras clave según propiedades de las mismas, tales como antigüedad, historia, longitud mínima, combinación de distintos tipos de caracteres y demás.

b. Particularidades de Windows NT

Inicio de Sesión

En Windows NT es necesario presionar la secuencia de teclas Ctrl+Alt+Del para loguearse (ver detalles en Apéndice C.1.a). Este procedimiento asegura que el sistema es esencialmente reiniciado y remueve cualquier utilidad falsa o troyana. [She97]

El proceso de inicio de sesión en los sistemas Windows NT está manejado por el servicio denominado *NETLOGON* y coordinado por la Autoridad de Seguridad Local (LSA). Estos componentes forman parte del subsistema de seguridad.

Dependiendo del tipo de inicio de sesión (local, en el dominio o en un dominio confiable), el proceso de autenticación se ejecuta en el servidor local o en el remoto mediante autenticación transferida.

Ya sea en uno u otro caso, la LSA correspondiente contacta al Administrador de Cuentas de Seguridad (SAM) que verifica la existencia del par usuario-clave en la base de cuentas de usuario local. De resultar positiva la comprobación, el SAM retorna la información de identificación del usuario con la que la LSA construye la ficha de acceso del usuario. Finalmente, el proceso de logueo del equipo del usuario utiliza esta ficha para lanzar el proceso inicial del usuario (el *shell*, es decir, el ambiente del usuario).

Cuando un usuario intenta el inicio de sesión en un servidor, la LSA trata la identificación de un usuario mediante la invocación al *paquete de identificación MSV1_0*. Los paquetes MSV pueden generar alguno de los inicios de sesión citados anteriormente.

Cuando el usuario accede al equipo que puede identificarlo directamente, la clave en texto plano que se ingresa por teclado es convertida mediante una función de un sentido en una contraseña OWF (One Way Function). El resultado obtenido es contrastado posteriormente con la contraseña OWF de la base de datos SAM almacenada en forma local. De este modo, la palabra clave real nunca se expone en el proceso. [She97]

En cambio, cuando el usuario intenta un inicio de sesión remoto en un servidor de dominio, se lo desafía a retornar una respuesta combinada con cierta información que sólo el servidor y el mismo conocen. Este proceso se fundamenta en el secreto compartido entre ambos definido en el momento de creación de la cuenta del usuario en cuestión. Así, el paradigma de desafío/respuesta permite la validación de usuarios sin transmitir claves de acceso a través de la red.

Los pasos involucrados en este proceso son:

- El usuario solicita el acceso al servidor de dominio.
- El servidor remite un desafío al proceso de logueo en el cliente.
- Este proceso combina dicho desafío con el nombre del cliente y cifra el producto utilizando como clave la contraseña del usuario que ya se encuentra cifrada bajo OWF.
- El servidor ejecuta el mismo procedimiento tomando la clave del usuario almacenada en la base de datos SAM y luego compara su propio resultado con la respuesta del cliente. De coincidir, el usuario es considerado válido.

De este modo, la contraseña del usuario no se transmite por la red reforzando la vulnerabilidad ante escuchas mediante un cifrado doble. El secreto compartido por el servidor y el usuario es utilizado para cifrar el desafío propuesto por el primero. Finalmente se contrastan los resultados de estas operaciones, no siendo necesario descifrar nada.

Windows NT también soporta una forma alternativa de autenticación denominada *Autenticación Lan Manager* (LM Authentication), con el fin de mantener la compatibilidad con versiones anteriores de Windows (Windows 3.1, Windows for Workgroups, Windows 95) [Lew98]. El servidor NT que autentica a los usuarios de dominio puede aceptar este modo de autenticación o permitir únicamente autenticación NT, de acuerdo con lo que determine el administrador del sistema.

Palabras Clave

Las palabras clave en Windows NT pueden contener hasta 14 caracteres, distinguiendo entre mayúsculas y minúsculas y permitiendo un conjunto de símbolos especiales.

Cabe recordar que el plan de cuentas del dominio permite establecer ciertas propiedades de las palabras clave, tales como longitud requerida, duraciones mínima y máxima, historia, así como características relacionadas con el bloqueo de la cuenta ante intentos fallidos de autenticación.

El Service Pack 2 incluyó una librería dinámica opcional (*passfilt.dll*) que fuerza el uso de palabras clave que contengan caracteres alfanuméricos y especiales. De activar esta facilidad, los usuarios deben seleccionar una secuencia de al menos 6 caracteres combinando al menos 3 de los tipos de caracteres permitidos (mayúsculas, minúsculas, números, símbolos especiales). [Lew98]

El SAM almacena la información sobre los usuarios en la base de datos de cuentas de seguridad. Las entradas de usuario y grupo contienen los nombres de usuario y grupo, los identificadores de seguridad y las palabras clave cifradas. El contenido de estos archivos se encuentra en un formato cifrado para dificultar eventuales ataques. [She97]

La base de datos actual forma parte del Registro y es guardada en la carpeta *System32\CONFIG* del directorio del sistema operativo. Desde el punto de vista del sistema operativo, sólo los procesos del sistema, tales como el proceso de logon y la LSA, pueden acceder al SAM. Estos procesos permiten la interacción con el SAM mediante programas administrativos ejecutados por los usuarios autorizados.

Cada contraseña se encuentra doblemente cifrada en la base de datos del SAM. El primer cifrado consiste en una versión de la función de un sentido (OWF) de clave en texto plano. El resultado se vuelve a cifrar para complicar aún más su descifrado. Para validar al usuario, la contraseña cifrada se compara con la clave correspondiente contenida en la base de datos del SAM sin necesidad de descifrarla.

De hecho, la base de datos del SAM nunca se descifra. Ni siquiera el SAM puede hacerlo. Este enfoque impide que alguien desarrolle un programa que haga uso de las API del SAM para leer el contenido de dicha base.

Cabe aclarar que Windows NT implementa la versión case-sensitive de la palabra clave para la autenticación de usuarios en entornos de servidores y estaciones de trabajo NT. Con la autenticación LM, el servidor NT traslada la palabra clave a mayúsculas y almacena una segunda entrada en la SAM junto a la versión correspondiente de NT.

c. Particularidades de UNIX

En la mayor parte de los sistemas UNIX, el uso de palabras clave para autenticar usuarios es el sistema en uso. Las palabras clave han sido parte integral del UNIX desde sus orígenes y tienen la gran ventaja que no se requiere de ningún tipo de equipamiento especial para su utilización. La desventaja es que es relativamente sencillo averiguar o conseguir la palabra clave de algún usuario del sistema para luego utilizarla para entrar al mismo.

Al ingresar al sistema, el usuario debe proveer su nombre de cuenta y a continuación su palabra clave, que no es impresa en la pantalla. Luego el sistema compara la palabra clave ingresada con la que el mismo mantiene y, de ser iguales, le permite al usuario ingresar. (consultar Apéndice C.2.a)

Algunos sistemas UNIX permiten que, tras el ingreso de una palabra clave inválida una cierta cantidad –pequeña- de veces, la cuenta queda inhabilitada y sólo el administrador puede volver a habilitarla.

Cada sistema UNIX es provisto de un conjunto de cuentas por defecto que son utilizadas por diferentes subsistemas del sistema operativo. La mayor parte de ellas no tienen una palabra clave válida por lo que no es posible entrar al sistema utilizándolas al no haber forma de autenticarse al sistema.

Además, hay algunos paquetes de programas que crean cuentas con palabras clave prefijadas. Es importante controlar tanto las cuentas del sistema como las de programas e inhabilitar o modificar las palabras clave según sea posible.

Si bien la mayor parte de los sistemas UNIX exigen el ingreso de una palabra clave al administrador durante la instalación, algunos sistemas no le asignan palabra clave alguna, dejando así una puerta de entrada al sistema.

Palabras Clave

Las palabras clave suelen ser de entre uno y ocho caracteres pero algunos sistemas más nuevos permiten palabras clave más largas. Algunos sistemas UNIX permiten determinar un número mínimo de caracteres para la palabra clave, usualmente de seis y, además, exigir el uso de caracteres alfabéticos, numéricos y de símbolos en una misma palabra clave. [Gar96]

El nombre de cuenta es utilizado para identificar al usuario y la palabra clave para autenticarlo. Cuando el usuario se conecta al sistema debe ingresar su palabra clave. El sistema entonces compara la palabra ingresada con la que tiene guardada en un archivo. Si el sistema guardara las claves en un archivo de texto, sería relativamente sencillo obtener una copia del archivo que guarda las claves y así poner en riesgo a todo el sistema.

Para evitar que las palabras clave sean visibles, las mismas no son conservadas. Lo que se coloca en el archivo es un valor generado utilizando la clave del usuario para cifrar un bloque de ceros con una función que no puede ser revertida (OWF). Cuando el usuario se conecta al sistema e ingresa su palabra clave, el sistema no compara la palabra ingresada con la guardada sino que cifra con la palabra clave ingresada un bloque de ceros y compara el resultado con el que se conserva en el sistema.

Este esquema de seguridad depende de la confiabilidad del algoritmo de cifrado y de la dificultad de adivinar la palabra clave del usuario. Hasta la fecha el algoritmo ha probado ser

muy resistente a los ataques pero desdichadamente los usuarios suelen elegir palabras clave fáciles de adivinar.

El algoritmo de cifrado está basado en el algoritmo DES del Instituto Nacional de Estándares y Tecnología (NIST) de los Estados Unidos. En operación normal DES utiliza una clave de 56 bits para cifrar bloques de 64 bits de longitud, ya que usa 8 de ellos de redundancia.

UNIX utiliza la palabra clave del usuario como clave para cifrar un bloque de 64 bits de ceros. Luego vuelve a cifrar el resultado con la misma clave 25 veces y el resultado final es desempaquetado y almacenado como una cadena de 11 caracteres. Además, en el momento de modificar la clave, el sistema utiliza la hora del día para crear un número de 12 bits que modifica ligeramente el resultado del cifrado. Este valor es conservado junto con la clave cifrada y se utiliza para cifrar y comparar la clave ingresada por el usuario al iniciar una conexión.

Aunque el algoritmo de cifrado esta disponible, no se conoce ninguna técnica que permita obtener la palabra clave del usuario, a partir del bloque cifrado con la misma. De este forma, la única manera de quebrar el sistema de seguridad de las palabras clave en UNIX es por fuerza bruta o mediante el uso de diccionarios.

El uso del número de 12 bits (salt value) hace muy compleja la creación de diccionarios cifrados de claves ya que cada clave debe ser cifrada con cada combinación posible de 12 bits (4096 posibilidades). De todas formas, los sistemas actuales permiten la creación de dichos diccionarios por lo que no es posible considerar el sistema de palabras clave como inquebrable.

Algunos sistemas UNIX más nuevos traen un algoritmo de cifrado que utiliza palabras clave de 16 o más caracteres y, además, utilizan un número mayor para variar el cifrado. La ventaja obvia es que, en estos sistemas, el almacenamiento es mucho más seguro, pero tienen la gran desventaja de no ser compatibles con otros sistemas con los que puedan tener que interactuar.

Ya que el punto débil en el sistema de palabras clave es precisamente la palabra clave que el usuario haya elegido, muchos sistemas UNIX actuales permiten configurar el sistema de palabras clave de forma de obligar al usuario a elegir palabras clave más difíciles de descubrir.

Si bien no hay un estándar de configuraciones posibles, algunas de las configuraciones posibles de la palabra clave comprenden longitud mínima, número mínimo de caracteres alfabéticos y no alfabéticos, número máximo de repeticiones de un carácter, mínimo número de caracteres diferentes entre dos claves consecutivas, tiempo de validez, historia de la clave y nombre de un archivo con palabras clave prohibidas en el sistema.

Algunos sistemas UNIX permiten definir un tiempo de validez de las palabras clave, forzando al usuario a cambiar la misma, una vez que el tiempo establecido expiró. También es posible hacer que una palabra clave excesivamente vieja, haga inhabilitar automáticamente la cuenta respectiva. Estos sistemas suelen llevar un registro de las últimas palabras clave utilizadas por el usuario de forma de prohibir que estas se repitan.

Además, muchos sistemas UNIX proveen generadores de palabras clave que pueden ser utilizados para forzar una palabra clave "buena". Los dos problemas principales de los

generadores de palabras clave es que las claves generadas no son fáciles de recordar y, por ello, los usuarios terminan anotándolas en un papel y, además, los usuarios siempre prefieren elegir palabras clave que tengan algún significado personal, que es lo que las suele hacer fáciles de descubrir.

Otras técnicas de uso de palabras clave que pueden ser utilizadas en sistemas UNIX, ya sea que el sistema lo provea o sea un producto de terceras partes, incluyen: palabras clave de uso único y tarjetas electrónicas de generación de palabras clave.

III. Análisis Comparativo

Tanto Windows NT como UNIX presentan el mismo enfoque de autenticación de usuarios por medio de la utilización de palabras clave. En consecuencia, ambos presentan las falencias propias de este mecanismo de autenticación y las respectivas con respecto a las implementaciones particulares.

Generalmente, tanto usuarios como administradores facilitan la tarea de los intrusos utilizando claves simples de adivinar o asentándolas por escrito en diversos lugares. Por tal motivo, la implementación de algún tipo de política para reforzar el tratamiento y la elección de palabras clave contribuye a reforzar la seguridad del esquema. Este tipo de facilidades existen en Windows NT y en algunas versiones de UNIX. Las opciones de configuración también son semejantes en ambos casos.

Más de un usuario podría tener la misma palabra clave dentro del sistema, esto querría decir que se eligió una mala palabra clave. Una persona puede tener más de una cuenta en un sistema NT o UNIX pero éstas tendrán distintos nombres de cuenta.

El almacenamiento de las palabras es un punto crucial en la seguridad del sistema. Si bien en ambos sistemas operativos se aplica alguna clase de cifrado, la potencia del algoritmo involucrado y la adecuada elección de contraseñas, determinan la calidad del resultado para afrontar ataques por fuerza bruta, a los que ambos sistemas son susceptibles.

Bajo Windows NT no se utiliza una semilla generada al azar (random salt value) para cifrar la palabra en cuestión. Este procedimiento permite que una misma palabra clave tenga distintos cifrados en distintos lugares, dificultando los ataques de diccionario ya que cada valor debe ser analizado individualmente con la semilla correspondiente. Dado que Windows NT no ofrece esta propiedad, cada entrada del diccionario está relacionada con un código (hash) determinado y los programas de quebrado de contraseñas pueden realizar chequeos del SAM con mayor velocidad y facilidad.

Además, la compatibilidad de Windows NT con versiones anteriores de Windows mediante la autenticación LM no hace más que empeorar la cuestión y aumentar las probabilidades en favor de los atacantes. La activación de este tipo de autenticación en el servidor NT implica la generación de dobles entradas en la SAM para cada palabra clave elevando aún más las posibilidades de quebrar la seguridad.

A diferencia de Windows NT, UNIX utiliza un número de 12 bits generado al azar para cifrar cada palabra clave. Este mecanismo implica que cada contraseña pueda tener más de 4000 posibilidades de cifrado. Por lo tanto, el éxito de los ataques de fuerza bruta se torna más dificultoso que en el caso de Windows NT ya que se debe analizar cada una de estas alternativas por separado.

Una de las cuestiones a considerar es la existencia de programas troyanos que intercepten los datos del usuario durante el inicio de sesión. Ambos sistemas operativos son vulnerables a este tipo de inconvenientes. En el caso de trabajar en un contexto Windows NT puro, es

posible aprovechar la funcionalidad de Trusted Path para evitar esta amenaza. Asimismo, resulta vital considerar la seguridad de las estaciones de trabajo en donde puede quedar algún registro de las credenciales del usuario a disposición de intrusos.

Otro problema fundamental es la transmisión del par usuario-contraseña a través de la red. La única manera segura de utilizar un servidor remotamente a través de una red es utilizando palabras clave de uso único o el cifrado de datos. Si las formas de conexión remota no implementan algún mecanismo de cifrado en la comunicación o en el intercambio de claves, la seguridad se ve considerablemente comprometida ante la amenaza de monitores y rastreadores de red. Por ejemplo, con el protocolo *Telnet* (ver referencia en Apéndice C.2.a) las credenciales del usuario viajan en texto claro.

Si bien ambos sistemas operativos contemplan mecanismos para cifrar el intercambio de las credenciales del usuario durante el inicio de sesión de trabajo, éstos únicamente son aplicables cuando la comunicación se establece en un ambiente exclusivo de cada sistema operativo. En redes heterogéneas, en cambio, no se contemplan procedimientos similares.

Tanto Windows NT como algunas versiones de UNIX preveen la posibilidad de integrar mecanismos de autenticación adicionales de terceras partes. Existe una amplia gama de alternativas que abarcan desde tarjetas electrónicas hasta verificaciones biométricas.

Por otra parte, Windows NT provee módulos de interface para que programas de aplicación accedan a los servicios criptográficos del sistema operativo. Si bien existen librerías similares disponibles en UNIX, éstas no forman parte de la distribución estándar del sistema operativo.

A diferencia de UNIX, Windows NT integra los mecanismos de autenticación dentro de todas las operaciones de seguridad y la arquitectura del sistema operativo. Las aplicaciones distribuidas utilizan el protocolo de desafío/respuesta de Windows NT para accesos cliente-servidor.

En UNIX, en cambio, las aplicaciones utilizan sus propios esquemas de autenticación. Algunas de ellas usan *Kerberos*[1] para autenticación distribuida. Si bien es una buena solución de seguridad, rara vez está integrada en el sistema operativo, sino que debe ser instalada por separado y configurada en sus aplicaciones.

Si bien la seguridad de los sistemas operativos es muy buena desde el punto de vista de la arquitectura y el diseño, la estructura puede resultar minada por el manejo descuidado de administradores y usuarios. El acceso a copias del archivo de contraseñas y la utilización de palabras clave simples pueden facilitar enormemente el trabajo de los intrusos. Este tipo de cuestiones refuerza la necesidad del establecimiento de medidas de seguridad que involucran a toda la organización.

[1] *Kerberos* es un protocolo de autenticación desarrollado por el Instituto Tecnológico de Massachusetts (M.I.T.) para permitir que las estaciones de trabajo accedan en forma segura a los recursos disponibles en la red.
Además del cliente y el equipo que ejecuta el trabajo efectivamente, el proceso de autenticación involucra la operación de dos servidores. El primero es el servidor de autenticación que se encarga de verificar al usuario en la fase de *login*. Luego, el servidor de emisión de tickets se ocupa de generar la clave de sesión para permitir la comunicación segura entre la estación de trabajo y el recurso al que ésta desea acceder [Tan96]. Dicha comunicación se basa en el intercambio de mensajes que den constancia de la autenticidad del emisor. Asimismo, es posible cifrarlos para lograr la privacidad de su contenido.
Este protocolo ha sufrido varias actualizaciones, siendo V4 la versión más difundida actualmente, aunque ya existe una sucesora. La versión 5 corrige algunas deficiencias de seguridad y ya ha sido publicada como borrador de estándar de Internet (RFC 1510).

Autorización

I. Conceptos Básicos

Una vez que el sistema operativo establece que un usuario es quien dice ser, debe determinar los recursos que están disponibles para dicho usuario y en qué condiciones. Los mecanismos de control de acceso permiten proteger los recursos del sistema de accesos no autorizados. [Mic98]

Típicamente, los sistemas operativos implementan mecanismos de control de acceso mediante los cuales especifican qué usuarios pueden leer, escribir, modificar o copiar determinados objetos del sistema. Sin embargo, la granularidad de dichos mecanismos es variable y depende de las facilidades que provea cada sistema operativo.

Asimismo, el servicio de autorización de acceso puede basarse en distintos modelos:

- Control de Acceso Obligatorio. Requiere una autoridad central que determina qué recursos están disponibles para cada usuario. Los dueños y los creadores no pueden modificar estos controles.

- Control de Acceso Discrecional. Permite que el dueño del objeto defina y cambie la asignación de autorizaciones de acceso sobre dicho recurso.

- Control de Acceso Basado en Roles. Posibilita la asignación de usuarios a roles y la aplicación de reglas de accesos a dichos roles. Este enfoque simplifica la administración de las normas de acceso y ofrece un mayor nivel de consistencia.

El servicio de autorización de acceso provisto por un sistema operativo puede ser evaluado por la granularidad de los controles, la robustez de los mecanismos utilizados para aplicarlos y el nivel de integración con las funciones de administración del sistema. [Mic98]

II. Implementación del Servicio de Autorización

a. Características Comunes

Tanto Windows NT como UNIX presentan un esquema de control de acceso fundamentado en un enfoque discrecional. Sin embargo, también implementan ciertas características del modelo basado en roles mediante el concepto de grupos.

La asignación de autorizaciones de acceso a los objetos se establece mediante la aplicación de permisos específicos en cada sistema operativo. Dichos privilegios pueden ser modificados por los propietarios de los objetos.

b. Particularidades de Windows NT

La seguridad de Windows NT se basa en los conceptos de cuentas de usuario, grupos, derechos y permisos. Los pilares de la seguridad de Windows NT que reúnen esta información son la base de datos del directorio y las listas de control de acceso que tiene cada objeto del sistema. De este modo, el Monitor de Referencia de Seguridad define quiénes pueden utilizar determinados recursos y bajo qué condiciones.

Información de Seguridad de los Objetos

Se pueden proteger todos los recursos con nombre de Windows NT y algunos objetos sin nombre. Un descriptor de seguridad detalla los atributos de seguridad de un objeto. Dicho descriptor consta de cuatro partes [Mic96]:

- Identificador de seguridad del dueño, que indica el usuario o el grupo al que pertenece el objeto.
- Identificador de seguridad del grupo, utilizado sólo por el subsistema POSIX e ignorado por el resto de Windows NT.
- Lista de control de acceso discrecional (ACL), que identifica a los usuarios y los grupos con permisos de acceso específico tanto concedidos como denegados. Las ACL discrecionales son manipuladas por el propietario del recurso.
- Lista de control de acceso del sistema, que controla los mensajes de auditoría que genera el sistema. Las ACL del sistema son controladas por los administradores de la seguridad.

Cada *lista de control de acceso* (ACL) está compuesta por *entradas de control de acceso* (ACE) que especifican los permisos de acceso o auditoría del recurso para un usuario o un grupo. Existen tres tipos de ACE, dos para el control de acceso discrecional y uno para la seguridad del sistema.

Las ACE discrecionales corresponden a las figuras de acceso permitido y acceso denegado. Estas directivas conceden y revocan, respectivamente, el acceso de un usuario o un grupo al recurso en cuestión. En las ACL, las entradas de acceso denegado se ordenan antes que las correspondientes a acceso permitido.

Cabe aclarar que existe una diferencia fundamental entre definir una ACL discrecional vacía para un objeto y no asignar una ACL al mismo. En el primer caso, no se otorgan permisos en forma explícita y por lo tanto se deniega el acceso implícitamente. En cambio, un objeto sin ACL no posee protección alguna y en consecuencia se autoriza cualquier petición de acceso al mismo.

La ACE de seguridad del sistema refleja los requerimientos de auditoría del sistema. Se utiliza para mantener un registro de los eventos de seguridad y genera mensajes de seguridad de auditoría.

Cada ACE contiene una *máscara de acceso*, que define todas las acciones posibles sobre un tipo de objeto en particular.

Los tipos estándar se aplican a todos los objetos:

- *SYNCHRONIZE*, para sincronizar el acceso y permitir a un proceso esperar a un recurso.
- *WRITE_OWNER*, para asignar un propietario de escritura.
- *WRITE_DAC*, para conceder o denegar el acceso de escritura sobre la ACL discrecional.
- *READ_CONTROL*, para conceder o denegar el acceso de lectura al descriptor de seguridad y de propietario.
- *DELETE*, para conceder o denegar el acceso de eliminación.

Los tipos específicos incluyen opciones de acceso que se aplican de forma particular a un tipo de recurso. Cada tipo de objeto puede tener hasta 16 tipos de acceso específicos. Colectivamente, los tipos de acceso específicos de un objeto determinado se denominan máscara de acceso específica. La máscara se determina a la vez que se define el tipo de objeto.

Por ejemplo, los archivos de Windows NT poseen los siguientes tipos de acceso específicos:

- ReadData (lectura de datos).
- WriteData (escritura de datos).
- AppendData (agregado de datos).
- ReadEA (atributos extendidos).
- WriteEA (atributos extendidos).
- Execute (ejecución).
- ReadAttributes (lectura de atributos).
- WriteAttributes (escritura de atributos).

Los tipos genéricos constituyen tipos de acceso relajados utilizados para proteger un recurso. La implementación exacta de estos tipos queda determinada por la aplicación al definir un objeto. Los tipos específicos y estándar se incluyen en los detalles del registro de seguridad del objeto. En lugar de los tipos genéricos, se detallan los tipos específicos y estándar correspondientes.

Cuando se crean objetos dentro de un objeto contenedor, los nuevos recursos heredan por defecto los permisos del padre. En el caso de archivos y directorios, el cambio de permisos de un directorio afecta a ese directorio y a sus archivos, pero no se aplica a los subdirectorios existentes y sus respectivos contenidos, salvo que se lo solicite explícitamente.

Windows NT utiliza descriptores de seguridad para proteger objetos de distintos componentes básicos del sistema operativo, tales como el sistema de archivo de Windows NT, el registro, o las tuberías con nombre (named pipes). Sin embargo, también las aplicaciones desarrolladas para Windows NT pueden aprovechar esta alternativa y asegurar sus objetos privados mediante descriptores de seguridad. Cada aplicación es responsable de mantener las relaciones entre objetos y descriptores correspondientes así como de invocar al monitor de referencia de seguridad para comprobar la validez del acceso. [Mic96]

Validación de Acceso

Cuando un usuario intenta acceder a un recurso, Windows NT compara los datos de seguridad que aparecen en la ficha de acceso del usuario con la información de seguridad del descriptor de seguridad del objeto. [Mic96]

Generalmente, el programa que está ejecutando el usuario genera una máscara de acceso apropiada para el sujeto basándose en el tipo de acceso solicitado. Esta máscara es contrastada con la ACL del objeto. Cabe recordar que todos los tipos de acceso genéricos de la ACL son asignados a tipos estándar y específicos. Cada ACE de la ACL se evalúa de la siguiente forma:

- Se compara el identificador de seguridad de la ACE con el conjunto de identificadores de seguridad contenidos en la ficha de acceso del usuario. Si no existen coincidencias, la ACE se ignora. De resultar positiva la comparación, el procesamiento depende del tipo de ACE.
- Si se deniega el acceso, el sistema comprueba si la máscara de acceso deseada contiene sólo tipos *READ_CONTROL* y *WRITE_DAC*. Si es así, el sistema verifica si el invocante es el propietario del objeto. En caso afirmativo, se concede el acceso.
- Para una ACE de acceso denegado, las acciones de la máscara de acceso ACE se comparan con la máscara de acceso requerida. Cualquier acceso que se encuentre en ambas máscaras, provoca un rechazo. Si no se encuentra, el proceso continúa con la siguiente ACE.
- Si el contenido de la máscara de acceso no coincide completamente al final de la ACL, el acceso se deniega implícitamente.

El administrador del objeto en cuestión mantiene una tabla de procesos que mantiene la información actual de los procesos que están accediendo al objeto. Por tal motivo, la ACL sólo

es chequeada con cada acceso inicial al objeto. Para resolver las solicitudes subsiguientes, basta con verificar los permisos correspondientes almacenados en la tabla de procesos. Dado que dicha información es estática, cualquier modificación de los privilegios no afecta los permisos corrientes y los cambios sólo son considerados en la siguiente apertura del objeto.

Si bien los objetos utilizados por un proceso son liberados automáticamente al finalizar dicho proceso, la liberación explícita de los recursos es una práctica de programación recomendable. Al ser liberado un objeto, el administrador del mismo remueve la entrada de la tabla de procesos.

Por otra parte, uno de los objetivos del modelo de seguridad de Windows NT consiste en asegurar que los programas que ejecuta un usuario no tengan más permisos de acceso a un objeto que los del propio usuario mediante la técnica de *suplantación* [Rus98d]. De este modo, cualquier programa restringe sus privilegios al contexto de seguridad del usuario que lo invoca.

Es importante evaluar las cuentas de servicios para que no posean autorizaciones innecesarias. Una cuenta de especial cuidado es *Sistema*, utilizada por el sistema operativo para ejecutar programas, utilidades y controladores. [She97]

c. Particularidades de UNIX

El servicio de autorización de acceso en UNIX se basa en los permisos de los objetos y los derechos de usuario. La mayoría de los recursos son accesibles desde el sistema de archivos de UNIX y los mecanismos de control de acceso se implementan como parte del mismo.

Información de Seguridad de los Objetos

Cada recurso tiene definidos tres grupos de permisos. Estos definen el nivel de acceso que tendrán los diferentes usuarios a dicho recurso. Los permisos posibles son lectura, escritura, ejecución e inspección, donde ejecución se aplica únicamente a archivos e inspección únicamente a directorios.

Los permisos de los recursos reflejan controles de acceso discrecionales. Se implementan mediante bits de permisos. Estos privilegios tienen una granularidad fija y reflejan las prerrogativas que se aplican al dueño del recurso, al grupo al cual el recurso esté asociado y al resto de los usuarios del sistema. Este conjunto de permisos define una *máscara* asociada a cada objeto.

Validación de Acceso

La máscara de cada objeto es chequeada en el momento en que el usuario intenta acceder por primera vez al mismo. Los datos del usuario (UID y GID) son contrastados con los permisos seteados en la máscara de modo de determinar si dicho usuario puede hacer uso del recurso en cuestión y con qué prerrogativas. En base al resultado de esta comprobación, se permite o deniega el acceso.

También es posible definir permisos para que un usuario pueda acceder a recursos como si fuera otro usuario o como si perteneciera a otro grupo. Mediante este mecanismo es posible hacer que un usuario tenga permisos especiales sobre ciertos recursos.

Listas de Control de Acceso

Algunas versiones de UNIX, en particular las comerciales tales como AIX de IBM, HP-UX de Hewlett-Packard y Solaris de Sun Microsystems, contemplan un sistema de listas de control de acceso que permite mejorar la granularidad de la seguridad de estos sistemas. Si bien este concepto no es nuevo en ambientes UNIX, su implementación en muchos sistemas está todavía en desarrollo.

Mediante la implementación de listas de control de acceso es posible especificar privilegios adicionales a cada objeto del sistema de archivos para un número determinado de individuos. Las listas de control de acceso ofrecen un refinamiento a los permisos estándar de UNIX permitiendo asignar permisos específicos a grupos arbitrarios de usuarios y/o grupos. En algunos casos, también es posible especificar tipos de acceso no contemplados en UNIX, tal como prohibir el acceso a un objeto por parte de grupos de usuarios.

Si bien existen diferencias entre las distintas implementaciones, la lista de control de acceso requiere tres entradas obligatorias correspondientes al dueño, el grupo asociado y el resto de los usuarios. [LeF98]

Opcionalmente, es posible definir una máscara y entradas específicas de permisos para usuarios y grupos determinados, más allá de los correspondientes al dueño y al grupo del objeto. Dicha máscara indica las capacidades máximas permitidas para cualquier usuario distinto del propietario del recurso y para todo grupo incluyendo aquel al que el mismo se encuentra asociado. [LeF98]

En todos los casos, los privilegios se definen mediante el clásico esquema de bits de modo, aunque también resulta posible inhibir el acceso a un determinado recurso al denegar los tres permisos existentes.

Además, es posible definir entradas de valores por defecto en un directorio para el propietario, el grupo del recurso, el resto de los usuarios y la máscara. Estas entradas no son consideradas para evaluar el acceso al directorio en cuestión, sino que se utilizan para construir la lista de control de acceso de los archivos que son agregados al directorio.

Al crear un archivo se genera una lista de control de acceso cuyas entradas responden a la intersección de las prerrogativas otorgadas por el valor de la máscara del usuario, aquellas definidas en las entradas por defecto del directorio padre y los permisos requeridos en el momento de creación. De no especificarse capacidades para un usuario o un grupo en particular, sólo se inicializan los bits de modo con el resultado de la intersección descripta. [LeF98]

Cabe aclarar que el propietario, el grupo y los bits de modo de un recurso pueden ser modificados mediante el manejo de la lista de control de acceso correspondiente. Asimismo, resulta posible alterar o restringir las capacidades de la lista de control de acceso utilizando comandos clásicos de UNIX que involucran la administración de permisos.

El problema con esta facilidad en el entorno UNIX radica en que cada fabricante ha realizado una implementación propia del sistema por lo que no existe un estándar hasta la fecha.

III. Análisis Comparativo

Ambos sistemas operativos implementan mecanismos de control de acceso similares mediante la aplicación de permisos de acceso discrecional sobre los recursos del sistema y de derechos de usuario.

Una diferencia radica en la granularidad de los permisos de los objetos. En líneas generales, UNIX presenta un esquema fijo para todos los recursos, aunque igualmente funcional. Windows NT, en cambio, presenta una gama más variada de prerrogativas aplicables por tipo de objeto.

Por otra parte, Windows NT provee un mecanismo mucho más flexible de asignación de permisos a usuarios individuales y grupos. UNIX no presenta una funcionalidad genérica semejante para otorgar privilegios en forma selectiva. Sin embargo, algunas versiones comerciales son equiparables a Windows NT en esta cuestión mediante la implementación de listas de control de acceso.

Cabe aclarar que el concepto de listas de acceso es considerado en mayor detalle en el próximo capítulo.

En cuanto a Windows NT, hay problemáticas relacionas con el concepto de dominio cuya relevancia no se debe soslayar. Cabe reiterar que los dominios no son barreras seguras entre redes. De hecho, cuando varios dominios coexisten en una misma red física, un usuario de un dominio puede aún ver los sistemas de otro dominio utilizando ciertas órdenes. Es más, este usuario puede llegar a ingresar a algunos recursos de dominios ajenos en determinadas circunstancias, aunque no estén establecidas las relaciones de confianza correspondientes.

Auditoría

I. Conceptos Básicos

El servicio de auditoría permite reunir la información de lo ocurrido en el sistema. Desde el punto de vista de la seguridad, este servicio facilita la reconstrucción de un evento relacionado con la seguridad del sistema de modo de examinar las causas y las consecuencias del mismo. [Mic98]

La importancia de este tipo de registros es indiscutible para determinar violaciones a las políticas de seguridad establecidas en el sistema o para detectar actividades sospechosas. Además, los archivos de auditoría permiten rastrear la fuente de un incidente de seguridad y proveer la evidencia requerida para aplicar medidas correctivas adecuadas.

Existen productos de análisis sofisticados que utilizan esta información generada por el sistema operativo como base de su estudio.

II. Implementación del Servicio de Auditoría

a. Características Comunes

En ambos sistemas se contempla la facilidad de auditar ciertos eventos aunque no siempre relacionados con la cuestión de la seguridad.

b. Particularidades de Windows NT

Windows NT incluye características de auditoría que pueden servir para recopilar información acerca de la utilización del sistema. En particular, incluye características de auditoría para monitorear sucesos relativos a la seguridad del sistema, identificar fallas de seguridad, determinar la extensión y la localización del daño. [Mic97]

Cada objeto del sistema es susceptible de ser auditado contando con la posibilidad de distintos niveles de rastreo. Este tipo de seguimiento puede aplicarse en forma selectiva a un usuario o grupo en particular o a todos los usuarios en general.

Los sucesos a auditar se identifican en el sistema mediante el nombre del módulo origen y por un identificador de suceso. El registro de seguridad del visor de procesos puede mostrar una lista de eventos por categoría y por identificador. (ver detalles en Apéndice C.2.b)

La granularidad de la auditoría se ajusta a las necesidades de la organización. Vale aclarar que la activación de este servicio y la incorporación de eventos a auditar afecta el rendimiento del sistema.

c. Particularidades de UNIX

UNIX ofrece un servicio de auditoría orientado a recopilar información de los sucesos ocurridos en el sistema sin poner especial énfasis en las cuestiones de seguridad (consultar Apéndice C.2.b). Las primeras versiones de UNIX sólo registraban los ingresos y egresos de usuarios al sistema. En los sistemas modernos es posible llevar control muy detallado de todas las acciones realizadas por los usuarios sobre los distintos componentes del sistema.

Hay un conjunto de archivos de auditoría de seguridad que son estándares en los sistemas UNIX. Estos permiten registrar eventos que en su mayor parte tienen que ver con las cuentas de usuario. Entre los sucesos auditados se encuentran el ingreso y egreso exitoso de cada usuario al sistema, el último intento fallido de ingreso de cada usuario, el uso de la facilidad de acceder a otra cuenta (*su*), los reseteos de la máquina y otros.

Asimismo, la mayor parte de los procesos del sistema también generan archivos de auditoría sobre las acciones que realizan. Si bien los mismos son de ayuda relativa en el caso de un evento de seguridad es preciso nombrarlos ya que forman parte de los registros del sistema. Entre los servicios que llevan a cabo este tipo de registro de sus acciones se encuentran el *FTP*, *UUCP*, *cron*, los servicios de impresión, de correo electrónico y de *NTP*.

Algunos sistemas UNIX, en particular las versiones comerciales más difundidas, proveen otras herramientas para auditar eventos del sistema. Estas herramientas apuntan a cumplir con los estándares C2 y son particulares a cada implementación. Las mismas permiten auditar eventos relacionados con los usuarios o con los objetos del sistema. Usualmente, la información se conserva en un servidor remoto y de forma binaria siendo necesaria la utilización de programas especiales para revisar la misma. [Lee97a]

Con relación a los eventos a auditar mediante estas herramientas, es posible registrar cada uno de los comandos de los usuarios durante cada sesión de trabajo. Respecto de los objetos, existe la posibilidad de registrar un gran número de acciones sobre los mismos como ser el acceso y modificación de los mismos y el cambio de la mascara de bits de permisos, entre otros. [Lee97b]

III. Análisis Comparativo

Por omisión, Windows NT presenta servicios más completos en lo que se refiere al rastreo de eventos de seguridad como parte integral del sistema operativo. En particular, cabe destacar la variedad y la granularidad de este servicio.

UNIX, en cambio, no ofrece una capacidad semejante. Existen ciertos archivos de auditoría pero no orientados específicamente a cuestiones relativas a la seguridad. No obstante, los registros generados por los procesos del sistema pueden tener cierta utilidad para contribuir en la reconstrucción de sucesos de seguridad.

Es importante destacar que las distintas versiones UNIX presentan grandes diferencias en cuanto a los servicios de auditoría. En general, los sistemas comerciales proveen herramientas de auditoría particulares a cada implementación con el fin de cumplimentar requerimientos del estándar C2.

Algunas de ellas cuentan con capacidades superiores a las soportadas por Windows NT, en especial en cuanto a la facilidad de seguimiento de las actividades de los usuarios y al almacenamiento de los registros de seguridad.

Tanto en UNIX como Windows NT, un problema que tienen los registros del sistema es que, al estar manejados por procesos del sistema, pueden ser alterados por el supervisor, o cualquiera que haya obtenido acceso como superusuario. Una forma de mejorar sensiblemente esta situación es conservar los archivos de auditoría en un servidor remoto.

La implementación de seguridad de Windows NT y UNIX no distingue las funciones de administrador y auditor. En un sistema ideal, tanto las actividades de los usuarios como de los administradores deberían ser registradas y posteriormente evaluadas por un auditor, sin que ninguno de ellos pudiera modificar estos archivos para cubrir sus huellas.

Actualmente, Windows NT permite seguir el rastro de las acciones de los administradores. Sin embargo, existen diversos modos en que éstos pueden alterar los registros correspondientes para esconder sus actividades, tal como limpiar el archivo de registro o deshabilitar la auditoría. Similarmente, bajo UNIX es posible eliminar o modificar los archivos de registro.

No obstante este tipo de situaciones es altamente sospechoso, es decir, que aunque un administrador intente ocultar sus acciones, generalmente no puede dejar de evidenciar el hecho de esconderlas.

Sistema de Archivos

I. Introducción

Uno de los aspectos más críticos en la seguridad de un sistema es la protección de los archivos. Son éstos los que contienen la información de la configuración del equipo y los datos de los usuarios, además de los programas de los mismos y los que conforman el sistema operativo. Es por ello que el sistema operativo debe proveer mecanismos para evitar que la información contenida en los archivos pueda verse alterada, corrompida, eliminada o robada.

En un principio los medios de almacenamiento hacían que la seguridad de la información contenida en el sistema dependiera en gran medida de la seguridad del centro de cómputos donde se encontraba el equipo. La única forma de transportar información era mediante cintas, listados o tarjetas perforadas. Al bajar el costo y tamaño de los dispositivos de almacenamiento portátiles de gran capacidad y al utilizarse un ambiente de computación distribuido, la posibilidad de acceso a la información por parte de personas no autorizadas aumentó proporcionalmente.

Los principales peligros que pueden afectar a los archivos se detallan a continuación junto con posibles causas de los mismos.

- Eliminación accidental de uno o más archivos. Un usuario o administrador puede borrar un archivo o directorio accidentalmente. En el caso de un administrador, esto puede llevar a la inutilización parcial o total del sistema.
- Corrupción de la información de uno o más archivos. Un fallo de programa puede corromper los archivos de datos del mismo. Un corte en la energía eléctrica puede tener iguales consecuencias.
- Eliminación premeditada de uno o más archivos. Un usuario que consiga acceso de supervisor o un administrador ofuscado puede eliminar premeditadamente archivos con el fin de inutilizar el sistema o borrar las huellas de su paso por el mismo.
- Modificación de uno o más archivos. Un usuario que consiga acceso de supervisor puede modificar archivos de forma de garantizarse el mismo acceso en el futuro. También puede alterar los registros de su paso por el sistema para evitar ser descubierto. Un usuario puede alterar registros de una base de datos con distintos fines y sobreescribir archivos con distinta información.
- Robo de información. Un usuario puede copiar archivos de configuración, datos de los usuarios, archivos de los mismos y/o bases de datos. Esto se puede deber desde espionaje industrial hasta actos de terrorismo.

Si bien la solución a los dos primeros puntos es la de contar con buenas copias de resguardo del sistema, la única solución efectiva para el resto de las amenazas, y en parte para minimizar las primeras, es mediante un sistema de control de accesos al sistema de archivos del sistema operativo.

Para obtener más detalles sobre las características de los sistemas de archivos adoptados por los sistemas operativos estudiados, consultar el Apéndice D.

II. Implementación de la Seguridad del Sistema de Archivos

a. Características Comunes

Muchos de los sistemas operativos multiusuario modernos utilizan un esquema de *dueño*, *grupo* y *resto de los usuarios* para la asignación de permisos de acceso a archivos. Otros utilizan el concepto de *Lista de Control de Acceso*. Tanto Windows NT como UNIX utilizan el primero de los métodos y, además, Windows NT y algunas versiones de UNIX permiten el uso de listas de control de acceso.

Es necesario aclarar la diferencia entre la *lista de control de acceso* y la *lista de capacidades*. Las listas de control de acceso están asociadas a los objetos y determinan todos los dominios que pueden acceder al mismo y de que manera. Las listas de capacidades están asociadas a los usuarios y/o procesos y determinan los objetos que los mismos pueden acceder y de qué forma.

En este sentido el esquema de *dueño*, *grupo* y *resto de los usuarios* para la asignación de permisos de acceso a archivos es una lista de control de accesos muy limitada y es por ello que se remarca la diferencia con el esquema de listas de control de acceso general.

Por otro lado, las acciones que permiten controlar cada uno de ellos varían ligeramente, siendo UNIX más rudimentario en este aspecto que Windows NT. UNIX sólo permite asignar permisos de lectura, escritura y ejecución mientras que Windows NT agrega a estas acciones básicas las de borrado, administración dinámica de permisos y toma de posesión. De todas formas la funcionalidad es similar en ambos casos.

La protección de los archivos que conforman el sistema operativo es un tema de principal importancia en cualquier sistema y, tanto en Windows NT como en UNIX, esta tarea se vuelve compleja por la cantidad y la distribución de dichos archivos. En general los permisos de estos archivos son suficientemente buenos en el momento de la instalación, pero ninguno de los sistemas operativos estudiados proveen una herramienta para verificar que estos permisos no se vean alterados con el tiempo. Hay herramientas desarrolladas por terceros que permiten llevar este tipo de controles.

Ambos sistemas operativos permiten compartir los recursos administrados por el sistema de archivos con otros equipos de la red. Si bien la seguridad de los recursos compartidos debería ser tan buena como la de los locales, esto no se consigue en la práctica por lo que se deben tomar medidas de precaución extraordinarias sobre los sistemas de archivos compartidos.

b. Sistema de Archivos de Windows NT

La administración y la seguridad en el modelo de red paritaria resultan sumamente complejas. Una alternativa es una red de dominios de Windows NT. Cada usuario debe poseer una cuenta de usuario en un controlador de dominio mediante la cual es identificado antes de acceder a recursos en el servidor o en computadoras personales que tengan recursos compartidos (consultar Apéndice D.1). Para implementar el nivel más alto de protección al compartir archivos en Windows NT, se debe utilizar el sistema de archivos propio de Windows NT. [Mic97]

NTFS constituye un sistema de archivos que ofrece más seguridad que los sistemas tales como FAT de DOS. Durante el proceso de instalación de Windows NT, existe la posibilidad de elegir dos tipos de sistemas de archivos, FAT o NTFS, pero si se está considerando la cuestión de la seguridad, se debe optar por NTFS. NTFS permite un número de protecciones sobre archivos y directorios que especifican qué usuarios y grupos pueden acceder a los mismos y con qué prerrogativas (ver detalles en el Apéndice D.1).

Hay otras distinciones importantes entre Windows NT y NTFS cuando se los compara con otros sistemas operativos tales como DOS. Estas características dan mayor seguridad y mejores prestaciones:

- Windows NT arranca por su cuenta y utiliza sus propios servicios.
- Todas las funciones de acceso a disco de bajo nivel se realizan mediante controladores específicos de Windows NT, no controladores almacenados en la BIOS del equipo.

- Al ejecutar un programa DOS bajo Windows NT, el sistema operativo impide que el programa escriba directamente sobre los discos duros.

Administración de Permisos

Tanto en un volumen FAT como en uno NTFS se brinda la facilidad de asignar permisos de compartición. Dichos permisos afectan a todo el recurso compartido, es decir, a todos los archivos y las carpetas que pertenezcan al mismo. Además, sólo inciden sobre los usuarios que operan a través de la red.

En volúmenes NTFS es posible activar permisos en archivos y carpetas que especifican qué grupos y usuarios tienen acceso a los mismos y con qué condiciones [Mic97]. Esta granularidad determina que NTFS sea mucho más versátil que otros sistemas de archivos de sistemas operativos en red en el aspecto de seguridad. Asimismo, NTFS dispone de una gama mucho más amplia de niveles de autorización para acceder a un recurso.

Por otra parte, a diferencia de los permisos de compartición, los privilegios de NTFS se aplican tanto a los usuarios que trabajan en forma local como a los que operan a través de la red. NTFS permite combinar dichos permisos con los específicos de las carpetas y los archivos.

La integración de NTFS con el sistema de seguridad de Windows NT coloca a NTFS como una alternativa adecuada para los volúmenes con requisitos elevados de seguridad. Los privilegios de los archivos y las carpetas se basan en la base de datos del directorio del equipo local o del dominio, permitiendo la asignación a usuarios individuales, grupos o a todos.

En la mayoría de los casos se otorgan privilegios por grupos en vez de a individuos. Esta política hace más sencillo y consistente el modo de administrar la seguridad del sistema de archivos. Dichos permisos determinan el nivel de acceso que los usuarios y los grupos tienen sobre directorios y archivos.

Luego, existen dos aspectos a tener en cuenta en la seguridad del sistema de archivos. El primero se refiere a la restricción de acceso a la información de una computadora local en la que un usuario inicia su sesión de trabajo. La segunda limitación compete a los recursos que se comparten en la red.

De este modo, en volúmenes NTFS es posible definir *permisos locales* y *permisos compartidos* para controlar el acceso a carpetas y archivos. Si un usuario en la red quiere acceder a una carpeta, sus privilegios están basados en la intersección de los permisos locales y compartidos [Mic97]. Los dos niveles de permisos posibilitan determinar distintos grados de acceso para un mismo usuario y un mismo recurso al trabajar en forma local o remota.

Una cuestión para considerar es el tema de la herencia en NTFS. Por defecto, los permisos de una carpeta se aplican a los archivos y subdirectorios contenidos en dicha carpeta. Se pueden hacer cambios específicos en los permisos de los mismos, pero en caso contrario todo usuario posee los mismos permisos sobre la carpeta y su contenido. Estas modificaciones posibilitan tanto restringir como ampliar el tipo de acceso a archivos y subdirectorios de la carpeta en cuestión. [She97]

Los permisos son configurados por el administrador o el propietario del recurso. Aunque rara vez se conceden por sí mismos, NTFS ofrece la posibilidad de configurar *permisos individuales*.

- Lectura (R). Abrir y ver el contenido de un archivo.
- Escritura (W). Cambiar los contenidos de un archivo o crear un nuevo archivo.
- Ejecución (X). Ejecución de un programa o archivo ejecutable.
- Borrado (D). Eliminación de archivos.

- Cambio de Permisos (P). Alteración de los permisos de un directorio o archivo.
- Toma de Posesión (O). Modificación del propietario de un directorio o archivo.

En su lugar, se definen *permisos estándar* que son el resultado de la combinación de los permisos individuales y que están diseñados para proveer un conjunto de privilegios apropiado para la mayoría de los usuarios. Desde luego, se pueden crear permisos especiales particulares en cualquier momento que resulte necesario.

Permisos de Directorios

Permisos Estándar	Permisos Individuales	Descripción	Nuevos Archivos
Sin Acceso	Ninguno	El directorio en cuestión no puede ser accedido ni listado su contenido. Esta figura se impone a cualquier otra prerrogativa.	Ninguno
Listar	Lectura Ejecución	Brinda la posibilidad de acceder al directorio y listar los archivos y carpetas del mismo con el fin de moverse por la estructura de directorio.	Sin especificar
Lectura	Lectura Ejecución	Además de la capacidad de *Listar*, permite visualizar los datos y los atributos de los archivos, ejecutar programas y moverse a cualquier subdirectorio.	Lectura Ejecución
Adición	Escritura Ejecución	El usuario puede agregar un archivo a la carpeta pero no tiene la capacidad de leer o modificar otros archivos.	Sin especificar
Adición y Lectura	Lectura Escritura Ejecución	Además de la capacidad de *Adición*, permite leer y ejecutar archivos sin modificarlos.	Lectura Ejecución
Modificación	Lectura Escritura Ejecución Borrado	Esta prerrogativa ofrece las capacidades del permiso de *Lectura*, el cambio de datos y atributos de archivos, así como también la creación y la eliminación de directorios y archivos.	Lectura Escritura Ejecución Borrado
Control Total	Todo	Incluye todos los permisos individuales con las capacidades de *Modificación* junto con la posibilidad de cambio de permisos para el directorio y su contenido. La modificación de la propiedad puede ocurrir bajo ciertas condiciones.	Todo

Los usuarios pueden obtener permisos para acceder a carpetas y archivos de diversas maneras, ya sea en forma individual, por su pertenencia a uno o varios grupos o por herencia de directorios padre. Dichos privilegios son acumulativos al combinar la asignación de permisos de distintas fuentes. Sin embargo, la figura de *No Acceso* de cualquier fuente niega el acceso a un archivo o una carpeta, sin importar que otros permisos lo concedan.

Permisos de Archivos

Permisos Estándar	Permisos Individuales	Descripción
Sin Acceso	Ninguno	Deniega el acceso al archivo, aún si existe algún otro permiso que lo otorgue.
Lectura	Lectura Ejecución	Esta figura posibilita leer archivos de datos y ejecutar programas además de listar los atributos de los mismos.
Modificación	Lectura Escritura Ejecución Borrado	Ofrece todas las capacidades de *Lectura* como así también la posibilidad de modificar el contenido de archivos y mostrar los permisos y el propietario de éstos.
Control Total	Todo	Además de las prerrogativas de *Modificación*, permite obtener la propiedad de archivos.
Acceso Especial		Esta figura ofrece la facilidad de crear esquemas de acceso especializados que incluyan cualquier combinación de permisos individuales.

A medida que nuevos directorios y archivos se añaden a carpetas, por defecto éstos van heredando los permisos configurados para su directorio padre. Sin embargo, es posible modificar cualquier archivo o subcarpeta para saltar los permisos heredados.

Tal como se mencionó anteriormente, la propiedad de un objeto otorga al usuario el derecho de configurar los permisos del mismo y transferir la propiedad a alguien más. Por defecto, el grupo *Administradores* posee las carpetas y los archivos al instalar Windows NT. Al crear nuevas cuentas de usuario, es recomendable crear carpetas personalizadas para cada uno de ellos. La propiedad permite que el usuario pueda mantener su información privada y disponer del manejo de los permisos.

Si bien es posible impedir que hasta los administradores accedan a recursos de un usuario, éstos cuentan con la capacidad de obtener la propiedad de dichos recursos. Estas acciones deben estar auditadas y vigiladas para que sólo se lleven a cabo en caso de necesidad como por ejemplo cuando una cuenta está bajo sospecha.

Al considerar los permisos por defecto, es necesario conocer los grupos o entidades especiales que poseen estos permisos. Tal como fue definido oportunamente, un grupo es una colección de cuentas de usuario que los administradores pueden manejar añadiendo o eliminando usuarios según sea requerido.

En cambio, las *identidades especiales* representan un conjunto particular de usuarios que tienen acceso al servidor de un modo particular. No es posible agregar usuarios a una identidad especial porque el sistema lo realiza automáticamente.

- Todos. Incluye a todos los usuarios que acceden al servidor. Si se tiene la intención de declarar pública una carpeta, hay que asegurar que este grupo tenga acceso a la misma.
- Interactivo. Abarca a todo aquel que acceda al equipo físicamente. Se pueden otorgar permisos particulares para el acceso a los recursos administrados mientras estén físicamente en el servidor.
- Red. Comprende a todos los usuarios que acceden vía red al servidor.
- Propietario Creador. Esta identidad existe en carpetas y tiene la cualidad de que la persona que crea un archivo o un directorio, se convierte en el propietario creador del mismo. Desde luego, el usuario debe tener el permiso para generar una nueva carpeta o agregar un archivo. Esta característica permite otorgar el control automático a los usuarios de los directorios creados.
- Sistema. Esta es la identidad del sistema operativo que le posibilita el acceso a los recursos del mismo modo en que lo hace una cuenta de usuario.

Al instalar el sistema operativo, se asignan un número de permisos por defecto. Algunos de estos permisos son adecuados para la mayoría de los entornos. No obstante, otros están diseñados para ámbitos donde la seguridad no es demasiado estricta, por lo cual puede ser necesario modificarlos. [She97]

Indudablemente resulta fácil perder el rastro de los permisos configurados en la estructura de todo el directorio de un servidor. Es conveniente utilizar algún utilitario para visualizar de una forma más amigable un informe de los permisos definidos. Especial atención merece el grupo *Todos* que incluye a todos los que acceden al sistema. Por defecto, este grupo sorprendentemente tiene amplias capacidades en los directorios raíz y del sistema.

Los archivos pueden obtener permisos inapropiados en el caso de ser movidos o copiados. Cuando un archivo se copia, hereda los permisos del directorio donde es copiado. En caso de ser movido, mantiene los privilegios originales. Esto puede generar situaciones de empobrecimiento o enriquecimiento en cuanto a los accesos autorizados para un archivo que es importante considerar.

Compartición de Directorios y Archivos

La compartición de recursos en entornos Windows está gestionada por el protocolo servicio de bloques de mensajes (Server Message Blocks, SMB) (consultar Apéndice D.1). SMB se

implementa en componentes de servidor y estaciones de trabajo en sistemas Windows NT. Provee servicios redirector que permiten a un cliente localizar archivos en otras computadoras de la red ejecutando SMB para abrir, leer, escribir y borrar dichos archivos. El mayor riesgo con la compartición de recursos usando SMB es potenciar el número de usuarios sin autorización que accedan a los recursos de la red. [She97]

Los recursos compartidos a través de SMB sólo tienen cuatro tipos básicos de acceso a saber: *Sin Acceso*, *Lectura*, *Modificación* y *Control Total*. Dada la simplicidad de esta clasificación, es recomendable combinar estos permisos con los que los usuarios o grupos poseen sobre el objeto compartido [Mic97]. Vale aclarar que al compartir una carpeta, toda la estructura que pende de ella queda compartida con los mismos permisos a menos que específicamente se modifiquen los privilegios o se deje de compartir algún subdirectorio.

En el Service Pack 3 se incluyeron algunas mejoras para el protocolo de compartición de archivos basado en SMB al implementar la firma de mensajes en los paquetes SMB. La introducción de este concepto permite la mutua autenticación de los extremos cliente y servidor. Sin embargo, esta facilidad cobra sentido exclusivamente en entornos de servidores y estaciones de trabajo NT que soporten esta característica. [Dai98b]

Auditoría

El sistema de auditoría de Windows NT permite monitorear sucesos relacionados con el sistema de archivos de modo de verificar las acciones realizadas por usuarios autorizados como no autorizados. La auditoría del sistema de archivos es muy fina porque ofrece la capacidad de rastrear cómo un usuario específico utiliza un directorio o un archivo determinados. Esta granularidad contribuye a minimizar los sucesos auditados y reduce el uso de recursos del sistema involucrado en dicho proceso.

Los cambios de auditoría se aplican a directorios, subdirectorios y archivos, ya sea en el directorio únicamente, en éste y sus archivos, solamente en el árbol de directorios, o en la estructura completa de subdirectorios y los contenidos de cada uno de ellos. Para ver estos sucesos del sistema de archivos, basta con acceder al *Visor de Eventos* y seleccionar la página *Seguridad*. Es posible guardar la lista de eventos en archivos externos para tomarlos con otra aplicación.

Cabe aclarar que el sistema de auditoría sólo indica qué cuentas de usuarios fueron utilizadas para los eventos auditados. Si alguien se ha apropiado de una cuenta sin permiso, podría pensarse erróneamente que el dueño legal es el responsable de las actividades no autorizadas. Además, únicamente los archivos y los directorios en particiones NTFS pueden ser auditadas.

c. Sistema de Archivos de UNIX

La seguridad de los recursos del sistema de archivos de UNIX se basa originalmente en permisos que se asignan individualmente a cada objeto. Cada objeto tiene un dueño y un grupo al que pertenece y, de esta forma, es posible asignar permisos específicos para el dueño, el grupo y el resto de los usuarios. Este es un esquema de ACL restringido.

Los procesos que intentan acceder a los objetos lo hacen a través del núcleo del sistema que controla, según que usuario sea el dueño del proceso, los derechos de acceso que tiene sobre el objeto accedido. De esta forma el administrador del sistema de archivos del núcleo es el encargado de llevar control de la seguridad del mismo.

En las versiones más modernas de UNIX se está haciendo cada vez más común la implementación de listas de control de acceso a la manera usual para otorgar una mayor granularidad a la asignación de permisos sobre los objetos. De esta manera dichas versiones proveen una mejor implementación del esquema de seguridad del sistema de archivos. Es posible que en un futuro próximo todas las variedades de UNIX de uso común provean algún tipo de facilidad de ACL.

Administración de Permisos: Bits de Modo

Los bits de permisos o de modo, determinan el tipo de acceso que tendrán los distintos usuarios del sistema a los objetos almacenados en el sistema de archivos. Los permisos se asignan a tres grupos de usuarios, el dueño del objeto, todos los usuarios que están en el grupo del objeto y todos los demás usuarios del sistema con excepción del supervisor.

Los bits de modo especifican el modo de acceso que cada uno de estos grupos tendrá sobre el objeto. Este modo puede ser lectura, escritura y ejecución. Un usuario que tenga permiso de lectura sobre un objeto, podrá leer información de él, si tiene permiso de escritura puede sobreescribir el objeto, agregar datos al objeto o eliminarlos. El permiso de ejecución tiene sentido para programas y es posible que un programa tenga permisos de ejecución y no de lectura, aunque esto no se aplica a sistema de archivos exportados a otros equipos.

En realidad los usuarios y programas no acceden al sistema de archivos, son los procesos los que lo hacen. Cada proceso tiene asignado un dueño y un grupo. Cuando un proceso trata de acceder a un objeto, si el dueño del mismo es igual al del proceso entonces se utilizan los permisos del dueño, si el dueño es distinto pero el grupo coincide, entonces se utilizan los permisos del grupo y en caso contrario se utilizan los permisos para el resto. [Win93a]

Los bits de modo no se aplican a los enlaces, pero sí a los dispositivos, *sockets* y colas (Named Pipes) de la misma forma que a los archivos comunes. Si se tiene permiso de lectura en un dispositivo, es posible leer información del mismo y si se tiene permiso de escritura en una cola, es posible enviar datos a la misma. Cabe aclarar que sólo el dueño de un objeto o el superusuario pueden cambiar los bits de modo del mismo. [LeF98]

En el caso de los directorios, al ser estos tratados como archivos comunes, tienen asignados los mismos bits de modo que el resto de los archivos pero, además, tienen una marca que los diferencia y esto hace que los permisos tengan una interpretación ligeramente diferente que el resto de los objetos.

El permiso de lectura de un directorio hace que sea posible listar el contenido del mismo. El permiso de escritura garantiza la creación, renombrado y remoción de entradas en este y el permiso de ejecución en un directorio permite hacer de este el directorio corriente y abrir archivos dentro del mismo.

Los permisos con los que un objeto es creado en el sistema de archivos están determinados por una máscara que cada proceso del sistema posee y que es heredada del proceso que lo lanzó. El usuario puede definir una máscara para el uso de los procesos que él ejecute. Habitualmente hay una máscara por omisión en el sistema que es utilizada por todos los procesos salvo que se especifique una nueva máscara. [LeF98]

Es posible que se modifiquen los permisos de un archivo al copiarlo mientras que al moverlo esto no ocurre. Es preciso tener especial cuidado al realizar copias de archivos que contengan

información sensitiva ya que se corre el riesgo de brindar más derechos de acceso que los del archivo original.

Set User ID y Set Group ID

A veces es necesario que un proceso no privilegiado pueda realizar tareas como si fuera un usuario privilegiado. Por supuesto que otorgarle derechos de administrador a un proceso de usuario trae aparejado un gran problema de seguridad. UNIX provee un mecanismo que permite a los procesos que ejecutan bajo un usuario y grupo, actuar con los privilegios de otro usuario o grupo durante su ejecución. Estos son llamados *set user ID* o SUID y *set group ID* o SGID. (ver Apéndice D.2)

El programa SUID cambia su número de usuario efectivo al del dueño del mismo, en lugar de aquel del usuario que ejecuta el programa. De la misma forma que un programa SGID cambia su número de grupo efectivo al del grupo del mismo, en lugar de aquel del usuario que ejecuta el programa.

Archivos de Acceso a Dispositivos

Salvo por algunas excepciones puntuales, los archivos de acceso a dispositivos no deben tener permiso de lectura o escritura para ningún usuario salvo el dueño, que en la mayoría de los casos es el supervisor. También es importante verificar la existencia de archivos de acceso a dispositivos fuera de los directorios donde se supone que deban estar. (consultar Apéndice D.2)

Listas de Control de Accesos (ACL)

Algunas versiones de UNIX soportan listas de control de accesos como una extensión a los permisos básicos de UNIX ya descriptos, entre ellas las versiones comerciales, el entorno de programación distribuida DCE también tiene una implementación de ACLs. Además, se espera que versiones futuras de Linux también lo provean. La idea de las listas de control de acceso no es nueva pero su implementación en muchos sistemas UNIX está todavía en desarrollo.

Mediante las listas de control de accesos es posible especificar derechos adicionales a cada objeto del sistema de archivos para un número determinado de individuos. También es posible asignar diferentes permisos a miembros de distintos grupos. El problema es que cada fabricante ha realizado una implementación propia del sistema por lo que no existe un estándar hasta la fecha. Esto es de especial importancia cuando se exportan sistemas de archivos a otros servidores de distintos proveedores.

Las listas de control de accesos ofrecen un refinamiento a los permisos estándar de UNIX permitiendo asignar permisos específicos a grupos arbitrarios de usuarios y/o grupos. También es posible especificar tipos de acceso no contemplados en UNIX como ser prohibir el acceso a un objeto por parte de grupos de los usuarios.

La lista de control de acceso requiere tres entradas obligatorias correspondientes al dueño, el grupo asociado y el resto de los usuarios. Es posible definir una máscara y entradas específicas de permisos para usuarios y grupos determinados, más allá de los correspondientes al dueño y al grupo del objeto. Dicha máscara indica las capacidades máximas permitidas para cualquier usuario distinto del propietario del recurso y para todo grupo incluyendo aquel al que el mismo se encuentra asociado. [LeF98]

En todos los casos, los privilegios se definen mediante el clásico esquema de bits de modo, aunque también resulta posible inhibir el acceso a un determinado recurso al denegar los tres permisos existentes. Al crear un archivo se genera una lista de control de acceso cuyas entradas responden a la intersección de las prerrogativas otorgadas por el valor de la máscara del usuario, aquellas definidas en las entradas por defecto del directorio padre y los permisos requeridos en el momento de creación.

Seguridad de Recursos Compartidos vía NFS

La seguridad de un sistema de archivos tiene dos aspectos: controlar el acceso y las operaciones sobre los archivos y limitar la exposición del contenido de los archivos. Controlar el acceso a archivos remotos involucra hacer una correspondencia entre la semántica de las operaciones de archivos de UNIX y del sistema NFS (ver Apéndice D.2) para que ciertas operaciones sean inhabilitadas si el usuario remoto no es autenticado.

Para prevenir el otorgamiento de privilegios de supervisor por sobre la red, restricciones adicionales deben ser impuestas para los accesos realizados por el supervisor. En redes donde la seguridad es imperativa, cada pedido de operación NFS deberá tener su correspondiente credencial.

Limitar la exposición del contenido de los archivos que son compartidos en la red es aún más complicado e involucra usualmente el cifrado del contenido del archivo. La aplicación cliente podría cifrar sus archivos de datos de forma que en el momento de accederlo a través de la red, no sea posible ver el contenido. Pero si un archivo está guardado en el servidor en forma legible, las operaciones NFS de leer o grabar el mismo contendrá este texto legible.

Cada pedido al servidor NFS contiene un conjunto de credenciales de usuario incluyendo el número de identificación y la lista de grupos a los que el mismo pertenece. Las credenciales son las mismas que las utilizadas para acceder a un sistema de archivos local. [Ste91]

En el servidor NFS, estas credenciales son utilizadas para realizar los chequeos de permisos que son parte del control de accesos de UNIX. Hay tres puntos en los que las credenciales NFS pueden diferir de la estructura de credenciales local: el usuario es el supervisor, el usuario pertenece a muchos grupos y la falta de credenciales en el pedido.

Al supervisor no se le otorgan los permisos normales en sistemas de archivos montados por NFS. Esto es así para evitar que el supervisor de un equipo pueda acceder con permisos de supervisor a los archivos de un sistema remoto. El supervisor de una máquina no tiene porque poder modificar los archivos de un servidor sobre los que no tiene acceso.

Habitualmente se hace corresponder al usuario supervisor del sistema cliente con un usuario del servidor que tiene muy pocos privilegios, por ejemplo el usuario *nobody*, pudiendo hacer esta correspondencia con cualquier usuario.

El problema del caso en que un usuario pertenezca a muchos grupos se da entre diferentes implementaciones del protocolo, cuando el cliente permite que el usuario pertenezca a más grupos de los que el servidor soporta y un usuario del cliente con más grupos de los aceptados por el servidor intenta un acceso remoto.

En este caso el usuario no es autenticado y la conexión es rechazada con un mensaje de error. La única solución es restringir el número de grupos a los que un usuario puede pertenecer al límite del servidor.

Para los requerimientos que no tienen las correspondientes credenciales, se hace una correspondencia con un usuario válido en el sistema que tenga pocos privilegios, de la misma manera como se hace con los supervisores de sistemas clientes. También es posible hacer que el sistema NFS rechace los pedidos que no tengan su credencial.

Además de proteger el acceso a los sistemas de archivos por los supervisores de los clientes, algunos sistemas de archivos pueden requerir protección de acceso desde ciertos servidores. Es posible y recomendable especificar una lista de clientes que pueden acceder a los sistemas de archivos exportados por el servidor NFS. [Col97]

También es posible exportar los sistemas de archivos en modo sólo lectura de forma que no puedan ser modificados por los clientes que lo monten y limitar el acceso a los subdirectorios de un directorio exportado. [Col97]

Por último, el servidor NFS puede realizar un monitoreo de los *ports* de origen de los pedidos de NFS. Como los pedidos de NFS válidos deben provenir del núcleo del cliente, los *ports* de comunicaciones utilizados por el cliente deben ser privilegiados. Como los paquetes de red contienen los números de *ports* origen y destino, el servidor NFS puede controlar que el *port* origen sea un *port* privilegiado. Sin embargo, hay algunas implementaciones de clientes NFS que no funcionan si el monitoreo de *ports* está habilitado. [Col97]

NFS Seguro sobre RPC Seguro

El protocolo RPC sobre el que NFS está construido, no asegura que las credenciales que son transmitidas correspondan realmente a quién dicen pertenecer ya que carece de un sistema de autenticación. El RPC seguro agrega mecanismos de validación de credenciales al sistema RPC estándar. Cuando NFS se utiliza en combinación con RPC seguro se obtiene lo que se denomina NFS seguro. [Ste91]

RPC seguro utiliza una combinación de cifrado de clave pública (cambio de claves exponencial) y de clave privada (DES) para validar las credenciales. Para establecer una comunicación, el cliente y el servidor eligen una clave de sesión cifrada utilizando un cifrado de clave (asimétrica) pública, la que es luego utilizada para descifrar una clave simétrica (DES) que es la utilizada en el resto de la sesión. Una vez que la comunicación RPC entre los equipos se realiza de forma segura, los datos que circulen por la red no serán expuestos.

Cuando un usuario desea acceder a un sistema de archivos remoto utilizando NFS seguro, el mismo debe generar una clave válida para que el servidor pueda autentificarlo. En caso en que el usuario no pueda proveer dicha clave, se lo hace corresponder con el usuario anónimo (*nobody*).

Para que los usuarios puedan utilizar el sistema NFS seguro, se le debe asignar a los mismos y también a los equipos clientes, pares de claves públicas y privadas para que les sea posible entablar las conexiones con el servidor.

Un inconveniente de este esquema es que requiere del uso de NIS para guardar los pares de claves. En el mismo se mantiene un mapa que contiene las claves secretas de cada usuario, cifradas con su propia palabra clave, y en el caso de los equipos, las claves se cifran con la palabra clave del supervisor.

III. ANÁLISIS COMPARATIVO

Ambos sistemas operativos basan la seguridad del sistema de archivos en permisos de acceso sobre los objetos. En Windows NT este esquema está implementado mediante listas de control de acceso mientras que en UNIX se utiliza una máscara de bits de permisos

asociada a cada objeto, aunque hay versiones de UNIX que agregan la facilidad de las listas de control de acceso.

La variedad de permisos que es posible asignar en Windows NT es mayor que en UNIX, aunque esto no signifique que este último sistema se vea limitado ya que la mayoría de los permisos no existentes directamente en UNIX, pueden ser construidos sobre la base de combinaciones de los permisos posibles. De todas formas el esquema de permisos de Windows NT permite un mayor grado de granularidad sólo equiparable mediante el uso de ACLs en las versiones de UNIX que las soportan.

Es necesario aclarar que un problema potencial de seguridad se produce al borrar un archivo. Ninguno de los sistemas operativos estudiados elimina la información del disco al momento de borrar el archivo, y no existen garantías que el espacio ocupado por el archivo sea sobrescrito con nueva información.

Este problema es aún mayor al utilizar las herramientas gráficas de manejo de archivos ya que, normalmente, las mismas guardan una copia del archivo eliminado en caso de una posterior restauración. En este caso la información no será sobrescrita de ninguna forma y, además, está siendo referenciada dentro del sistema de archivos.

En un ambiente donde se requiere un cierto nivel de seguridad de la información es necesario que el espacio ocupado por un objeto eliminado sea sobreescrito con información aleatoria o carente de sentido de forma de garantizar que la información no sea recuperable. Para esto existen herramientas de terceras partes.

Tanto en Windows NT como en UNIX los permisos de los archivos están incluidos en la información del mismo. Esto hace que al moverlos, la información de permisos no se vea alterada ya que es movida de la misma forma. En el caso de la copia de un archivo, un nuevo archivo es creado con los permisos del directorio (en caso de Windows NT) o de la máscara del usuario (en caso de UNIX). Estos permisos podrían ser menos restrictivos que los del archivo fuente originando de esta forma una debilitación en la seguridad de la información.

El envío de información por una red implica un cierto riesgo a la exposición de los datos. En general, si un archivo o parte de él contiene información que pueda comprometer a la seguridad del sistema, entonces dicho archivo no deberá ser colocado en un sistema de archivos exportado.

En Windows NT, los permisos de compartición se intersecan a los del sistema de archivos NTFS. El problema es que no es posible limitar la exportación de los sistemas de archivos de ninguna forma. Una vez que un directorio es compartido mediante SMB, el mismo es visible desde cualquier lugar de la red salvo que se utilicen filtros de paquetes o Firewalls.

El protocolo NFS de UNIX provee permisos de exportación para limitar el acceso a la información compartida. Estos permisos se agregan a los del sistema de archivos. Una característica fundamental de este protocolo es que permite limitar la exportación de los sistemas de archivos a un grupo de equipos, pudiendo de esta forma, limitar la exposición de los datos a un grupo de clientes confiables.

Por otra parte SMB sólo provee un mecanismo de firma que puede ser utilizado en la identificación de los clientes mientras que NFS provee un mecanismo de cifrado de toda la comunicación a través de la red.

Ningún sistema operativo puede prevenir el robo de la información en caso en que un atacante consiga acceder al equipo y robar el mismo o sus discos. Una vez que el disco es robado puede ser conectado a otro equipo similar donde el atacante sea el supervisor y su información estará expuesta.

Es por ello que se recomienda que los servidores que contengan información sensitiva se encuentren ubicados en sitios vigilados y de difícil acceso al común de la gente. Por otro lado hay herramientas de terceras partes que permiten cifrar la información contenida en los discos, pero éstas no son muy difundidas por los problemas que traen aparejado al tener que reemplazar parte del núcleo del sistema operativo.

Administración de Procesos

I. Introducción

En todo sistema operativo multitarea hay más de un proceso en ejecución en un momento determinado. Este proceso pertenece a un usuario, ya sea un usuario administrativo o un usuario común y ocupa un espacio de memoria.

El sistema operativo multitarea debe proveer básicamente protección del espacio de memoria de cada proceso, esto es que ningún otro proceso o usuario pueda acceder al espacio de memoria del mismo y también prevenir que un usuario no autorizado pueda controlar el comportamiento de un proceso que no le pertenece.

Un problema grave se origina en los defectos de programación de los procesos. Casi ningún programa está libre de este tipo de fallas. Si bien no es la intención de este trabajo disertar sobre las fallas de programación, dado que el sistema operativo está conformado por un conjunto de programas y las fallas de estos pueden afectar y de hecho afectan a la integridad del equipo, se tratará este tema limitándolo a fallas de programas del sistema.

Debido a que todas las acciones realizadas por la computadora, son en realidad llevadas a cabo por la ejecución de procesos sobre la misma, son los procesos los que controlan la seguridad del equipo. Ellos son los que pueden o no permitir el acceso al equipo, a la información y al uso de los servicios, los que auditan el sistema y todas las funciones de las que es capaz el sistema. Es por ello que la importancia de la seguridad de los procesos no debe ser subestimada.

Para obtener más detalles sobre las características de la administración de procesos de los sistemas operativos considerados consultar el apéndice E.

II. Implementación de la Seguridad de Procesos

Los conceptos tratados en este capítulo son sumamente generales para cualquier sistema operativo multitarea y las implementaciones estudiadas se ciñen a las características generales. De esta forma las características comunes entre ellos serán dominantes mientras que las particularidades de cada uno serán limitadas y distintivas, siendo a veces de difícil comparación.

a. Características Comunes

Tanto Windows NT como todas las versiones de UNIX proveen protección del espacio de memoria de cada proceso en ejecución. Esto se realiza de la manera habitual, utilizando registros base y desplazamiento para cada proceso y mapas de ocupación de memoria. En caso en que un proceso intente acceder a una porción de memoria fuera de su espacio de direcciones válido, tanto Windows NT como UNIX cancelan el proceso.

Ambos sistemas operativos controlan que el envío de señales a un proceso provenga únicamente del dueño del mismo, evitando que un usuario no autorizado pueda controlar el comportamiento de un proceso que no le pertenece. Además, el dueño es el único autorizado para cambiar parámetros de la ejecución de un proceso como la prioridad del mismo. En todo caso el administrador del sistema tiene la capacidad de enviar señales y modificar parámetros de ejecución de cualquier proceso en memoria.

Como casi todos los programas, los procesos del sistema operativo no están libres de fallas. El peligro se hace aquí presente ya que una falla en un programa del sistema puede provocar

desde la caída de un servicio o de todo el equipo hasta permitir acceso irrestricto al sistema por parte de un usuario no autorizado.

Si bien los errores de programación de cada sistema operativo son diametralmente disímiles, hay una serie de fallas que suelen ser consistentes en los sistemas operativos (en red) y que se deben a deficiencias en los controles de errores en el manejo de *buffers* de comunicaciones. Estas fallas se dan tanto en las distintas versiones de UNIX como en Windows NT y afectan principalmente de dos formas, inhabilitando algún servicio del sistema o permitiendo a un atacante acceso irrestricto al sistema.

Otro punto importante a tener en cuenta es la posibilidad de un usuario de impostarse como otro en el momento de la ejecución de un comando. Si bien esto es necesario para algunas de las funciones del sistema, las implicancias de la seguridad de esta facilidad son notorias, en particular porque habitualmente el usuario impostado es el administrador.

b. Seguridad de Procesos en Windows NT

Procesos y *Threads*

Un *proceso* es un programa de aplicación o una parte modular de un programa. El sistema operativo define al proceso mediante un espacio de direcciones, un conjunto de objetos y de subprocesos que se ejecutan en el contexto del proceso.

El *objeto proceso* especifica la correspondencia de direcciones virtuales para el subproceso y acumula el tiempo de ejecución del subproceso. Así, posee un puntero al mapa de direcciones y mantiene una lista de listas de subprocesos que pertenecen al proceso y de aquellos preparados cuando el proceso no está activo, incluyendo la información sobre tiempos de ejecución, prioridades y afinidades respectivas. [Mic96]

Un *subproceso* o *thread* es la entidad básica planificable del sistema. Tiene su propio conjunto de registros, su propia pila de núcleo, un bloque de entorno de subproceso y una pila de usuario en el espacio de direcciones de su proceso.

El administrador de procesos es el componente del sistema operativo encargado de generar y eliminar los procesos realizando el seguimiento de los objetos proceso y los objetos thread. Ofrece un conjunto estándar de servicios para crear y utilizar *threads* y procesos en un entorno particular.

El administrador de procesos interactúa con el monitor de referencia de seguridad y el administrador de memoria virtual para proporcionar protección entre procesos. Cada proceso tiene asignado a un identificador de acceso de seguridad, utilizado por las rutinas de validación de acceso de Windows NT cuando los *threads* de un proceso hacen referencia a objetos protegidos. [Mic96]

Para más detalles sobre la administración de procesos en Windows NT ver el Apéndice E.1.

Impostación de Usuarios: Sujeto y Suplantación

Un *sujeto* es la combinación de un programa que está actuando en nombre de un usuario y de la *ficha de control de acceso*. Windows NT utiliza esta figura para todas las tareas relacionadas con el control de derechos de un proceso en ejecución. Cuando un programa trabaja en nombre de un usuario, el mismo ejecuta bajo el contexto de seguridad del usuario.

Cuando un proceso hace una llamada a un objeto de un subsistema protegido, se utiliza la ficha de control de acceso del sujeto para determinar quién ha realizado la llamada y definir si el mismo tiene derechos suficientes para ejecutar la acción requerida. [Mic97]

Windows NT permite que un proceso adquiera los atributos de seguridad de otro proceso mediante la técnica de *suplantación* [Rus98d]. Un proceso servidor puede, de esta forma, completar una tarea que involucra recursos sobre los que el servidor no tiene privilegios y que es requerida por un cliente.

Cuenta *Sistema*

La cuenta *Sistema* no es una cuenta de usuario, sino una cuenta que emplea el sistema operativo para ejecutar programas, utilidades y controladores. Tiene poder ilimitado por lo cual resulta imprescindible mantener un estricto control sobre la misma para impedir que programas destructivos tales como troyanos puedan ser ejecutados bajo su figura. También podría crear y cambiar cuentas de usuario o realizar otras actividades de violación de los sistemas. [She97]

Esta cuenta permite ejecutar servicios sin la necesidad de iniciar una sesión de trabajo administrativa. Es similar a la cuenta *Administrador*, si bien es únicamente utilizada por el sistema operativo y los servicios que se ejecutan bajo Windows NT. Al instalar un nuevo servicio, es recomendable ejecutar el mismo desde una cuenta especial con la menor cantidad de prerrogativas posible necesaria para su operación. Este procedimiento evita que la eventual infección del servicio destruya otras partes del sistema.

Ejecución Planificada de Tareas: Servicio *Schedule* (Comando AT)

El servicio *Schedule*, también conocido como comando AT es utilizado para planificar tareas a ejecutar automáticamente en el momento. Por defecto sólo los administradores tienen el derecho de hacer uso de este servicio. También es posible otorgar a los operadores del sistema esta capacidad. Sin embargo, no existe modo de permitir a otros usuarios ejecutar el comando AT.

La tarea en cuestión corre en el contexto del servicio *Schedule* que generalmente implica el contexto del sistema operativo mismo. Por tal motivo, desde el punto de vista de seguridad no es recomendable hacer uso de este servicio en ámbitos con elevados requerimientos de seguridad. Asimismo, es conveniente restringir el acceso a la clave del Registro que referencia a este servicio a aquellos que están autorizados a hacer uso del mismo.

Mapeo de Archivos y Memoria Compartida

Una característica importante del administrador de memoria de Windows NT consiste permitir a los procesos acceder a archivos en su propio espacio de direccionamiento virtual [Rus98a]. En un primer paso el proceso crea un objeto que describe el archivo que el administrador de memoria puede mapear y, a continuación, el programa hace corresponder todo o parte del archivo en el espacio de direcciones.

El administrador de memoria trabaja en forma transparente con el sistema de archivos para asegurar la integridad del archivo en cuestión volcando las eventuales modificaciones que pudieran ocurrir.

Windows NT utiliza extensivamente la capacidad de mapeo de archivos. Cuando se lanza una aplicación, el administrador de procesos hace corresponder una vista de la imagen de la aplicación al espacio de direcciones del proceso de la aplicación. Luego transfiere el control al punto de entrada de la imagen. A medida que se ejecuta la aplicación, cualquier página referenciada por primera vez provoca un fallo de página que resulta en la lectura de la página correspondiente de la aplicación desde el archivo en disco por parte del administrador de memoria.

Asimismo, Windows NT utiliza una variante de mapeo de archivos para compartir memoria entre procesos de forma que los procesos pueden comunicarse entre sí mediante este método.

c. Seguridad de Procesos en UNIX

Procesos y *Threads*

Un proceso en UNIX es un programa que está en ejecución. Los procesos llevan usualmente el nombre del archivo que guarda el programa en el sistema de archivos. Entre los datos que el núcleo del sistema maneja de cada proceso en ejecución, los que tienen relación con la seguridad de los mismos son:

- Número de identificación del proceso (Process ID o PID). Número de identificación unívoco del proceso dentro del sistema.

- Número de identificación de usuario real y efectivo. Cada proceso UNIX tiene dos identificadores de usuario (UID), un UID real y un UID efectivo. El UID real es, normalmente, el UID del usuario que invocó el programa. El UID efectivo identifica los privilegios del programa en ejecución. Normalmente estos números coinciden pero en el caso de los programas "Set user ID" el UID efectivo es modificado.

- Número de identificación de grupo real y efectivo. Cada proceso UNIX tiene dos identificadores de grupo (GID), un GID real y un GID efectivo. Normalmente estos números coinciden pero en el caso de los programas "Set group ID" el GID efectivo es modificado.

- Prioridad del proceso. Valor que se utiliza para calcular la prioridad de cada proceso en el momento de la elección del siguiente proceso a hacer uso del procesador.

- Terminal que lo controla: este valor es utilizado para el control del envío de señales.

- Tamaño del proceso en memoria y espacio de direccionamiento. Tamaño del proceso indicado en páginas de memoria y espacio de direcciones que puede utilizar.

Finalmente, las versiones más modernas de UNIX proveen soporte de *threads*. En este sentido es preciso indicar que cada *thread* es tratada como un proceso independiente que comparte algunos de los valores nombrados con los demás *threads* que componen el proceso. (consultar apéndice E.2)

Impostación de Usuarios y Grupos: Set User ID y Set Group ID

A veces es necesario que un proceso no privilegiado pueda realizar tareas con los derechos de un usuario privilegiado. Por supuesto que otorgarle derechos de administrador a un proceso de usuario trae aparejado un gran problema de seguridad. UNIX provee un mecanismo que permite a los procesos que ejecutan bajo un usuario y grupo, actuar con los privilegios de otro usuario o grupo durante su ejecución. Estos son llamados *set user ID* o SUID y *set group ID* o SGID.

El programa SUID cambia su número de usuario efectivo al del dueño del mismo, en lugar de aquel del usuario que ejecuta el programa. De la misma forma que un programa SGID cambia su número de grupo efectivo al del grupo del mismo, en lugar de aquel del usuario que ejecuta el programa.

Seguridad de Archivos SUID y SGID

Cualquier programa puede ser SUID o SGID y esto produce varios problemas de seguridad. En principio cualquier usuario que ejecute un programa SUID tiene los mismos privilegios que el dueño del programa y, a la vez, la mayor parte de los programas SUID o SGID de un sistema pertenecen al administrador. Es por ello que el riesgo es mayor.

Cualquier usuario que sea capaz de interrumpir la ejecución de un programa SIUD y obtener una línea de comandos, habrá obtenido acceso irrestricto al sistema. También, cualquier usuario que sea capaz de copiar un programa interactivo (shell, editor, etcétera) con permisos de supervisor y asignarle permisos de SUID, tendrá acceso irrestricto todas las veces que lo desee. [Gar96]

Los permisos de SUID y SGID no representan un peligro "per se", ya que un programa compilado sólo puede realizar las funciones para las que fue programado. Sin embargo, estos deben ser cuidadosamente protegidos ya que muchas fallas de seguridad han sido descubiertas por usuarios que encontraron la forma de hacer que un programa se comporte de forma diferente a lo previsto.

Por la forma en que UNIX maneja la ejecución de scripts (archivos ejecutables con comandos UNIX) es sumamente riesgoso incluir scripts con permisos SUID o SGID ya que es relativamente sencillo interrumpir el proceso de carga del script y obtener accesos de supervisor. Es por ello que algunos de los sistemas UNIX modernos, ignoran explícitamente los permisos de SIUD o SGID en scripts.

Por otro lado, es poco deseado que un programa que es SUID en un equipo, pueda ser ejecutado de la misma forma en otro al que el sistema de archivos en que se encuentra el programa esté exportado. Es por ello que los sistemas de archivos pueden ser montados con una opción que inhabilita el uso de programas SUID o SGID de dichos sistemas de archivos.

Manejo de Señales

El sistema operativo UNIX controla que la señal enviada a un proceso, provenga del usuario que tiene el mismo UID que el UID real del proceso. En caso en que esto sea así, el sistema deja que la señal llegue al proceso, sino es descartada. Las señales enviadas por el administrador son siempre permitidas. [Bec95]

Temporización y Planificación de Tareas

Las facilidades de la ejecución temporizada de tareas no tiene un riesgo *per se*, pero el hecho que en ciertos momentos el administrador del sistema no esté conectado o la carga del equipo sea menor, puede permitir a un usuario intentar la ejecución de procesos con fines de obtener privilegios sin que esto sea notado, al menos en un primer momento.

Por esto el sistema UNIX provee archivos que controlan quienes pueden hacer uso de las facilidades del *at* y del *cron*. Estos archivos (llamados *at.deny*, *at.allow*, *cron.deny* y *cron.allow*) especifican explícitamente los usuarios que tienen prohibido el uso de estas facilidades y los que tienen permitido utilizarla. En muchos sistemas sólo existen los archivos que prohiben el uso ya que todo usuario que no esté en ellos, tendrá permitido su uso. Si los archivos que listan a los usuarios autorizados existen y están vacíos, entonces ningún usuario tendrá acceso a estas facilidades.

III. ANÁLISIS COMPARATIVO

El hecho que los mecanismos de administración de memoria y procesos sean por demás conocidos y hayan sido probados extensamente, hace que las principales fallas de seguridad en cuanto a este aspecto del funcionamiento de los sistemas se deban a errores de programación, ya sea de los módulos de administración de memoria del núcleo, como de los procesos mismos.

La protección de los procesos de usuario es igualmente buena en Windows NT como en UNIX. El espacio de memoria de los procesos está protegido por el sistema operativo quién controla, además, el envío de señales a los procesos. La protección del espacio de memoria y el control de los procesos del sistema es igualmente buena. La debilidad aquí es la protección del equipo de las fallas en los procesos del sistema operativo.

En este caso todos los sistemas operativos son igualmente susceptibles. Tanto Windows NT como UNIX han probado ser muy vulnerables a fallas de programación del sistema operativo como de los servicios. Las fallas en los programas que componen el sistema operativo suelen conllevar el riesgo de un ataque local que tiene como consecuencias principales la obtención de derechos de administrador por parte del atacante o la caída del sistema.

En cuanto a los servicios que brinda el sistema, muchas veces permiten acceso irrestricto a la información, la disfunción de uno o más servicios o la caída del sistema. Los ataques a los servicios del sistema dependen de los que sean activados en un equipo en particular. Es por ello que se recomienda, en equipos de funciones especificas, no habilitar más servicios de los estrictamente necesarios.

También es posible realizar ataques a un sistema mediante el uso de servicios que estén correctamente configurados y libres de fallas. Por ejemplo se puede acceder a un servidor FTP anónimo y llenar un disco con basura, se puede utilizar el sistema de correo electrónico de un servidor remoto para enviar mensajes a terceros sin ser descubierto y se puede utilizar la facilidad de bloqueo de cuentas tras varios intentos fallidos para bloquear cuentas de usuarios *ex profeso*.

La facilidad de impostación de un usuario en otro es uno de los puntos de mayor riesgo en cuanto a los procesos del sistema, aunque no deja de ser necesaria para la correcta realización de las tareas del mismo. La programación de procesos que deberán tener esta característica debe ser especialmente cuidadosa. Dado que todos los sistemas operativos estudiados poseen esta propiedad, el riesgo debe ser considerado en ambos casos.

La ejecución temporizada de tareas en Windows NT está restringida en un primer momento al supervisor, aunque este puede otorgar derecho a hacer uso de esta facilidad a cualquier usuario. En UNIX, cualquier usuario puede hacer uso de ella, a menos que el administrador restrinja el uso de la misma a ciertos o a todos los usuarios. Aunque la misma no es de por sí un riesgo, puede ser aprovechada para pasar desapercibido un ataque al sistema.

Riesgos Asociados

En los capítulos anteriores se analizaron las características de seguridad de los servicios de autenticación, autorización, auditoría, archivos y procesos de los sistemas operativos estudiados. En este sentido es necesario considerar los riesgos a la seguridad de los sistemas que dichas características suponen.

Las decisiones tomadas por los desarrolladores de un sistema operativo respecto a la implementación de un determinado servicio, específicamente en aquellos estudiados, tienen una profunda implicancia en la definición de la seguridad.

A continuación se señalan algunos de los riesgos asociados a las características de implementación de los servicios analizados.

AUTENTICACIÓN

La importancia de las características del servicio de autenticación no pueden ser negada ya que involucra la identificación de los usuarios del sistema.

De esta forma, el análisis exhaustivo de la configuración de las cuentas de usuario por parte del administrador es imprescindible para evitar que falencias en las mismas puedan favorecer el ingreso no autorizado de individuos.

- Características de la configuración de las cuentas de usuario
  - Cuentas bloqueadas e inactivas. Es importante que estas cuentas no posean una forma de ingreso ya que, si no fuera así, las mismas podrían ser explotadas por atacantes.
  - Cuentas sin expiración de palabra clave. Una cuenta sin expiración de palabra clave puede ser explotada muchas veces a lo largo del tiempo. Es importante conocer estas cuentas para proveerles de una palabra clave suficientemente segura y, en la medida de lo posible, un patrón de caracteres al azar.
  - Cuentas públicas. Las mismas representan el riesgo de servir de primera puerta de ingreso a un sistema ya que no poseen palabra clave o la misma es pública.
  - Intentos de login fallidos. La repetición de intentos fallidos de ingreso al sistema pueden denotar a un intruso intentando ingresar al sistema por fuerza bruta.

Considerando que el esquema de autenticación en los sistemas operativos estudiados se basa en el uso de palabras clave, las características y propiedades de las mismas deben ser de especial cuidado para conseguir un nivel de seguridad aceptable.

- Características de la palabra clave
  - Longitud mínima. Una palabra clave pequeña se puede descubrir por fuerza bruta.
  - Expiración. Una palabra clave que no es modificada puede ser explotada repetidas veces a lo largo del tiempo.
  - Período de validez. Una palabra clave que es modificada muy a menudo puede representar un intruso intentando esconderse.
  - Composición y palabras prohibidas. Una palabra clave con caracteres sólo alfabéticos o sólo numéricos, utilizando palabras sencillas o combinaciones simples puede ser fácil de descubrir.

De igual manera, los algoritmos de cifrado y el almacenamiento de contraseñas suponen riesgos adicionales que deben ser minimizados y controlados. De no tomar los recaudos necesarios, esta información vital puede quedar indebidamente expuesta para usos dañinos.

Otro problema fundamental es la transmisión de credenciales a través de la red. La única manera segura de utilizar un servidor remotamente a través de una red es utilizando palabras clave de uso único o el cifrado de datos. De otro modo, existe mayor probabilidad de obtener esta información de forma más sencilla, ya que la simple captura de los paquetes que se intercambian en este proceso permite conseguir dichos datos.

AUTORIZACIÓN

El análisis de los objetos en los directorios de los usuarios es de vital importancia para detectar anomalías que puedan significar un intento de obtener privilegios no otorgados por el administrador o el dueño de la información, así como facilitar la obtención de información que pueda servir para facilitar el ingreso de intrusos en el sistema.

Entre los hechos que pueden denotar intentos de conseguir privilegios, la obtención de los mismos o que puedan significar un riesgo a la integridad del sistema, se encuentran los siguientes:

- Consistencia e integridad de las cuentas de usuario. Cuentas de usuarios con intérpretes de comando extraños pueden facilitar la ejecución de programas sin autorización. Cuentas sin directorio de trabajo pueden representar un riesgo para el sistema ya que no queda claro el sitio al que se accede al ingresar con las mismas.

- Usuarios con el mismo número de identificación o privilegios de administrador. Los mismos deben ser conocidos ya que esta situación (anómala en general) puede otorgarle a varios usuarios los derechos de un tercero. El riesgo es mucho mayor si los derechos otorgados permiten realizar tareas normalmente reservadas a perfiles de administración.

- Propiedad de los objetos del directorio del usuario. La existencia de objetos de otros usuarios puede implicar el otorgamiento de derechos de escritura sobre directorios privados. De esta forma, información sensitiva puede quedar al alcance de un intruso.

- Permisos de objetos de configuración del usuario. La posibilidad de modificación de los mismos por parte de terceros puede conllevar el riesgo de comprometer la integridad de las cuentas de usuarios, así como la ejecución de comandos por parte de atacantes utilizando como escudo la identidad del usuario con perfil modificado.

- Permisos de objetos del directorio del usuario. Objetos con permisos de escritura pueden ser modificados con fines maliciosos. La privacidad de la información de los usuarios puede verse de igual manera comprometida.

- Accesos a dispositivos y objetos compartidos montados en los directorios de trabajo del usuario. Un usuario que sea capaz de acceder a un dispositivo a través de su directorio de trabajo, podría modificar los derechos del mismo para acceder a la información resguardada en él.

AUDITORÍA

La existencia de un servicio de auditoría reviste suma importancia al momento de detectar anomalías y realizar un seguimiento de actividades sospechosas dentro del sistema. También es necesario como prueba para inculpar al responsable de dichas actividades.

El nivel de auditoría provisto por el sistema operativo debe ser lo suficientemente completo como para permitir el registro de las actividades relacionadas con la seguridad del sistema.

ARCHIVOS

La seguridad de los objetos del sistema es fundamental en la confiabilidad del mismo. En este sentido deben ser analizados primordialmente los permisos de modificación y eliminación de los objetos, así como los de lectura de objetos que contengan información privada o sensitiva. También se debe tener especial control de los dispositivos y de los objetos compartidos.

Cualquier descuido en la configuración de los permisos de estos archivos conllevan distintos niveles de gravedad. Sin embargo, en todos los casos comprometen la seguridad del sistema ya sea en cuanto al acceso a datos confidenciales o en el uso de recursos restringidos.

La realización de una copia o reubicación de un objeto dentro del sistema de archivos trae aparejado el riesgo de una posible disminución de los permisos originales del objeto en cuestión. La eliminación debería ser segura ya que, al quedar la información grabada en el medio físico, existe la posibilidad que la misma pueda ser recuperada con fines maliciosos.

Finalmente, al compartir objetos entre sistemas de computadoras a través de un medio inseguro, como es una red, implica el riesgo de permitir que información sensitiva sea obtenida por terceras partes aumenta notoriamente.

Entre las cuestiones más importantes a controlar se encuentran las siguientes.

- Permiso de escritura. La posibilidad de modificar archivos del sistema puede facilitar el ataque de un intruso, cambiando el comportamiento del sistema con fines maliciosos.

- Permisos de objetos del sistema operativo. La modificación de objetos del sistema operativo, o de algunos de sus atributos pueden provocar que el mismo se vea comprometido. El permiso de lectura sobre ciertos objetos puede permitir obtener información sensitiva a un atacante.

- Permisos de dispositivos. En general los dispositivos deben ser accedidos únicamente a través del sistema operativo. Los permisos de los mismos deben prohibir el acceso directo por parte de los usuarios ya que, de ser así, la información contenida en los mismos podría quedar al alcance de extraños.

- Objetos compartidos. Se debe llevar un control estricto de los objetos compartidos. Si los objetos se comparten otorgando derechos excesivos, estos pueden ser objeto de abuso. Además, los objetos compartidos por la red pueden ser objeto de exposición de los datos contenidos.

- Archivos con Set UID o Set GID (UNIX). Estos archivos pueden ser utilizados para obtener los derechos de otros usuarios, principalmente de administración y, de esta forma, obtener prerrogativas no otorgadas que posibiliten un ataque al sistema.

PROCESOS

La imposibilidad de la búsqueda de fallas de programación por parte de cualquier herramienta en la estructura del sistema hace que este sea aun el mayor riesgo para la seguridad del mismo.

Por otra parte, es notorio el riesgo que presentan los procesos que implementan los diferentes servicios que presta un servidor. Tanto los que prestan servicios locales que, en general pueden sólo ser explotados una vez que se ingresó en el sistema, como los que relacionan equipos a través de la red, que pueden facilitar el acceso remoto.

Dentro de la planificación de procesos en UNIX, dos riesgos pueden presentarse que deben ser debidamente controlados para evitar abusos.

- Permisos de archivos de crontab. La posibilidad de modificar un archivo de crontab puede permitirle a un intruso ejecutar comandos en horas donde se dificulte su detección e incluso bajo la identidad de otro usuario.

- Permisos de archivos referenciados por crontab. Un programa referenciado por crontab con permisos inadecuados puede ser modificado para realizar tareas que afecten la seguridad del sistema y, además, permitir la posibilidad de inculpar a otro usuario por el ataque.

PARTE II

Análisis Práctico

Evaluación de Seguridad

Introducción

Una de las causas más frecuentes del ataque efectivo a la seguridad de un sistema se debe a la configuración inadecuada del mismo. En este contexto, la integridad del servidor en cuanto a la protección de los componentes que administra, constituye un punto primordial.

Esta cuestión comprende aspectos muy variados y particulares al tipo de servidor que se trate. Claramente, el nivel de seguridad a establecer sobre un servidor depende de la función que cumple el mismo y de su importancia relativa en el sistema. Esta determinación deriva del análisis de riesgos que se lleve a cabo sobre éste.

En dicho marco, el sistema operativo se erige como la primera línea de defensa del servidor. Dado el papel fundamental que cumple el sistema operativo en el funcionamiento de un equipo, la estrategia de seguridad implementada por el mismo resulta crucial en la definición de seguridad del sistema como un todo.

Todo sistema operativo moderno ofrece un modelo de seguridad que busca proteger los componentes del sistema, incluyendo hardware, software y datos almacenados en dicho sistema. Cada plataforma comprende mecanismos muy distintos y no tan obvios para proteger estos recursos de acciones indebidas, ya sean intencionadas o accidentales.

El hecho de conocer a fondo las opciones que provee el sistema operativo utilizado en cuanto a la seguridad de los recursos, permite aprovechar correctamente tales mecanismos. Asimismo, es fundamental considerar las vulnerabilidades descubiertas día a día sobre cada plataforma y aplicar las medidas de precaución convenientes en cada caso.

Sin embargo, no basta con definir e implementar las restricciones de seguridad pertinentes. Desafortunadamente, la asimilación de nueva tecnología trae aparejados nuevos riesgos de seguridad. Por tal motivo, es vital llevar a cabo un monitoreo del comportamiento del sistema y las actividades de sus usuarios.

Estos procedimientos apuntan a detectar eventuales anomalías o conductas sospechosas de modo de rectificar con celeridad fallas u omisiones. Debe inculcarse el concepto de que el dinamismo y la evolución permanente de aplicaciones y recursos no permiten una actitud de relajo frente a la cuestión de seguridad.

Resumiendo, además de lograr una configuración inicial de adecuada a las circunstancias, el servidor debe ser objeto de control y seguimiento de los distintos aspectos que impliquen riesgos de seguridad para el mismo. La fidelidad a esta conducta puede evitar la generación ciertas situaciones indeseadas, o al menos atenuar sus consecuencias.

Herramientas de Seguridad

La evaluación de la seguridad de un servidor puede convertirse en una tarea sumamente ardua. Tal como lo refleja el análisis teórico, existen un gran número de variables a tener en cuenta y es sumamente fácil dejar en olvido alguna de ellas, en particular si se realiza un seguimiento manual de las trazas registradas por el servidor.

Es más, si bien los sistemas operativos soportan opciones de registro de eventos que permiten analizar la operatoria corriente del servidor, muchas veces no se habilitan por desconocimiento o escasez de recursos.

En otros casos, la falta de aplicaciones adecuadas para procesar el volumen de información en forma automática, dificulta el análisis de la misma. El almacenamiento de sucesos resulta inútil si no es posible evaluarlos con certeza y rapidez.

Por tal motivo, es conveniente desarrollar o utilizar alguna clase de programa utilitario de modo de implementar procedimientos automáticos o semiautómaticos de control. Básicamente, una *herramienta de seguridad* es un paquete de software que permite evaluar una o varias facetas involucradas en la definición de seguridad del sistema.

Como herramientas de seguridad, se puede apelar tanto a las alternativas integradas que ofrezca la plataforma del sistema operativo de dicho servidor, como a desarrollos de terceros que realicen evaluaciones en base a la información registrada por el sistema o que ejecuten comprobaciones específicas.

Existe una gran variedad de este tipo de herramientas tanto comerciales como gratuitas. Algunas de ellas tienen la capacidad de evaluar cualquier sistema operativo. Sin embargo, dado que son escasos los estándares de seguridad y marcadas las diferencias, la gran mayoría de estos paquetes está destinado a una plataforma específica o bien existen distintas versiones del mismo utilitario para cada sistema operativo.

Los puntos a considerar en esta evaluación de seguridad son todos aquellos que impliquen algún riesgo para la seguridad del sistema, tales como la administración de accesos a recursos, la protección de los archivos de configuración del sistema o cualquier servicio que se ejecute en el servidor en estudio.

Entre todos estos aspectos, las cuestiones relacionadas con la protección de usuarios, archivos y procesos son objetivos típicos de chequeos de seguridad.

Mientras algunas herramientas se abocan a analizar una cuestión específica, tal como un programa que intenta descubrir palabras clave, otras constituyen sistemas complejos abarcando distintas áreas del perímetro de seguridad.

Recursos del Análisis Práctico

El mayor inconveniente para establecer un análisis comparativo desde el punto de vista práctico entre los sistemas operativos estudiados consiste en la falta de una herramienta de seguridad adecuada para las cuestiones que incumben al estudio, para cada una de dichas plataformas.

Estado del Arte

A continuación se citan las herramientas más conocidas que analizan alguno de los aspectos que importan a este estudio.

- COPS (Computer Oracle and Password System)
  Este conjunto de programas fue diseñado por la Universidad de Purdue con el propósito de chequear ciertos aspectos de seguridad de sistemas operativos UNIX. Existen dos versiones, una desarrollada en shell script y C y otra en Perl, con funcionalidad similar. Esta herramienta monitorea el sistema en forma pasiva y reporta potenciales vulnerabilidades relacionadas con contraseñas débiles, formato y contenido de archivos *passwd* y *group*, permisos de objetos, archivos SUID y servicios de red riesgosos.

- Crack
  Este utilitario de dominio público fue desarrollado por Alec Muffet y es considerado el más famoso programa de quebrado de palabras clave encriptadas en UNIX. A partir de ciertas reglas se comprueba cada entrada del archivo de contraseñas utilizando el algoritmo de cifrado DES. Además, es posible agregar diccionarios complementarios para lograr un mayor efectividad. El resultado de la ejecución es almacenado en un archivo indicando el usuario y la palabra clave descubierta.

- DumpACL
  Este utilitario comercial desarrollado por Somarsoft permite obtener información de seguridad de sistemas Windows NT. Brinda la posibilidad de examinar los permisos del sistema de archivos, el Registro, las comparticiones y las impresoras, la información de usuarios y grupos, así como también la configuración de los planes de seguridad, los derechos de usuario y las relaciones de confianza. Estos chequeos se pueden llevar a cabo tanto en el equipo local como en sistemas remotos.

- ESM (Enterprise Security Manager)
  Este producto de origen comercial diseñado por Axent constituye una solución de administración de seguridad amplia que permite definir, manejar y reforzar la información de las políticas de seguridad. Opera en distintas plataformas mediante agentes específicos. Audita el sistema en forma pasiva reportando la conformidad a las políticas establecidas y comprueba las vulnerabilidades del sistema operativo en cuestión. Proporciona chequeos de integridad del sistema detectando, entre otras cuestiones, modificaciones de los archivos de configuración de seguridad.

- Kane Security Analyst
  Esta herramienta de naturaleza comercial de Intrusion Detection System tiene como objetivo monitorear un amplio rango de computadoras y actividades en redes Windows NT. Utiliza inteligencia artificial y estadísticas para comparar las actividades de los usuarios frente a patrones de usuario históricos recolectados con muestras diarias. Estos perfiles dejan una firma distintiva que puede contribuir a rastrear actividades inusuales al detectar discrepancias significativas. Este producto considera seis áreas de seguridad: fortaleza de contraseñas, control de acceso, restricciones en las cuentas de usuario, monitoreo del sistema, integridad y confidencialidad de datos. Además, proporciona una aplicación adicional que permite investigar los derechos y los privilegios asociados a varios usuarios, grupos y directorios.

- SAFEsuite
  Producto de origen comercial, esta familia de herramientas fue diseñada por Internet Security Systems, Inc. (ISS) para auditar, monitorear y corregir aspectos de seguridad de redes UNIX y Windows NT. De hecho, fue una de las primeras aplicaciones de seguridad desarrollada para evaluar la seguridad de Windows NT. Este programa ejecuta una amplia variedad de ataques sobre la red especificada diagnosticando distintos servicios, tales como FTP, RPC, NFS, NNTP, entre otros. Además, ofrece la capacidad de analizar la vulnerabilidad de un equipo frente a enmascaramiento de direcciones IP y ataques de denegación de servicio.

- SATAN (Security Administrator Tool for Analyzing Networks)
  Este paquete de software de dominio público fue desarrollado por Dan Farmer y Weitse Venema con el fin de mejorar la seguridad de la red en ambientes UNIX. El programa monitorea la red analizando los equipos hallados y determinando la existencia de vulnerabilidades conocidas en relación con servicios tales como NIS, NFS, FTP, FTPD, entre otros. Con esta información genera una base de datos relacionando las máquinas consideradas inseguras con aquellas que éstas interactúan. Dispone de tres niveles de

control (mínimo, normal y máximo) y, en caso de encontrar un fallo, genera una breve explicación del mismo.

- SeOS - Unicenter TNG
  Estas dos herramientas trabajan de manera muy similar por lo que se describen juntas. Ambas constituyen productos comerciales (SeOS es desarrollada por Memco y Unicenter TNG por Computer Associates) que proveen monitoreo activo de los equipos UNIX. Al reemplazar la interfaz de llamadas al sistema del núcleo del UNIX, interceptan las mismas y las validan contra una base de reglas definidas por el administrador de seguridad. Es posible alcanzar un nivel de restricciones altamente específico pero la degradación de la performance es, a la vez, notoria.

- Tiger
  Esta aplicación fue desarrollada por la Universidad de Texas y evalúa en forma pasiva sistemas UNIX para detectar posibles vulnerabilidades. Constituye un conjunto de programas C y shell scripts que realizan distintos chequeos de seguridad, tales como comprobaciones de configuración del sistema, archivos especiales (.netrc, .rhosts, etc.), servicios en */etc/inetd.conf*, entre otras. Es posible determinar las pruebas a realizar mediante un archivo de configuración. Asimismo, cuenta con una herramienta extra que, dado un archivo de resultados, se especifican explicaciones adicionales para cada entrada generada por la ejecución del programa estándar.

- Tripwire
  Este paquete de software de dominio público fue elaborado por el Departamento de Informática de la Universidad de Purdue para comprobar la integridad de sistemas UNIX. Esta herramienta crea una base de datos con un identificador de cada archivo analizado con la información referente a propietario, permisos, última fecha de modificación y demás. Posteriormente dicha base puede ser contrastada en ejecuciones subsiguientes de modo de detectar alteraciones no autorizadas al sistema de archivos. Dispone de un archivo de configuración que permite determinar la porción del sistema de archivos a estudiar.

Es posible mencionar muchos otros productos que evalúan cuestiones afines con el presente estudio. Sin embargo, estas herramientas son representativas del amplio espectro de aplicaciones desarrolladas por terceras partes que pueden contribuir a realizar chequeos de seguridad más allá de las provistas por los mismos sistemas operativos.

Selección de Herramientas

Entre todas las herramientas vistas, los productos que más se ajustan a las necesidades del caso son ESM y SAFEsuite ya que operan sobre ambas plataformas realizando los chequeos correspondientes sobre los aspectos de seguridad considerados. Sin embargo, la posibilidad de utilización de alguno de ellos, salvo en versiones reducidas de prueba, se dificulta dado el origen comercial de ambos.

Afortunadamente, fue posible acceder a la versión completa de ESM. Por tal motivo, se optó por trabajar con dicha herramienta.

Breve Descripción de ESM

ESM es un paquete de programas que permite llevar a cabo controles de seguridad sobre diversas plataformas invocando agentes particulares para cada sistema operativo. Utilizando una interfaz gráfica es posible definir las pruebas a llevar a cabo mediante la definición de políticas.

Una política es una colección de módulos que se ejecutan sobre un conjunto de máquinas para obtener un reporte de los problemas potenciales de seguridad. Dicha política puede estar compuesta por numerosas pruebas agrupadas en uno o más módulos.

Los tests son aplicables a un usuario en particular, un grupo de usuarios, un único equipo o toda la organización. Las políticas definidas deben reflejar las normas y las guías a respetar por cada individuo o grupo de usuarios o nodos, de modo de asegurar la conformidad a las líneas de trabajo generales sustentadas por la institución.

En particular, ESM ofrece un conjunto de módulos relacionados con la cuestión de la seguridad. Dichos módulos comprenden tres grandes áreas consideradas de alto riesgo: cuentas de usuario y autorización, configuración de red y servidor y sistema de archivos.

Los módulos y los chequeos definidos para cada una de estas áreas asumen características particulares para las diversas plataformas soportadas por ESM. De este modo, cada agente contiene su propia combinación de controles de seguridad dependiendo de los atributos propios del sistema operativo en cuestión.

Los resultados de la ejecución de ESM reflejan el grado de conformidad con las políticas definidas y las vulnerabilidades que de este hecho se derivan. La comprobación puede realizarse en base a opciones de configuración establecidas en el sistema o mediante plantillas (templates) de reglas a cumplimentar.

En principio, ESM constituye una herramienta sumamente flexible y poderosa para efectuar controles de seguridad adecuados en relación con los aspectos analizados en forma teórica. Sin embargo, el inconveniente se asienta en que los agentes provistos por este producto no cubren las versiones de UNIX consideradas en el presente estudio.

Con el fin de salvar esta cuestión, se consideró la posibilidad de elaborar una aplicación *ad hoc* de modo de asimilar las plataformas faltantes.

USR: una nueva herramienta

Basándose fuertemente en el concepto implementado por ESM, se desarrollo una nueva herramienta de seguridad que realiza controles similares, sobre las problemáticas descriptas en el análisis teórico, para Solaris Intel y Linux.

La idea consiste en adoptar el modelo de ESM, en cuanto a la estructura y la operatoria, adaptándolo a las necesidades planteadas para diseñar una aplicación específica sobre los sistemas operativos citados.

Tal como ESM, se armó un conjunto de módulos que abarcaran cada una de las áreas de seguridad estudiadas en el análisis teórico. Cada unos de ellos comprende un número de tests que examinan los distintos puntos a considerar en dicha área.

Las definiciones de los diversos agentes de ESM para otras versiones de UNIX resultaron sumamente útiles, tanto en el momento de definir los controles a realizar, como en la construcción las plantillas modelo utilizadas en cada test.

Finalmente, la programación de la nueva herramienta se basó en shell script y lenguaje C. El motivo de está determinación fue la de facilitar la posibilidad de exportar este producto a otras plataformas de UNIX.

Considerando estas líneas de trabajo, se concibió y desarrolló la herramienta de seguridad denominada USR - Unix Security Reporter.

Resumen

Se utilizará ESM para evaluar la seguridad establecida en Windows NT definiendo las políticas adecuadas para comprobar los aspectos de seguridad que competen al presente trabajo. Por otro lado, Solaris Intel y Linux serán analizados con USR, aplicación desarrollada con este fin particular.

Los resultados obtenidos con estas herramientas permitirán establecer la base para realizar el análisis comparativo desde el punto de vista práctico de las cuestiones de seguridad equiparables entre los sistemas operativos estudiados.

Herramientas de Seguridad utilizadas en la Tesis

ENTERPRISE SECURITY MANAGER (ESM) DE AXENT TECHNOLOGIES

Descripción

El ESM es una herramienta de software diseñada para administrar y reforzar las políticas de seguridad sobre una gran variedad de plataformas. Una vez que el usuario define las políticas de seguridad deseadas, el ESM se encarga de verificar que las mismas se cumplan, reportando todo tipo de desviación de la política establecida.

El ESM controla los sistemas en busca de vulnerabilidades o privilegios no autorizados, realiza controles de integridad y detecta cambios en archivos y en configuraciones de seguridad de forma de abarcar los tres temas de seguridad de la información: confidencialidad, integridad y disponibilidad.

El mismo utiliza un modelo de Administrador/Agente. Un servicio de Administrador controla la ejecución de políticas por parte de los agentes. A su vez los agentes son específicos para cada sistema operativo, haciendo de esta forma, sencilla la implementación de nuevos agentes para nuevos sistemas operativos.

Este sistema permite definir políticas generales de seguridad. Las mismas están compuestas por módulos que atacan diferentes aspectos de la seguridad de los sistemas. Cada módulo posee controles específicos para cada sistema operativo. De la misma forma es posible determinar grupos de servidores sobre los que se aplicarán las diferentes políticas. Estos grupos son llamados dominios y, dependiendo del tipo de agentes que se estén utilizando, habrá un número de dominios predefinidos, Unix, Windows NT, Novell, etcétera.

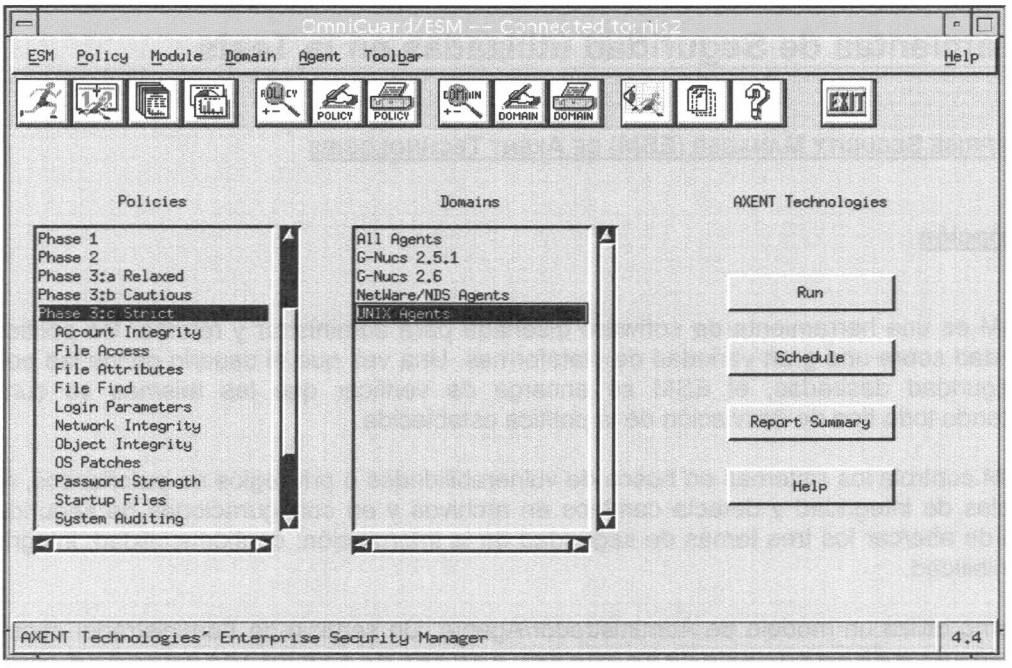
El sistema reporta todas las anomalías halladas sin realizar cambio alguno sobre el equipo tratado. Queda bajo la total responsabilidad de los administradores de seguridad el tomar las medidas necesarias para incrementar la seguridad del sistema, de acuerdo a los problemas encontrados.

Manual del Usuario

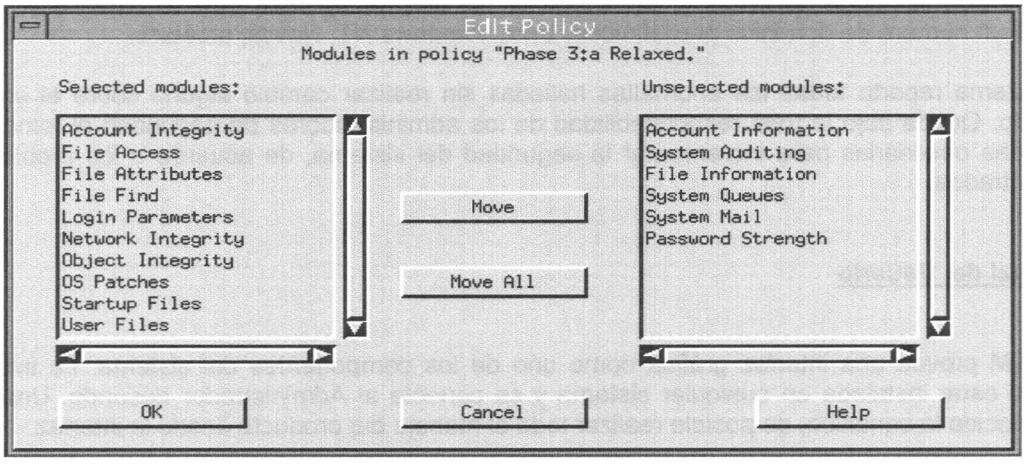
El ESM provee una interfaz gráfica como uno de los componentes del sistema. La interfaz puede estar instalada en cualquier sistema y se conecta al Administrador deseado. Una vez establecida la conexión, es posible realizar todo el manejo del producto desde la interfaz.

La interfaz muestra básicamente dos ventanas, una con un listado de las políticas definidas y otra con los dominios existentes. A través de opciones de los diferentes menús se pueden controlar todas las acciones de las que el sistema es capaz. Además, varios botones permiten realizar las tareas más usuales en forma rápida. Para la mayor parte de las tareas es necesario seleccionar una política y un dominio previo a la acción.

Una vez seleccionados un dominio y una política es posible ejecutar la política sobre el dominio, ver los resultados de las últimas corridas de la política sobre el dominio y agendar la ejecución periódica de la política sobre el dominio.

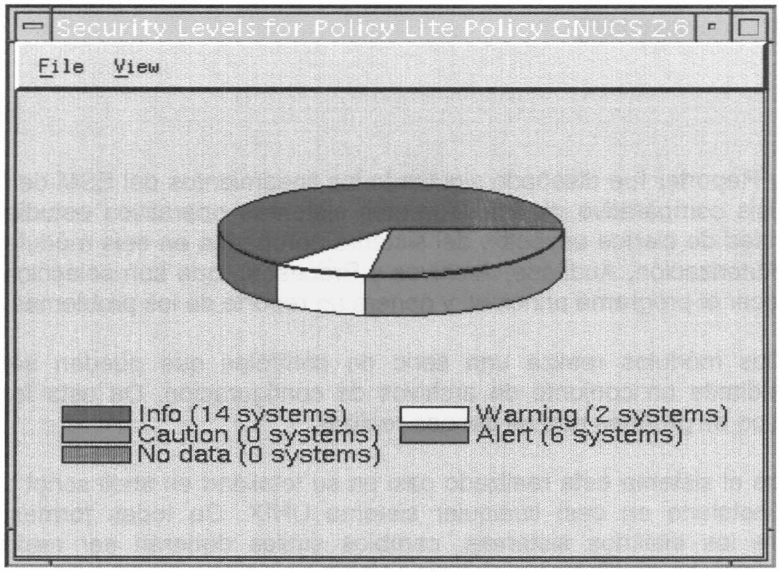


Una vez seleccionada una política es posible editarla, o sea, seleccionar los módulos que serán utilizados en la misma. También es posible configurar los módulos para cada sistema operativo de forma de configurar los controles que se llevaran a cabo. Las opciones de cada módulo serán sumamente disímiles dependiendo del módulo y del sistema operativo al que se aplique. En el caso de los dominios, es posible definir que servidores pertenecen a cada dominio.

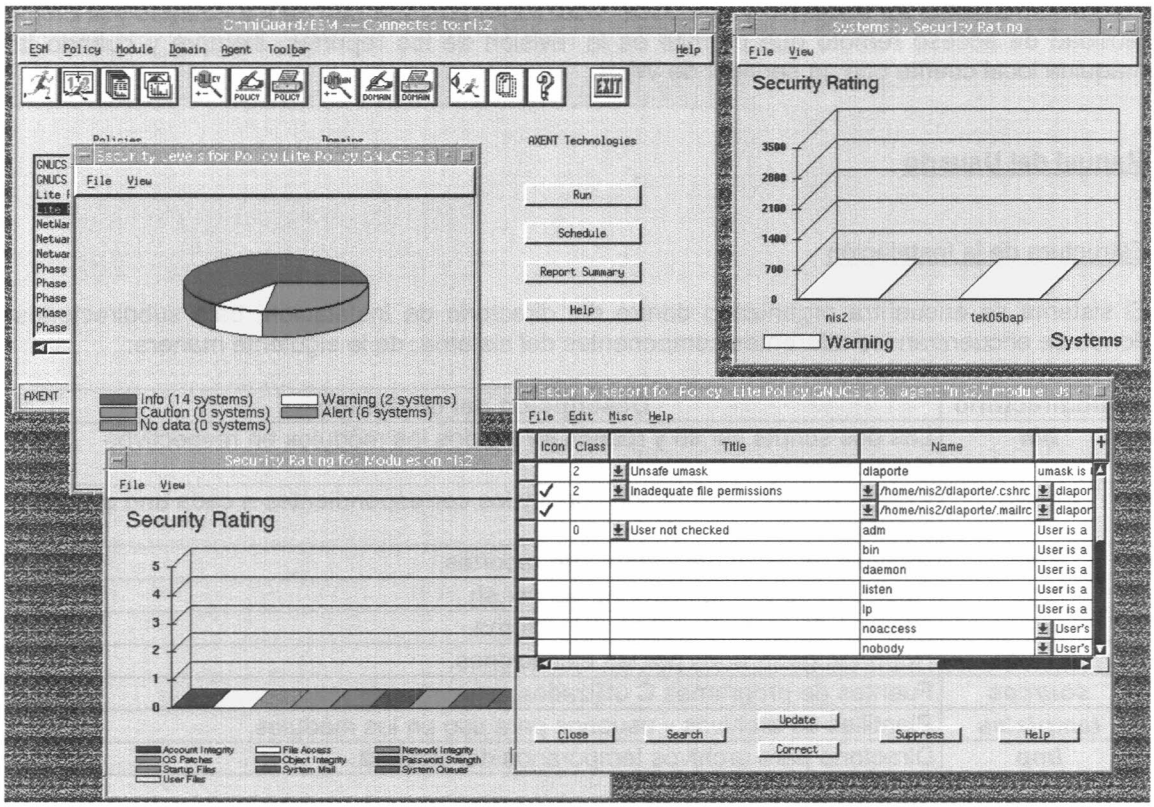


El ESM maneja cuatro tipos de fallas de seguridad según su importancia, asignándoles un puntaje de 0 (Información), 1 (falla leve), 5 (falla de mediana importancia), 10 (falla importante) y 100 (falla grave). De esta forma se calcula un puntaje para cada máquina revisada, de acuerdo al número y tipo de fallas halladas.

Una vez lanzadas una o más políticas sobre uno o más dominios, es posible monitorear el estado de la ejecución de cada una de ellas, detenerlas y/o cancelarlas. Los reportes generados pueden ser revisados de dos maneras diferentes. En forma de tablas o en forma gráfica. Para la presentación gráfica se utilizan colores de acuerdo al puntaje obtenido por cada maquina. De esta forma una falla grave recibe el color rojo y varias fallas leves o de mediana importancia estarán representadas por el color amarillo.



Es posible seleccionar un color y ver todos los equipos que están catalogados dentro de dicha categoría y, seleccionando uno de ellos, observar el resultado de cada uno de los chequeos correspondientes. Finalmente, seleccionando cada uno de los chequeos, se puede ver el detalle de los resultados reportados por el modulo correspondiente sobre el servidor seleccionado.



Mediante la posibilidad de la ejecución periódica es posible llevar un control de los cambios que ocurren en los sistemas y, de esta manera, mantener la seguridad de muchos o todos los sistemas bajo un ambiente de forma sencilla.

UNIX SECURITY REPORTER (USR)

Descripción

El Unix Security Reporter fue diseñado siguiendo los lineamientos del ESM de forma de poder realizar el análisis comparativo de los diferentes sistemas operativos estudiados. El mismo evalúa la seguridad de ciertos aspectos del sistema agrupados en seis módulos (Información, Autenticación, Autorización, Auditoria, Archivos y Procesos), que son seleccionados en forma individual al invocar el programa principal, y genera un reporte de los problemas hallados

Cada uno de los módulos realiza una serie de controles que pueden ser habilitados o inhabilitados mediante un conjunto de archivos de configuración. De esta forma es posible personalizar el tipo de pruebas que se deseen realizar.

Cabe aclarar que el sistema esta realizado casi en su totalidad en shell script previendo así la posibilidad de instalarlo en casi cualquier sistema UNIX. De todas formas, debido a las diferencias entre los distintos sistemas, cambios sutiles deberán ser realizados para un correcto funcionamiento en diferentes sistemas UNIX.

Los scripts se realizaron de manera de módulos funcionales y se previó la incorporación de nuevas versiones de UNIX en la estructura de los mismos. Asimismo, fue necesario incluir un par de procedimientos desarrollados en lenguaje C ya que su desarrollo en shell script se tornaba complicado y excesivamente lenta su ejecución.

Este sistema debe ser instalado y ejecutado localmente en cada uno de los equipos y la única facilidad de acceso remoto que permite es la revisión de los reportes, siempre y cuando la máquina local cuente con un servidor de WEB.

Manual del Usuario

Estructura de la Instalación

El sistema se encuentra organizado dentro del directorio de instalación, bajo subdirectorios donde se encuentran los diferentes componentes del sistema, de la siguiente manera:

Subdirectorio	Descripción del contenido
bin	Los dos scripts usr.sh y rpthtml.sh y todos los módulos en respectivos subdirectorios.
config	El archivo de configuración usr.cf y los correspondientes a cada uno de los módulos.
html	Páginas de WEB del sistema de reportes.
logs	Registros de las ejecuciones del usr.sh.
man	Páginas de manual on-line del sistema.
reports	Reportes generados por las ejecuciones.
sources	Fuentes de programas C utilizados.
templates	Plantillas de archivos y usuarios para uso en los módulos.
tmp	Directorio para archivos temporarios del sistema.

Configuración del USR

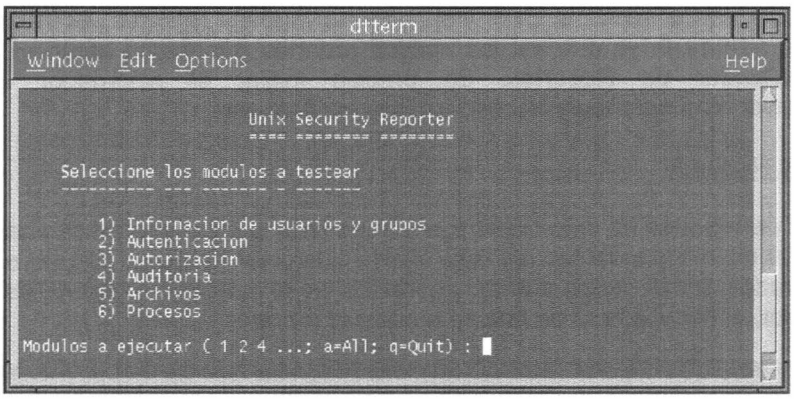
El USR utiliza el archivo de configuración usr.cf para indicar los directorios donde hallar los diferentes componentes. Es preciso editar este archivo y modificar la linea indicada al

momento de la instalación, correspondiente a la variable USR_HOME, que indica el directorio de instalación.

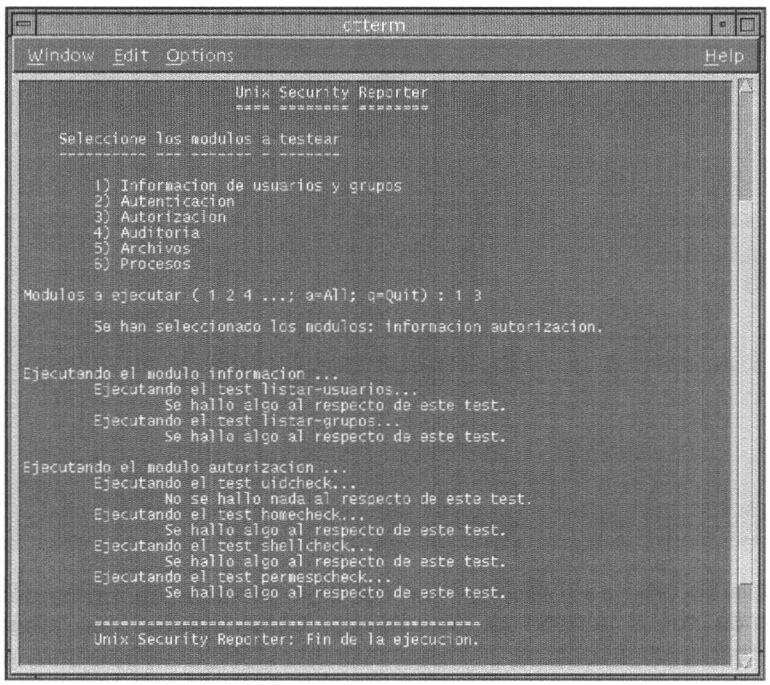
A la vez, hay un archivo de configuración por cada uno de los módulos provistos. En estos archivos es posible especificar los tests a realizar en la ejecución del módulo. A continuación del nombre de cada uno de los tests se debe indicar con "si" o "no" si se desea realizar dicho testeo.

Utilización del Sistema

El Unix Security Reporter posee dos scripts que controlan toda su operación. El usr.sh una vez invocado, presenta una lista de los módulos y el usuario puede elegir el o los módulos que desee ejecutar. Es posible cancelar la ejecución en este punto la ejecución ingresando la opción 'q' o 'Q'. Para seleccionar un grupo de módulos se deben ingresar los números correspondientes separados por un espacio. También es posible seleccionar todos los módulos ingresando 'a' o 'A'.



Una vez seleccionados los módulos deseados, el sistema los ejecuta mostrando un resumen de la corrida, esto es el módulo y test en ejecución y si se halló algo de significación. Esta misma información es resguardada en el log de la ejecución en el subdirectorio logs.



Una vez ejecutados todos los tests deseados se utiliza el script rpthtml.sh para generar la página de WEB que permite seleccionar el reporte a visualizar mediante el USR Report Viewer.

El sistema termina con un código de error (>0) si hubo algún problema y 0 en caso de una terminación exitosa.

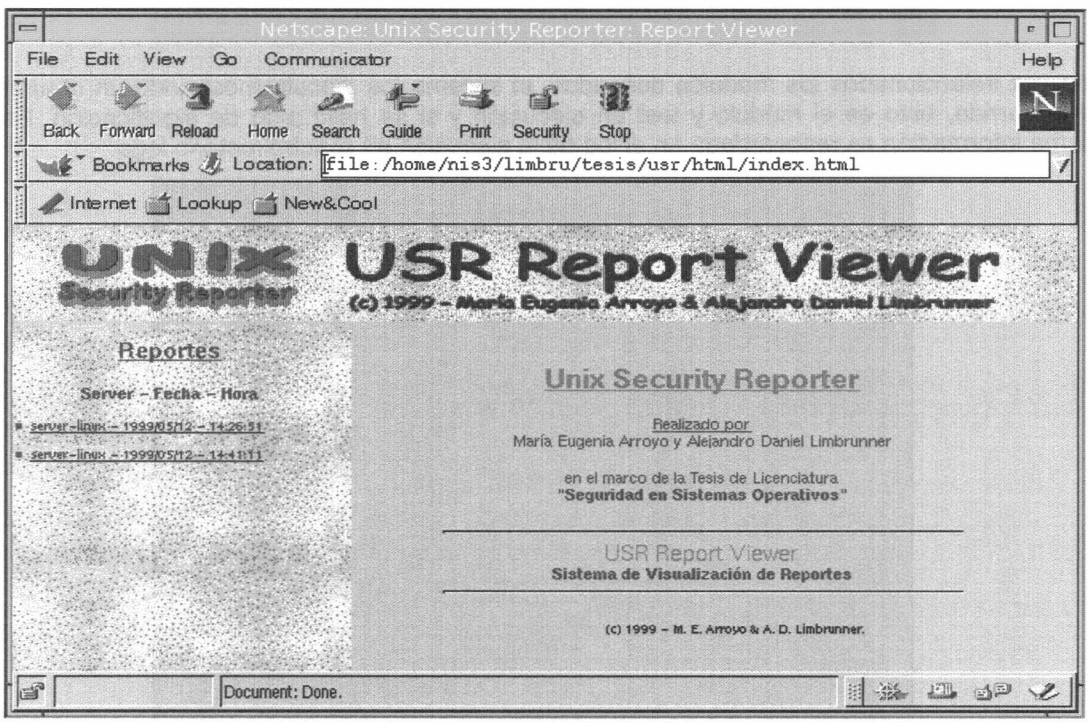
Generación de Reportes

Cada ejecución del USR genera un reporte de la máquina testeada. El mismo indica el nombre del servidor, día y hora de la corrida. A continuación está la información recabada por la ejecución de cada uno de los módulos seleccionados. De acuerdo con los módulos seleccionados y a los hallazgos obtenidos, el reporte puede ser extenso.

Para poder observar cómodamente los reportes generados por el USR, se provee de una interfaz mediante páginas de WEB. El script rpthtml.sh genera una página con todos los reportes existentes en el directorio ~reports y que será utilizada desde el navegador. Es por ello que si se ejecuta varias veces el USR y se generan nuevos reportes, es necesario ejecutar nuevamente el rpthtml.sh ya que así los nuevos reportes generados serán visibles desde el sistema Visualizador de Reportes. Es preciso iniciar el navegador en la página ~html/reportviewer.html para tener acceso a todos los reportes.

Visualizador de Reportes

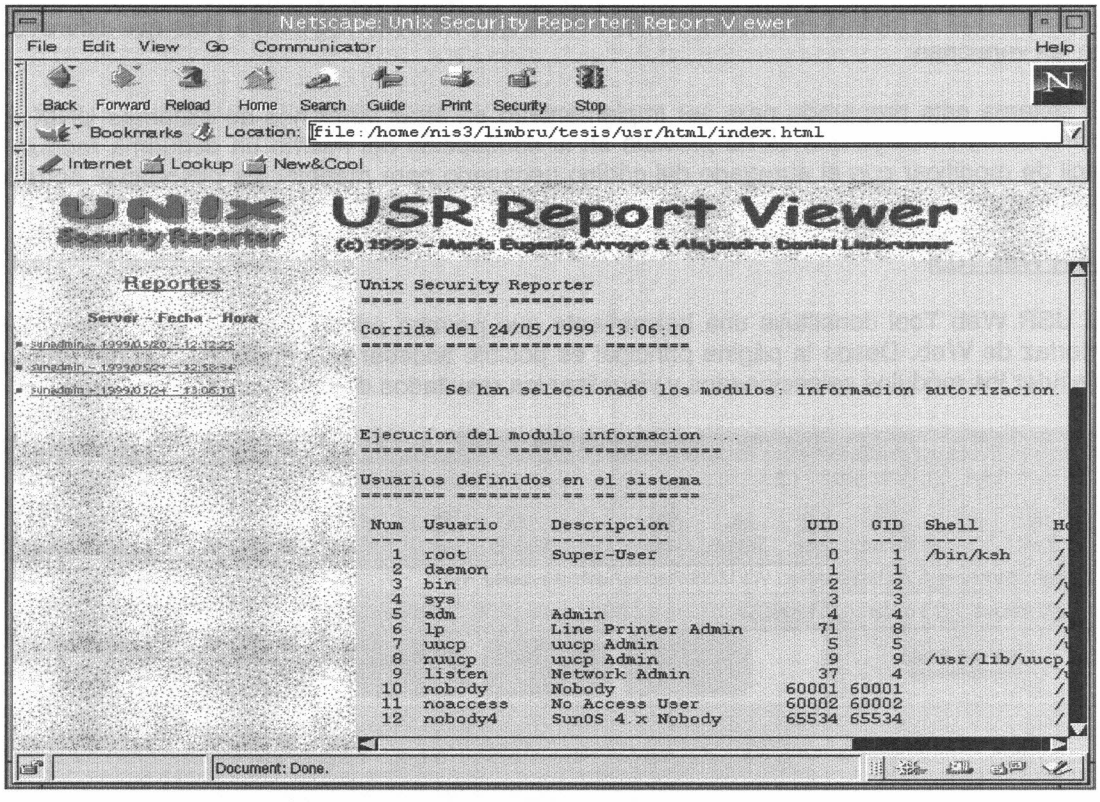
El visualizador de reportes ha sido desarrollado mediante un conjunto de páginas de WEB que permiten que el mismo pueda ser utilizado desde cualquier máquina con un navegador. Los reportes generados por el usr.sh son almacenados en el directorio ~reports de forma que se indica el servidor, el día y la hora en que se ejecutó el reporte.



Si bien es posible acceder a los reportes con herramientas estándar de edición de textos, esto no se recomienda por el tamaño de los mismos. Cabe recordar que, dependiendo de los chequeos solicitados, los resultados de la ejecución del USR pueden ser muy extensos.

Para visualizar los reportes más cómodamente a través de las páginas de WEB es necesario cargar en el navegador la página principal ~html/reportviewer.html. A partir de allí se selecciona del menú de la izquierda el reporte deseado y el mismo será visible en el marco de la derecha.

Nuevamente es necesario aclarar que, tras una nueva ejecución del usr.sh, es necesario ejecutar el rpthtml.sh para regenerar la página del menú y, a continuación, hacer una recarga de la página del menú, donde se verán reflejados los cambios.



Variables de Ambiente

El sistema inicializa un conjunto de variables de ambiente durante su ejecución. Casi todas estas representan los diferentes directorios de la estructura y dependen del valor de la variable $USR_HOME.

Variable	Contenido
USR_HOME	Contiene el directorio base de instalación del Unix Security Reporter.
BIN	Indica el directorio que contiene los archivos ejecutables del sistema.
CONFIG	Indica el directorio que contiene los archivos de configuración.
HTML	Indica el directorio que contiene las paginas de WEB del visualizador de reportes.
LOGS	Indica el directorio que contiene los archivos de registro de las ejecuciones.
REPORTS	Indica el directorio que contiene los reportes de las ejecuciones.
TEMPLATES	Indica el directorio que contiene los templates de los módulos.
COMMON	Indica el directorio que contiene ejecutables de uso común.
TMP	Indica el directorio para archivos temporarios.
SO	Esta variable se inicializa con el nombre del sistema operativo de la máquina donde se ejecuta el sistema.

También es necesario aclarar que se agrega a la variable MANPATH el directorio $USR_HOME/man, para que las páginas de man correspondientes estén disponibles.

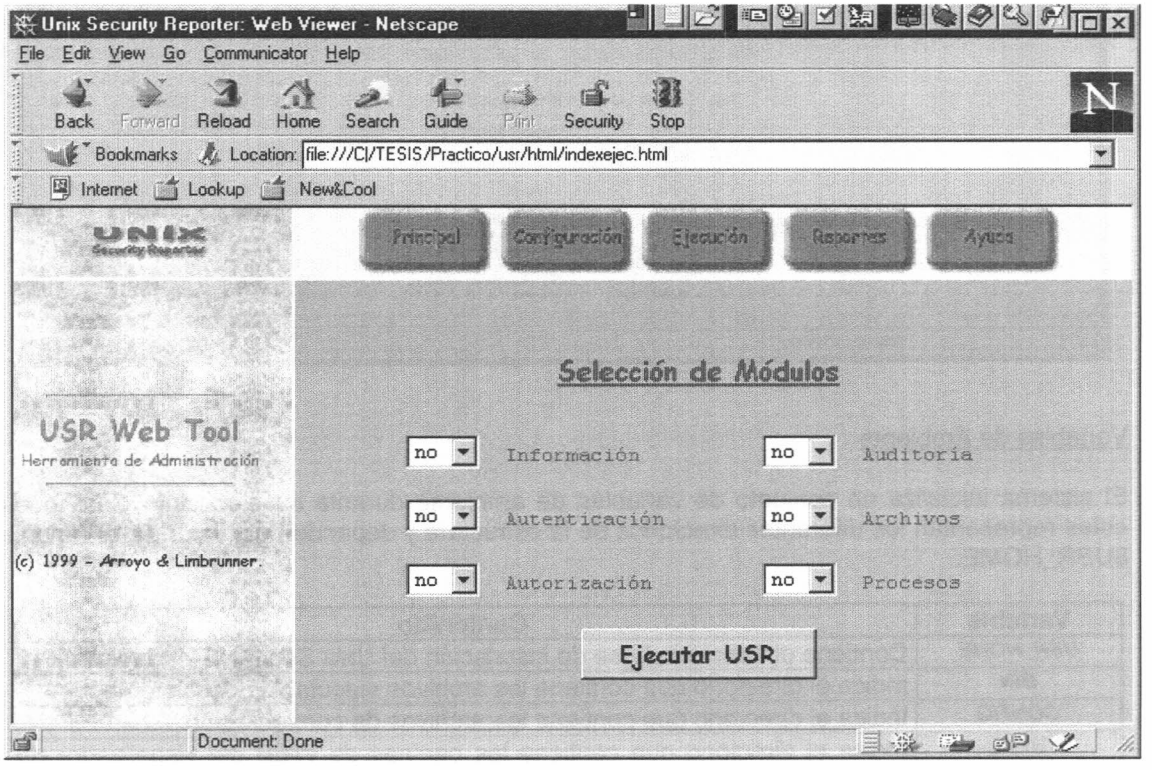
Otras Consideraciones

El sistema, pese a ser genérico en muchos aspectos, depende de la versión de UNIX. Por el momento sólo están soportadas las versiones Solaris y Linux . El sistema ha sido desarrollado sobre Solaris 2.6 y Linux 2.0.36 (Slackware 3.6) y no ha sido probado en otras versiones de los mismos sistemas operativos. Pese a que posiblemente el sistema funcione sin problemas, es probable que el reporte respectivo indique fallas que no sean realmente más que diferencias en las versiones.

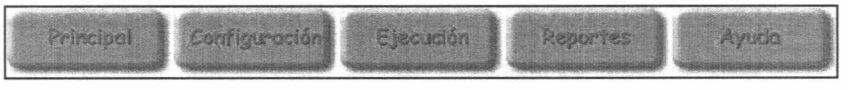
El sistema esta preparado para ser implementado en otras versiones de sistemas UNIX con cambios mínimos ya que se ha previsto en la concepción del mismo un esquema modular y fácil de modificar con el agregado del código necesario para manejar las diferencias entre los distintos sistemas.

USR Web Tool

El USR Web Tool constituye una herramienta que permite administrar el USR mediante una interfaz de Web. Desde la página principal es posible acceder a la configuración del sistema, ejecutar los módulos seleccionados y visualizar los resultados de los chequeos realizados.



En la parte superior de la página principal existe una barra de botones que permiten acceder a las distintas funcionalidades de la herramienta.



Presionando el botón de *Configuración*, se accede a la página que permite editar y modificar los chequeos a ejecutar en cada módulo del sistema.

El botón *Ejecución* explaya la página en la cual es posible especificar los módulos a ejecutar. Los controles a llevar a cabo corresponden a los definidos en dicho momento en los archivos de configuración de cada módulo.

Presionando el botón de *Reportes*, se accede al menú de todos aquellos reportes registrados en el sistema. Al seleccionar un reporte de dicha lista, este se despliega en el marco derecho del mismo modo que en el Visualizador de Reportes.

El botón *Ayuda* posibilita el acceso a una pequeña referencia de ayuda.

Cabe aclarar que el equipo en el que se utilice esta facilidad debe contar con un servidor de WEB de modo de poder ejecutar los scripts referenciados por los distintos vínculos.

Dado que muchos de los controles realizados por el sistema revisan archivos generalmente de acceso restringido, incluso de lectura, resulta necesario ejecutar dicho servidor con privilegios especiales o modificar los permisos de ciertos archivos claves para poder ejecutar las acciones necesarias.

Ambas alternativas implican riesgos de seguridad de gran importancia por lo que se recomienda tomar especiales recaudos al respecto.

Evaluación de Sistemas Operativos

INTRODUCCIÓN

El propósito de este análisis consiste en estudiar el nivel de seguridad de los aspectos que incumben al trabajo en las distintas plataformas. En todos los casos se utilizó el sistema operativo recién instalado, luego de la aplicación de los parches de seguridad sugeridos en cada uno de los casos.

Los sistemas operativos considerados fueron evaluados con las herramientas de seguridad detalladas en el capítulo anterior.

De este modo, Solaris Intel 2.6 y Linux Slackware 3.6 fueron sometidos a los todos chequeos provistos por USR. La versión servidor de Windows NT 4.0 (SP4) fue controlada por ESM mediante la definición de una política adecuada a los intereses del trabajo que permita establecer parámetros de comparación de los aspectos equiparables.

Para facilitar el análisis comparativo siguiente se tomó como base el enfoque de módulos propuesto por USR ya que responde a las mismas líneas establecidas en el estudio teórico.

Cabe aclarar que los registros y los resultados completos de las corridas se incluyen en el Apéndice F. En el caso de Windows NT, también se asienta la política de ESM aplicada.

Vale reiterar que las pruebas fueron llevadas a cabo sobre la base de cuentas que instala por omisión cada uno de los sistemas operativos. Muchas de estas cuentas tienen propósitos administrativos o corresponden a servicios ofrecidos por el servidor y no a cuentas de usuario ordinarias.

Si bien se obtuvieron resultados interesantes, este tipo de testeos puede resultar mucho más fructífero y revelador en la evaluación de sistemas en operación.

INFORMACIÓN

Este módulo sólo brinda información sobre los usuarios y los grupos definidos en el sistema operativo. Se incluyó por compatibilidad con algunos de los testeos realizados en el módulo *Account Information* de ESM.

En ambos casos, la información obtenida puede resultar útil ya que es referenciada por otros controles.

AUTENTICACIÓN

Dado que el modelo de seguridad de los sistemas operativos estudiados se fundamenta en un esquema común de autenticación basado en cuentas de usuario y palabras clave, es posible realizar varias comprobaciones respecto de la administración de estos componentes.

Estado de la Cuenta de Usuario

Una cuenta de usuario puede encontrarse inhabilitada para operar por distintos motivos. En base a esta causa es posible clasificarla como bloqueada, deshabilitada o expirada.

En todos los casos la cuenta no puede ser utilizada como cuenta común de usuario salvo para el uso de algunos servicios específicos.

Cuenta Bloqueada

Una cuenta *bloqueada* impide que el usuario haga uso de los recursos del sistema. En el caso de NT, la cuenta puede ser bloqueada temporalmente por el administrador o por el sistema en respuesta a algún evento. En cuanto a UNIX, el bloqueo refleja la inexistencia de una palabra clave válida.

Todas las cuentas en Solaris y Linux se encontraron bloqueadas con excepción de *root*. Cabe reiterar que muchas de dichas cuentas corresponden a servicios ofrecidos por el servidor por lo cual es lógico que se definan con este estado.

Cuenta Deshabilitada

Windows NT maneja el concepto de cuenta *deshabilitada* que representa la desactivación de la cuenta de forma definitiva. Sólo el administrador tiene la capacidad de modificar esta situación.

Windows NT no presenta cuentas bloqueadas (ESM - Account Information). La cuenta *Guest* está por omisión deshabilitada (ESM - Account Information).

Cuenta Expirada - Desactivada

La cuenta *expirada* es aquella que queda inactiva luego de determinado período de tiempo. Esta figura facilita la administración de cuentas temporales.

Vale aclarar que una cuenta puede quedar inactiva para determinados servicios como resultados del vencimiento de la palabra clave. Por ejemplo, en Linux si un usuario tiene la palabra clave vencida no puede acceder al sistema mediante *login* pero tiene la posibilidad de acceder a su casilla de correo mediante *POP3*.

Por tal motivo, en USR se incluyo el control de cuentas desactivadas que verifica la expiración de la cuenta y de la palabra clave correspondiente.

En ninguno de los casos considerados se registraron cuentas expiradas (Windows NT: ESM - Account Information). Este resultado es el esperable dadas las características de las bases de cuentas analizadas.

Cuenta sin Expiración

Tal como se mencionó anteriormente, el concepto de expiración permite administrar cuentas temporales. Generalmente, las cuentas definidas en el proceso de instalación no se encuentran en estas condiciones.

Esta premisa se refleja en los resultados obtenidos en las evaluaciones de todos los sistemas operativos (Windows NT: ESM - Account Integrity).

Actividad de la Cuenta de Usuario

Una cuenta que no registra actividad puede convertirse en blanco fácil de ataques dado que el propietario no detectará el uso de la misma por parte de terceros. De allí la importancia de determinar las cuentas que no registran actividad durante un período prolongado de tiempo.

En el caso de Windows NT, las cuentas en estas condiciones son *Guest*, *IUSR_SERVER-NT* y *IWAM_SERVER-NT* (ESM - Account Integrity). Mientras que *Guest* se encuentra deshabilitada, las otras cuentas corresponden a usuarios definidos por el sistemas para la utilización de servicios no utilizados al momento de la evaluación.

USR evalúa la actividad de una cuenta diferenciando a los ususarios según tengan o no un *shell* definido. En el primer caso, comprueba la fecha de último *login*. Para la segunda alternativa, verifica el movimiento de la casilla de correo ya que en muchos casos se otorgan cuentas con estas características para brindar servicio de correo electrónico.

De tratarse de una cuenta administrativa o definida para algún servicio, es probable que no posea *shell* y no registre modificaciones en la casilla de correo. Por tal motivo, no es posible determinar a priori la actividad de esta cuenta, tal como los reflejan los resultados de los chequeos realizados tanto para Solaris Intel como para Linux.

Cuentas Públicas

Este tipo de cuentas representan un peligro de seguridad ya que posibilitan la utilización de recursos del sistema sin una autenticación individualizada.

Principalmente, se pueden distinguir las cuentas de invitado y las correspondientes a servicios públicos que exponen determinadas áreas del sistema al acceso en forma anónima.

Mientras que Solaris no presenta cuentas con estas características, Linux define las correspondientes a *guest* y *ftp* (bloqueadas por omisión).

Cabe aclarar que ESM no provee un testeo para detectar la existencia de este tipo de cuentas para Windows NT. De todas formas, ya se ha visto que existen dos cuentas en esta situación, *Guest* y *IUSR_SERVER-NT*, que corresponden a la condición de invitado al sistema y a través de Internet.

Consistencia de los archivos *passwd* y *shadow*

Este testeo surgió como consecuencia de las experiencias prácticas realizadas sobre las distintas versiones de UNIX. No tiene parangón en ESM para Windows NT pero fue incluido en USR dado que expone serias implicancias en el nivel de seguridad del sistema.

El archivo *passwd* almacena las cuentas definidas en el sistema con sus características de configuración en cuanto a la identificación de usuario y grupo, descripción, definición de directorio de trabajo y el *shell* del usuario. Dicho archivo tiene permiso de lectura para todos.

El archivo *shadow* debería contener las mismas entradas que *passwd* y registra las palabras claves encriptadas de cada usuario, además de un número de parámetros relacionados con el período de validez de la cuenta y la contraseña, última fecha de modificación y demás. En principio, este archivo sólo es accesible con privilegios de superusuario.

El control propuesto verifica la existencia de entradas duplicadas en cada uno de estos archivos y la consistencia entre ambos.

En Solaris Intel, si un usuario está definido en *passwd* pero no cuenta con la entrada correspondiente en *shadow*, para el sistema dicho usuario es inexistente. En cambio, Linux no sólo le permite operar sino que expone la contraseña encriptada del usuario en cuestión en el archivo *passwd*.

Si bien una instalación recién establecida no presenta este tipo de problemáticas, son inmediatos los riegos de seguridad que reporta este comportamiento por parte del sistema operativo.

Logins Fallidos

La instalación de Windows NT no activa el registro de eventos de seguridad. Por otro lado, Solaris Intel y Linux por omisión no registran los intentos fallidos de *login*. En todos los casos, es necesario habilitar manualmente estas opciones.

Una vez habilitadas, estas plataformas difieren en la información registrada. Mientras que Solaris almacena cada una de las tentativas fallidas para usuarios reconocidos o no, Linux sólo mantiene una entrada por cada usuario existente con un contador que es reseteado cuando dicho usuario ingresa al sistema exitosamente.

En Windows NT es posible activar el registro de *logins* fallidos en la política de auditoría. Esta opción permite guardar cada uno de los intentos consignando el usuario (válido o no), la fecha, el equipo desde el que se realizó dicho intento así como también el motivo del fallo.

Características de Palabras Clave

La versatilidad de la política de contraseñas que provea un sistemas operativo es de vital importancia cuando la seguridad del sistema puede resultar comprometida como consecuencia de las elecciones de los usuarios.

Si bien existen varias opciones para delinear esta política, en la mayoría de los casos es necesario definirlas en forma explícita ya que no son configuradas por omisión. Otras veces, los valores son demasiado flexibles y pierden su significado.

A pesar de este hecho, es importante destacar que todos los sistemas operativos evaluados tienen en cuenta este tipo de consideraciones en cuanto a la fortaleza de las contraseñas.

A continuación, se resumen las características fundamentales seteadas durante el momento de instalación (Windows NT: ESM - Password Strength).

	Windows NT	**Solaris**	**Linux**
Contraseña Vacía	permitida	no permitida	no permitida
Validez Mínima	0 días	———	0 días
Validez Máxima	42 días	———	99999 días
Longitud Mínima	———	6 caracteres	5 caracteres
Longitud Máxima	14 caracteres	8 caracteres	8 caracteres

Asimismo, bajo las dos versiones de UNIX, existen restricciones sobre la complejidad de las palabras clave, en cuanto a la utilización obligatoria de combinaciones de caracteres alfanuméricos, símbolos especiales, letras mayúsculas y minúsculas.

Cabe aclarar que, en ambos casos, el superusuario tiene la capacidad de definir contraseñas que no respeten estas condiciones (vacías, menor cantidad de caracteres, palabras clave simples, etcétera).

En este aspecto, y tal como se mencionó en el análisis teórico, el Service Pack 2 de Windows NT incluyó una biblioteca dinámica opcional (*passfilt.dll*) que fuerza el uso de palabras clave que contengan caracteres alfanuméricos y especiales. De activar esta facilidad, los usuarios deben seleccionar una secuencia de al menos 6 caracteres combinando al menos 3 de los tipos de caracteres permitidos (mayúsculas, minúsculas, números, símbolos especiales).

Por último, vale aclarar que es posible configurar algunos de estos parámetros para usuarios individuales en las plataformas UNIX, a diferencia de Windows NT que aplica la política en forma global.

Una cuestión de suma importancia está relacionada con el seteo de la palabra clave de *Administrator*, para Windows NT, y *root*, para UNIX. Durante el proceso de instalación tanto de Windows NT como de Solaris, se requiere dicha contraseña, aunque es posible no ingresar ninguna. En cambio, Linux completa la instalación sin realizar esta solicitud, de modo que la cuenta de superusuario no tiene palabra clave.

En el caso de haber expirado la contraseña de un usuario, únicamente *root* tiene la capacidad de cambiarla en el entorno UNIX. En cambio, Windows NT requiere al usuario en cuestión la modificación de la misma al detectar esta situación en el momento de intentar ingresar al sistema.

Programa de Quebrado de Contraseñas

ESM (Password Strength) provee una aplicación con este propósito mucho más completa en cuanto a opciones que USR. Sin embargo, el objetivo de USR consiste en facilitar un chequeo básico sin apuntar a competir con productos específicos desarrollados con este fin en particular.

En Solaris y Linux, no se obtuvo ninguna clave. En cambio, en Windows NT se detectó la contraseña vacía para el administrador ya que no había sido seteada durante el proceso de instalación del sistema operativo.

Resumen de Resultados

	Windows NT	Solaris	Linux
CUENTAS DE USUARIO	base de instalación	base de instalación	base de instalación
Estado			
Bloqueada	no existen	todas salvo *root*	todas salvo *root*
Deshabilitada	*Guest*	no aplicable	no aplicable
Expirada	ninguna	ninguna	ninguna
Sin Expiración	todas	todas	todas
Actividad	*Guest - IUSR_SERVER-NT - IWAM_SERVER-NT*	cuentas del sistema y administrativas	cuentas del sistema y administrativas
Públicas	control no disponible (*Guest - IUSR_SERVER-NT*)	no existen	*guest - ftp*
Consistencia entre passwd y shadow	no aplicable	no aplicable	consistente
PALABRAS CLAVE	valores de instalación	valores de instalación	valores de instalación
Características			
Vacía	permite	no permite	no permite
Validez (día)	Mínima: 0 | Máxima: 42	Mínima: – | Máxima: –	Mínima: 0 | Máxima: 99999
Longitud (carácter)	Mínima: – | Máxima: 14	Mínima: 6 | Máxima: 8	Mínima: 5 | Máxima: 8
Instalación: root	requerida - no obligatoria	requerida - no obligatoria	no requerida
Quebrado	*roo*t (vacía)	sin resultados	sin resultados

AUTORIZACIÓN

Una vez que el sistema operativo acepta que el usuario es quién dice ser, debe determinar qué recursos están disponibles para dicho usuario y con qué privilegios.

Los controles propuestos apuntan a verificar algunas cuestiones básicas de seguridad relacionadas con ciertos parámetros que caracterizan a los usuarios del sistema y al contenido de los directorios de trabajo definidos para éstos.

Consistencia e Integridad

Algunos de los parámetros definidos para un usuario pueden determinar el otorgamiento de privilegios innecesarios y/o indebidos en cuanto al acceso o a la utilización de recursos.

Identificación de Usuario

En el entorno UNIX, el control del número de identificación de usuario (UID) es fundamental ya que este valor es utilizado para controlar el acceso a los recursos del sistema. Por tal motivo, el poseer el mismo UID implica contar con los mismos privilegios sobre los objetos del sistema.

A modo de ejemplo, dado que la cuenta *root* tiene asignado el número de usuario 0, cualquier otro usuario que posea el UID 0, tendrá exactamente los mismos derechos que el superusuario.

Mientras que en Linux no se detectan identificadores repetidos, las cuentas *root* y *smtp* presentan el mismo número de usuario en Solaris (0). En consecuencia, el usuario *smtp* adquiere los mismos derechos que *root* en este sistema operativo.

Si bien este testeo no es aplicable en Windows NT, es posible contrastar los resultados obtenidos en relación a los usuarios con privilegios de administrador (ESM - Account Information). En este caso, únicamente la cuenta *Administrator* contó con tales prerrogativas.

Directorio de Trabajo

Es posible que algunas cuentas del sistema no posean un directorio de trabajo o que éste sea inexistente.

Este hecho de no poseer un directorio de trabajo puede responder a distintas posibilidades. En principio, puede deberse a que el usuario no necesite un directorio de trabajo para hacer uso de los servicios que le son permitidos.

Por otro lado, la cuenta en cuestión puede tener propósitos administrativos o estar asociada con alguno de los servicios ofrecidos por el servidor. Por lo general, esta cuenta también está bloqueada.

En el caso de que el directorio no exista, puede ser consecuencia de un error en la configuración o el resultado de la instalación de los usuarios necesarios para un servicio que luego no es instalado o configurado.

En el caso de Windows NT, las cuentas con directorio de trabajo inexistente son *Guest*, *IUSR_SERVER-NT* y *IWAM_SERVER-NT* (ESM - User Files). Mientras que la primera cuenta se encuentra deshabilitada, las restantes están relacionadas con servicios no utilizados al momento de la evaluación.

Todos los usuarios de Solaris tienen definido un directorio de trabajo válido.

En Linux, en cambio, los usuarios *guest* y *nobody* no tienen un directorio de trabajo, mientras que aquel propuesto para *uucp* es inexistente. Este último caso se debe a que Linux define todos los usuarios administrativos y correspondientes a servicios, sin importar los servicios que luego son instalados.

Shell de Trabajo

Del mismo modo que con el directorio de trabajo, existe la posibilidad que un usuario no tenga definido un *shell* o que éste no sea válido. La validez del *shell* se basa en su presencia en el archivo del sistema */etc/shells*.

Los argumentos para no establecer un *shell* para un usuario son los mismos por los cuales no se define un directorio de trabajo.

Por tal motivo, tanto en Solaris como en Linux, se detectaron muchas cuentas en estas condiciones. Tal como se mencionó en el punto anterior, en general dichos usuarios tienen la cuenta bloqueada.

Cabe destacar que si un usuario no tiene *shell* definido pero su cuenta se halla activa (palabra clave válida), se encuentra en condiciones de ingresar al sistema ya que, por omisión, se le asigna *bourne shell*.

La invalidez del *shell* puede deberse a que la cuenta en cuestión no representa a un usuario ordinario sino que cumple una tarea particular en el sistema.

En este sentido, en Linux se registran los usuarios *halt*, *shutdown* y *sync*, que corresponden a figuras mediante las que se ejecutan funciones específicas para el sistema.

Por omisión, Solaris no instala el archivo */etc/shells*, por lo cual no es posible realizar este control. Sin embargo, es recomendable definir este archivo manualmente por cuestiones de seguridad.

El concepto de *shell* de trabajo no adquiere el mismo sentido en Windows NT que en el entorno UNIX. Por tal motivo, esta comprobación no es aplicable en Windows NT.

Objetos del Directorio de Trabajo

Estos controles apuntan a comprobar la existencia de ciertos objetos en los directorios de trabajo que, si bien pueden ser válidos, pueden comprometer la seguridad del sistema.

Cabe aclarar que la mayoría de estos chequeos no revisten sentido en Windows NT ya que se evalúan características intrínsecas al entorno UNIX.

Asimismo, algunos de estos chequeos revisan únicamente los directorios de trabajo correspondientes a los usuarios ordinarios, ya que carece de sentido realizar tales verificaciones sobre los usuarios definidos para las tareas y los servicios del sistema.

Archivos Especiales

Este control informa sobre los archivos especiales hallados en la raíz de los directorios de trabajo.

Existen un puñado de archivos que pueden incidir seriamente sobre la seguridad del sistema. Sin embargo, la mayoría corresponden a información de configuración relacionada con las aplicaciones utilizadas por el usuario.

De este modo, en USR los primeros adquieren un nivel de criticidad de *Advertencia*, mientras que los últimos sólo revisten el grado de *Información*.

Los resultados obtenidos, tanto en Solaris y como en Linux, reflejan este último punto.

Vale reiterar que muchos usuarios tienen definido el mismo directorio de trabajo (ver resultados del módulo Información), por lo cual comparten los archivos mencionados.

Directorios Ocultos

Si bien un directorio oculto no representa a priori una amenaza de seguridad para el sistema, resulta sospechoso que tenga semejantes características. Por tal motivo, USR busca este tipo de objeto en todo el directorio de trabajo del usuario.

Muchas aplicaciones hacen uso de este tipo de directorios para almacenar las configuraciones de cada usuario del mismo modo que otras lo hacen en un único archivo.

Por lo general, esta situación es la que se aprecia para Solaris y para Linux.

Sin embargo, en el caso de Solaris, también es necesario aclarar que, dado que el directorio de trabajo de *root*, y otros tantos usuarios (ver resultados del módulo Información), es "*/*", la revisión se lleva a cabo sobre todo el sistema, detectándose una veintena de directorios ocultos que están asociados con servicios instalados en el equipo.

Comandos del Sistema

Este chequeo verifica la existencia de archivos cuyo nombre corresponda a comandos del sistema en el directorio de trabajo del usuario. Aunque puede tratarse de un archivo inocente, en principio es recomendable desconfiar de este tipo de coincidencias.

En el caso de Windows NT, no se registra ningún archivo sospechoso (ESM - User Files).

Dadas las características de este control, USR sólo evalúa los directorios de trabajo de los usuarios ordinarios. Los usuarios instalados por el sistema son descartados en base a un *template* que fue construido a partir de las cuentas definidas en la instalación.

Por tal motivo, no se obtuvieron resultados en Solaris y Linux ya que la base de cuentas utilizada en el control es la resultante del proceso de instalación.

Tal como se recalcó al comienzo del análisis, este testeo adquiere importancia al evaluar sistemas en operación.

Links a Dispositivos

Este control apunta a detectar los *links* a dispositivos establecidos desde los directorios de trabajo de usuarios ordinarios. En principio, esta situación es sumamente irregular y objeto de sospecha.

Valen las mismas aclaraciones que en el testeo anterior respecto a los usuarios evaluados por USR y a los resultados obtenidos en la ejecución del chequeo.

Dispositivos Montados

Este testeo controla que la existencia de puntos de montaje de dispositivos en directorios de trabajo de usuarios ordinarios.

Cabe realizar las mismas aclaraciones que en los dos chequeos anteriores en relación a los usuarios evaluados por USR y a los resultados obtenidos en la ejecución del control.

Propiedad de Objetos del Directorio de Trabajo

Los chequeos propuestos en este sentido evalúan que el propietario y el grupo definido para cada objeto del directorio de trabajo, correspondan al dueño de dicho directorio y a un grupo al que dicho usuario pertenezca.

ESM no ofrece este tipo de control para Windows NT.

En el caso de USR, se opera del mismo modo que en los testeos explicados con anterioridad, revisando únicamente los directorios de usuarios ordinarios. Por tal motivo, no se obtuvieron resultados ni en Solaris ni en Linux.

Asimismo, se reitera la conveniencia de llevar a cabo este control en sistemas en operación con el fin de detectar este tipo de irregularidades en los directorios de usuarios ordinarios.

Permisos de Objetos del Directorio de Trabajo

El objetivo de este conjunto de testeos es evaluar los permisos de los objetos del directorio de trabajo.

Este tipo de control no es soportado por ESM para Windows NT.

Permisos de Archivos Especiales

En base a un template de archivos especiales con sus respectivos permisos sugeridos, se verifican los privilegios de los archivos especiales hallados en la raíz del directorio de trabajo.

En el caso de Linux, se detectó que el archivo */root/.bash_history* posee permiso de lectura para todos. Este hecho constituye un compromiso a la seguridad ya que cualquier usuario está en condiciones de averiguar los comandos ejecutados por el superusuario.

En Solaris el archivo /.dtprofile que almacena la configuración del ambiente gráfico también es accesible para todos para lectura. Si bien, esta cuestión no puede brindar información tan valiosa como en el caso anterior, es recomendable modificar estos permisos.

Objetos con Permiso de Escritura para Todos

Este chequeo advierte sobre aquellos objetos de los directorios de trabajo que posean acceso de escritura para todos.

Linux sólo registra el directorio */var/spool/mail* con tales privilegios. Este valor es lógico ya que este directorio contiene las casillas de correo de todos los usuarios del sistema.

En el caso de Solaris, vale realizar las mismas aclaraciones que en chequeos anteriores con respecto al directorio de trabajo de *root* y otros usuarios (ver resultados del módulo Información). Dado que el mismo es "/", la verificación se lleva a cabo sobre todo el sistema.

De este modo, se detectan un gran número de objetos en estas condiciones. Como en Linux, la mayoría de éstos responden a la configuración necesaria para que las aplicaciones y los servicios funcionen adecuadamente.

Objetos con Permiso de Escritura para el Grupo

Este control informa los objetos de los directorios de trabajo que tienen privilegio de escritura para el grupo.

Sólo se hallaron un puñado de objetos en esta situación bajo Linux. Los mismos corresponden a servicios provistos por el sistema y mantienen dicho nivel de acceso para permitir su funcionamiento.

Solaris plantea una problemática más compleja en cuanto a la enorme cantidad de objetos hallados en este chequeo. Por el mismo motivo expuesto en la verificación anterior, se detectaron muchísimos objetos con este grado de privilegio. Por lo general, éste es el adecuado para la función que involucra dichos objetos.

Cabe destacar que el resultado de esta comprobación para Solaris no fue incluido en el Apéndice F dado el volumen de información generada.

Resumen de Resultados

	Windows NT	Solaris	Linux
CONSISTENCIA E INTEGRIDAD			
Identificación del Usuario	chequeo de privilegios de administrador	*root - smtp*	no presenta UID duplicados
Directorio de Trabajo	inexistente: *Guest - IUSR_SERVER-NT - IWAM_SERVER-NT*	sin resultados	indefinido: *guest - nobody* inexistente: *uucp*
Shell de Trabajo	no aplicable	cuentas del sistema y administrativas	cuentas del sistema y administrativas
OBJETOS DIRECTORIO DE TRABAJO			
Archivos Especiales	no aplicable	configuración de aplicaciones del usuario	configuración de aplicaciones del usuario
Directorios Ocultos	no aplicable	configuración de aplicaciones del usuario	configuración de aplicaciones del usuario
Comandos del Sistema	sin resultados	sin resultados	sin resultados
Links a Dispositivos	no aplicable	sin resultados	sin resultados
Dispositivos Montados	no aplicable	sin resultados	sin resultados
PROPIEDAD DE OBJETOS			
Directorios de Usuarios	control no disponible	sin resultados	sin resultados
PERMISOS DE OBJETOS			
Archivos Especiales	no aplicable	*/.dtprofile* (lectura para todos)	*/root/.bash_history* (lectura para todos)
Escritura para Todos	control no disponible	configuración de servicios y aplicaciones	*/var/spoolmail*
Escritura para Grupo	control no disponible	configuración de servicios y aplicaciones	configuración de servicios y aplicaciones

AUDITORÍA

Del mismo modo que con otras cuestiones relativas a la seguridad, los tres sistemas operativos contemplan varias alternativas en lo que respecta a auditoría. Sin embargo, pocos son las que se activan por omisión. En la mayor parte de los casos, es necesario habilitar las opciones manualmente.

Por omisión, Windows NT no registra los eventos relacionados con la seguridad (ESM - System Auditing). Contempla únicamente el almacenamiento de eventos generales.

Solaris y Linux habilitan el demonio de registro de eventos del sistema, aunque se diferencian en la información que almacenan a partir de la instalación. Mientras que Solaris registra los *logins*, en particular los de *root*, y la actividad *su*, Linux sólo considera los últimos *logins*.

Tal como se ha mencionado, en las tres plataformas es posible mejorar sensiblemente este aspecto activando las propiedades soportadas por cada uno de los sistemas operativos.

Resumen de Resultados

	Windows NT	Solaris	Linux
AUDITORÍA			
Registros Activados	eventos generales	eventos del sistema (todos *logins* y actividad *su*)	eventos del sistema (últimos *logins*)

ARCHIVOS

Este módulo brinda información sobre permisos y derechos de archivos del sistema, sistemas de archivos compartidos y dispositivos.

Atributos de Archivos

Este testeo controla los permisos de acceso de los archivos del sistema contra un *template*. El mismo se utiliza para detectar modificaciones de dichos archivos.

Es de notar que Linux instala una gran cantidad de paquetes auxiliares que no son necesarios en una instalación para su correcto funcionamiento y que, a la vez, podría introducir problemas de seguridad. El control halló dos archivos privilegiados con permisos de grupo diferentes de los esperados.

La evaluación de Solaris denota muchas diferencias que se suponen provocadas debido a que el *template* utilizado corresponde a la versión 2.5.1 de Solaris y se utilizó la versión 2.6 de este sistema operativo. Si bien no es tarea compleja la actualización del *template*, se optó por trabajar con el provisto por ESM para mayor consistencia en la comparación. Sin embargo, cabe aclarar que la seguridad no se ve comprometida en ninguno de los casos en que se detectaron permisos dispares a los especificados en el *template*.

En ambos casos se verificaron diferencias en *links* a directorios que, de todas formas, tienen los permisos correctos.

En el caso de Windows NT, se halló que los derechos de acceso a cinco bibliotecas dinámicas han sido limitados (mejorados), debido a la aplicación de service packs (ESM - File Attributes).

De todas formas, hay un sinnúmero de bibliotecas, bajo */WINNT/System32*, que no poseen los permisos recomendados de acuerdo a un documento de Microsoft que especifica diversas configuraciones de distinto grado de seguridad. Vale destacar que cualquiera de ellas sólo puede obtenerse mediante procedimiento manual tras la instalación del sistema operativo.

Archivos con Permiso de Escritura

Este control busca archivos dentro del sistema que posean permisos de escritura por parte de cualquier usuario.

En el caso de Windows NT, esta comprobación es realizada en cierta forma por el testeo anterior mediante los *templates*, para los archivos del sistema.

Esta evaluación en Solaris y Linux es de suma importancia ya que permite detectar posibles puntos de falla de la seguridad en caso en que los archivos con permiso de escritura pudiesen ser alterados maliciosamente.

En el caso de Linux, sólo se hallaron algunos archivos en */tmp* y tres dispositivos que corresponden a *sockets*.

En Solaris la cantidad de archivos con permisos de escritura por parte de cualquier usuario fue mayor, pero en todos los casos se trata de archivos auxiliares que no revisten importancia alguna.

Archivos con Set User o Set Group ID

Este control carece de sentido en el caso de Windows NT pero reviste especial importancia en UNIX. La verificación que realiza USR consiste en la búsqueda exhaustiva de todos los archivos que tengan SUID o SGID habilitado.

Los resultados muestran una gran cantidad de archivos en estas condiciones, pero en todos los casos se trata de archivos en directorios del sistema, en su mayoría bajo */usr*, y que deben tener estos bits habilitados para un correcto desempeño de las funciones.

Seguridad de Objetos

Se consideran objetos a los dispositivos físicos conectados con el equipo. Todos los sistemas operativos proveen mecanismos para acceder a estos dispositivos de forma controlada. El acceso a algunos de estos dispositivos de manera irrestricta puede provocar daños en la integridad de la información o del sistema.

En el caso de Windows NT, ESM (Object Integrity) informa sobre los volúmenes (discos) que no poseen listas de control de acceso. Este control no arroja resultado alguno.

En la evaluación de Linux tampoco se obtuvieron resultados.

En Solaris, en cambio, se detectaron ciertas diferencias en cuanto al grupo de pertenencia de algunos dispositivos y también en algunos derechos de acceso. Es posible que este hecho se deba a que el *template* utilizado responde a la versión 2.5.1. Como se expresó anteriormente, aunque no es tarea compleja la actualización del *template*, se decidió trabajar con el provisto por ESM para mayor consistencia en la comparación

Recursos Compartidos

Todos los recursos que estén compartidos por un sistema pueden ser objeto de intentos de acceso a la información o daño de la misma.

Por omisión ningún sistema UNIX comparte recursos con otros por lo que los testeos no dieron ningún resultado.

En el caso de Windows NT, este sistema operativo comparte ciertas particiones administrativas desde su instalación, a saber: *C:*, *C:\WinNT* y *C:\WinNT\System32\Repl\Import\scripts* (ESM - Network Integrity).

Resumen de Resultados

	Windows NT	Solaris	Linux
ARCHIVOS			
Atributos	accesos a bibliotecas más restringidos	muchas diferencias no fundamentales (*template* de la versión)	diferencias en grupos de algunos archivos (*template* de la versión)
Permiso de Escritura	control efectuado por el test anterior	archivos auxiliares	archivos en */tmp - sockets*
SUID - SGID	no aplicable	gran número de archivos en directorios del sistema (en su mayoría bajo */usr*)	gran número de archivos en directorios del sistema (en su mayoría bajo */usr*)
Dispositivos Físicos	sin resultados	diferencias en propietarios de algunos dispositivos (*template* de la versión)	sin resultados
Recursos Compartidos	*C:\ -C:\WinNT - C:\WinNT\System32\Repl\Import\scripts*	sin resultados	sin resultados

PROCESOS

Los aspectos de seguridad de los procesos se verificaron según lo planteado en este trabajo. ESM no realiza ningún control de los mismos para Windows NT. Pese a que no se ha utilizado este producto para los sistemas UNIX estudiados, cabe aclarar que ESM se ocupa de revisar los permisos de los archivos de *cron* en dichas plataformas.

Siguiendo esta línea de trabajo, se ha implementado en el USR un módulo que controla precisamente los permisos de los archivos de *cron*, como así también los permisos de los archivos referenciados por estos últimos.

En este sentido se controla que los archivos de *cron* de cada usuario tengan únicamente permisos de lectura o lectura y escritura para el dueño de los mismos y ningún otro permiso, considerando que la información de dichos archivos es privada a cada usuario.

También se verifica que los programas que son llamados a través del *cron* no tengan permiso de escritura por parte de otros usuarios que no sean estrictamente el dueño del archivo de trabajos del *cron*, dado que esto podría permitir que un usuario modificara maliciosamente un trabajo que será ejecutado, a través del *cron*, por otro usuario.

Por los resultados obtenidos se observa que los archivos de *cron* de los usuarios en Solaris permiten a cualquier usuario, ver el contenido de los de otros usuarios, aunque la modificación esta restringida al dueño.

En cambio, en el caso de Linux, la situación es distinta ya que sólo el dueño tiene derechos de lectura y escritura.

Los archivos referenciados por archivos del *cron* poseen en todos los casos los permisos adecuados para no comprometer la seguridad del sistema.

Resumen de Resultados

	Windows NT	Solaris	Linux
PROCESOS			
Archivos del cron	control no disponible (disponible en UNIX)	visibles a cualquier usuario archivos referenciados correctos	accesibles sólo por el dueño archivos referenciados correctos

PARTE III

Conclusiones

Conclusiones

Todos los sistemas operativos presentan un modelo de seguridad integrado que incluye características básicas, tales como autenticación de usuario, control de acceso y procedimientos de auditoría.

Con respecto a los tópicos considerados en el presente estudio, cabe destacar que los conceptos contemplados en los modelos de seguridad de las plataformas analizadas son muy similares, aunque la implementación de los mismos presenta notorias diferencias.

En todos los casos, la autenticación se fundamenta en un esquema de identificación del usuario por el nombre definido en el sistema y la acreditación del mismo mediante una contraseña. Más allá de las falencias propias de este enfoque, UNIX presenta mecanismos de cifrado más fuertes que los utilizados por Windows NT.

Este modelo de autenticación, típicamente implementado en todo sistema operativo, no constituye un medio seguro en ambientes con altos requerimientos de seguridad. Por tal motivo, es conveniente que el sistema operativo utilizado contemple mecanismos adicionales de modo de combinarlos con el esquema tradicional del secreto compartido (palabra clave), resultando en un sistema de autenticación fuerte.

Si bien siempre se contempla la restricción de acceso a la información utilizada en el proceso de autenticación de usuarios al administrador, se ha comprobado que, en la versión de Linux analizada, dichos datos pueden quedar expuestos a todos los usuarios del sistema ("Consistencia de los archivos *passwd* y *shadow*", página 83).

Asimismo, vale aclarar que en un ambiente Windows NT existe la posibilidad de implementar formas de transmisión segura de los pares nombre-contraseña a través de la red. Esta funcionalidad no es provista por UNIX en forma nativa.

La confidencialidad de los datos utilizados en el proceso de autenticación es un punto fundamental. El sistema operativo debe considerar las medidas necesarias para impedir que dicha información quede al alcance de cualquier usuario.

Por otra parte, la transmisión de las credenciales del usuario por la red constituye un punto débil. Si bien es posible contemplar la implementación de mecanismos de cifrado en el intercambio de este tipo de datos, es posible aplicarlos únicamente en un entorno homógeneo. Proveer este tipo de funcionalidad en ambientes heterogéneos implica el desarrollo y la adopción de un protocolo estándar para la transmisión de información sensible vía red.

Tanto UNIX como Windows NT proveen modelos de control de acceso discrecional sobre los recursos del sistema basándose en las figuras de usuario, grupo y dominio. Mientras que Windows NT se apoya en listas de control de acceso, UNIX presenta un enfoque rudimentario de este concepto mediante la máscara de bits de permisos. Sin embargo, cabe destacar que la tendencia general en el ambiente UNIX apunta a ofrecer esta facilidad, tal como ya la proveen la mayoría de las versiones comerciales.

El concepto de listas de control de acceso es sumamente útil. El inconveniente se centra en que cada plataforma realiza una implementación diferente del mismo. Resultaría interesante considerar el desarrollo de algún estándar de modo de permitir la interacción entre los distintos sistemas operativos. Asimismo, esta alternativa combinada con el cifrado de las comunicaciones posibilita elevar el nivel de seguridad en la compartición de archivos.

Si bien existen diferencias en las herramientas de auditoría implementadas en las plataformas analizadas, en líneas generales son muy rudimentarias en cuanto al monitoreo de la seguridad

del sistema. Resulta factible incrementar notoriamente el grado de auditoría, aunque esto requiere gran trabajo manual y conocimiento del sistema por parte del administrador o la generación y la implementación de herramientas adecuadas a tal fin.

Sería conveniente que los sistemas operativos ofrecieran herramientas para el monitoreo activo del sistema en cuanto a los intentos de acceso u otras acciones sospechosas. Al menos deberían contemplar una serie de facilidades que permitieran desarrollar este tipo de funcionalidades.

El diseño de la protección de procesos es igualmente bueno en los sistemas operativos contemplados. Sin embargo, todos ellos se ven afectados en gran medida por fallas de programación en el sistema operativo mismo o en los servicios ofrecidos, que comprometen la estabilidad del sistema. En este sentido, se ha observado que las disfunciones son más frecuentes bajo Windows NT que en el ambiente UNIX.

Por tal motivo, se debería poner especial énfasis en el proceso de desarrollo y en la fase de testeo del sistema operativo con el fin de evitar que fallas de las características mencionadas facilitaran el abuso, el acceso y el eventual daño de los recursos del sistema, ya sea hardware, software o datos.

Tal como se ha expuesto, todas las plataformas soportan propiedades de seguridad, algunas de ellas muy fuertes. No obstante, la instalación original del sistema operativo está orientada hacia la facilidad de uso y la conectividad.

La configuración de los aspectos concernientes a la seguridad queda exclusivamente bajo la responsabilidad de los administradores. Los valores establecidos por omisión durante el proceso de instalación son increíblemente relajados. En consecuencia, los administradores deben realizar una serie pasos manuales habilitando estas opciones en forma explícita con el fin de incrementar la seguridad del sistema.

En este sentido, resultaría conveniente que el sistema operativo ofreciera opciones de instalación que permitieran especificar el nivel de seguridad inicial deseado así como mecanismos de administración más sencillos y automáticos para administrar la seguridad del sistema.

Líneas de Trabajo a Futuro

A partir de la presente investigación es posible derivar diversas líneas de trabajo.

En cuanto al estudio en sí mismo, surgen dos alternativas inmediatas que apuntan a la completitud del análisis. En principio, resulta factible considerar otros aspectos del modelo de seguridad de los sistemas operativos estudiados. Por otro lado, existe la posibilidad de incorporar al análisis otros sistemas operativos.

Para cualquiera de estas opciones, se pueden seleccionar otras herramientas de seguridad adecuadas a las necesidades del estudio en cuestión o utilizar las ya dadas con las modificaciones y los agregados pertinentes.

En este último caso, es recomendable trabajar sobre la generalidad de los templates definidos para cada sistema operativo de modo de asimilar con mayor facilidad las distintas versiones de los mismos.

Otra cuestión a tratar apunta a la ejecución remota del USR. Si bien se provee una herramienta de WEB con este fin, resulta fundamental mejorar el nivel de seguridad de la misma.

De hecho, la seguridad de la transferencia de datos sensitivos por la red, en particular mediante el protocolo *http*, abre un nuevo abanico de interesantes problemáticas de seguridad que bien pueden brindar la base de muchas otras investigaciones.

Dentro de la gama de aspectos considerados en este estudio, es posible seleccionar alguno de ellos para ser analizado en mayor nivel de detalle e, incluso, no restringido únicamente en el entorno de sistemas operativos. A modo de ejemplo, retomar en profundidad el tema de autenticación.

Asimismo, existe la posibilidad de considerar los problemas detectados sobre las distintas plataformas y plantear posibles soluciones. En el caso de sistemas operativos abiertos es más factible encarar la aplicación práctica de las soluciones propuestas.

Bibliografía

[Atk97] Atkins, D.; Buis, P.; Hare, C.; Kelley, R.; Nachenberg, C.; Nelson, A.B.; Phillips, P.; Ritchey, T.; Sheldon, T.; Snyder, J. "Internet Security Professional Reference". New Riders. 1997.

[Bec95] Becker, G.; Morris, M.; Slattery, K. "Solaris Implementation". Sunsoft Press. 1995.

[Cof92] Coffin, S. "Unix SVR4. Manual de Referencia". McGraw-Hill Interamericana. 1992.

[Col97] Collinson, P. "Network File Systems". En "SunExpert", Vol. 8 No. 9. 1997.

[Dai98a] Daily, S. K. "NT Server Security Checklist", part 1. Windows NT Magazine. July 1998.

[Dai98b] Daily, S. K. "NT Server Security Checklist", part 2. Windows NT Magazine. October 1998.

[Gar94] Garfinkel, S.; Spafford, G. "Practical Unix Security". O'Reilly & Associates, Inc. 1994.

[Gar96] Garfinkel, S.; Spafford, G. "Practical Unix Security and Internet Security", second edition. O'Reilly & Associates, Inc. 1996.

[Gil92] Gilly, D. "Unix in a Nutshell System V edition", second edition. O'Reilly & Associates, Inc. 1992.

[Hun92] Hunt, C. "TCP/IP Network Administration". Prentice-Hall, Inc. 1992.

[Ker84] Kernigan, R.; Pike, S. "The Unix Programming Enviroment". Prentice-Hall, Inc. 1984.

[Lee97a] Lee Henry, S. "SunExpert". Vol. 8 Nº 6. 1997. Pp. 47-51.

[Lee97b] Lee Henry, S. "SunExpert". Vol. 8 Nº 7. 1997. Pp. 38-39.

[LeF98] LeFebvre, W. "Permissions and access control lists". En "Performance Computing". October 1998.

[Lew98] Lewis, J.; Morse, S. "Network Strategy Report. Windows NT Security". The Burton Group. 1998.

[Mah98] Mahan, R.E. "Computer and Network Security". Pacific Northwest National Laboratory. 1998.

[Mic96] Microsoft Corporation. "Windows NT Workstation Resource Kit". Microsoft Press. 1996.

[Mic97] Microsoft Corporation. "Windows NT Server - Guía de Redes". McGraw-Hill. 1997.

[Mic98] Microsoft Corporation. "Meeting Enterprise Security Needs: Microsoft Windows NT and Unix". 1998.

[Ple96] Pleas, K. "Securing Windows NT". Windows NT Magazine. 1996.

[Ram94] Ramsey, R. "All about administering NIS+", second edition. SunSoft Press. 1994.

[She97] Sheldon, T. "Manual de Seguridad de Windows NT". Mc-Graw Hill. 1997.

[Ste91] Stern, H. "Managing NFS and NIS". O'Reilly & Associates, Inc. 1991.

[Ste90] Stevens, W.R. "Unix Network Programming". Prentice-Hall, Inc. 1990.

[Rus98a] Russinovich, M. "Inside Memory Management", part 1. Windows NT Magazine. August 1998.

[Rus98b] Russinovich, M. "Inside Memory Management", part 2. Windows NT Magazine. September 1998.

[Rus98c] Russinovich, M. "Windows NT Security", part 1. Windows NT Magazine. May 1998.

[Rus98d] Russinovich, M. "Windows NT Security", part 2. Windows NT Magazine. July 1998.

[Tan92] Tanenbaum, A.S. "Modern Operating Systems". Prentice-Hall, Inc. 1992.

[Tan96] Tanenbaum, A.S. "Computer Networks", third edition. Prentice-Hall, Inc. 1996.

[Win93a] Winsor, J. "Solaris System Administrator's Guide". SunSoft Press. 1993.

[Win93b] Winsor, J. "Solaris Advanced System Administrator's Guide". SunSoft Press. 1993.