PROCEDIMIENTOS PARA EL DIAGNÓSTICO Y EVALUACIÓN DE LA SEGURIDAD DE LAS PLATAFORMAS UNIX SUNOS DE LA CUTB

HERMANN DAVID AGUALIMPIA DUALIBY
ABDÓN ELÍAS RAMÍREZ PLAZAS

CORPORACION UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERÍA DE SISTEMAS
CARTAGENA DE INDIAS D.T. Y C.
1999

PROCEDIMIENTOS PARA EL DIAGNÓSTICO Y EVALUACIÓN DE LA SEGURIDAD DE LAS PLATAFORMAS UNIX SUNOS DE LA CUTB

HERMANN DAVID AGUALIMPIA DUALIBY
ABDÓN ELÍAS RAMÍREZ PLAZAS

Tesis de Grado para optar el título de
Ingeniero de Sistemas

Director
JAIME TADEO ARCILA IRIARTE
Ingeniero Electricista

CORPORACION UNIVERSITARIA TECNOLÓGICA DE BOLÍVAR
FACULTAD DE INGENIERÍA DE SISTEMAS
CARTAGENA DE INDIAS D.T. Y C.
1999

Nota de aceptación

_____

_____

_____

_____
Presidente del jurado

_____
Jurado

_____
Jurado

Cartagena, 23 de abril de 1999

LISTA DE TABLAS

Pág.

LISTA DE FIGURAS

Pág.

LISTA DE ANEXOS

Pág.

AGRADECIMIENTOS

Los autores expresan sus agradecimientos a:

Dios por darnos vida y salud.

JAIME TADEO ARCILA IRIARTE, Ingeniero Electricista. Profesor catedrático de la Corporación Universitaria de Bolívar y Director de la tesis, por sus valiosas orientaciones.

JUAN MARTÍNEZ LAMBRAÑO, Ingeniero de Sistemas. Profesor catedrático de la Corporación Universitaria de Bolívar, por sus aportes y colaboración.

A nuestras familias por su sacrificio y apoyo incondicional en todo momento.

Y a todas las personas que de una u otra manera facilitaron la consecución de este Trabajo de Grado

CONTENIDO

INTRODUCCION

En todo sistema de cómputo es muy importante tener en cuenta los criterios de seguridad y mucho más si el sistema está conectado a una red. La razón es obvia ya que a medida de que más personas tengan la oportunidad de accesar directa o indirectamente el sistema de cómputo, aumenta la posibilidad de que se vayan a realizar ataques al sistema. Por lo anterior, cuando la red a la cual está conectado el sistema de cómputo es ***Internet***, son muchos los riesgos que se corren y los problemas que se pueden tener.

Las últimas estadísticas indican que ***Internet*** consiste de más de 30000 redes con un total de 2500 millones de ***host***. Con tantos usuarios de red en ***Internet***, hay, desafortunadamente, un pequeño número de usuarios que son ***hackers*** maliciosos que se dedican a realizar ataques a los sistemas de cómputo.

La potencialidad que ofrece un ataque interno o externo puede ser de tal magnitud que afecta la infraestructura de la red de una organización. Por lo tanto, son muchos los riesgos que se corren.

Debido a esto, se podría pensar que la seguridad en ***Internet*** es un objetivo inalcanzable; pero no es así, ya que existen técnicas y herramientas, que si se

aplican correctamente, pueden permitirle a una plataforma computacional **UNIX** conectada a ***Internet*** contar con un nivel de Seguridad apropiado para satisfacer las necesidades de los usuarios y proporcionar al exterior servicios útiles.

Como **UNIX** es la plataforma más usada hoy en día en ***Internet***, en este proyecto se pretende introducir los aspectos referentes a la seguridad de dicho sistema operativo, los cuales no fueron considerados en el desarrollado inicial realizado por ***AT&T Bell Labs***. Con la aparición de ***hackers*** y ***crackers*** se han adicionado características de seguridad al sistema **UNIX** y otras mejoras a través de herramientas.

Inicialmente se parte desde lo más general del sistema operativo **UNIX**, aclarando algunos conceptos básicos, que servirán de ayuda para comprender los temas posteriores del proyecto; llegando a puntos más específicos dentro de los cuales están: control de acceso, protección de archivos, seguridad en los servicios ***Internet*** y algunas herramientas para evaluar la seguridad de un sistema de cómputo.

El propósito de este proyecto es el de guiar de una manera fácil y práctica para administrar, de una forma moderadamente segura, una estación de trabajo basada en **UNIX**.

1 GENERALIDADES

Para tratar los conceptos de seguridad del sistema operativo **UNIX**, primero es ‘necesario revisar algunos conceptos básicos de los sistemas operativos, la arquitectura de **UNIX** y la manera como interactúa con el hardware, con el usuario final, y con el software especializado o aplicativo.

1.1 SISTEMAS OPERATIVOS

Un sistema operativo es un conjunto de programas que se utilizan para administrar los recursos del computador; estos pueden ser tanto físicos (teclado, memorias, discos), como lógicos (software, servicios de impresión, servicios de red).

Cuando el sistema operativo está administrando los recursos, debe existir un control al momento en que éste sea requerido, debido a que se pueden presentar varias solicitudes a un mismo recurso simultáneamente. Este mecanismo de control debe ser lo más eficiente posible, para evitar que el tiempo de espera sea demasiado extenso o quede en un ciclo indefinido.

Existe un esquema llamado **MULTITAREA**, que se presenta cuando un sistema operativo tiene la capacidad de realizar varias tareas al mismo tiempo. Pero realmente si únicamente contamos con un procesador, éste sólo puede tener un

programa en ejecución o proceso en un instante de tiempo determinado; debido a las altas velocidades de procesamiento y a los eficientes algoritmos de planificación de procesos, se le da la ilusión al usuario que se estuvieran ejecutando varias tareas simultáneamente. Bajo este esquema, un usuario puede ejecutar una o varias aplicaciones mientras otros de sus procesos realizan una impresión, o una actualización de archivos.

En los sistemas operativos multiusuario, por ejemplo **UNIX**, a este esquema en particular se le conoce como **MULTIPROGRAMACION**; es decir, soportan varios usuarios accesando los mismos recursos al mismo tiempo.

Cuando uno o varios procesos se ejecutan en varios procesadores, este procedimiento se llama **MULTIPROCESAMIENTO**.

1.2 UNIX

El eje central del sistema operativo **UNIX** que siempre reside en memoria se denomina ***Kernel*** (***ver figura 1***). Las funciones del ***Kernel*** son: Controla los recursos de hardware a disposición de los procesos de usuarios; permite a distintos usuarios compartir recursos y ejecutar sus programas; proporciona un sistema de archivos para la administración del almacenamiento de la información; posee rutinas de administración de memoria y procesos.

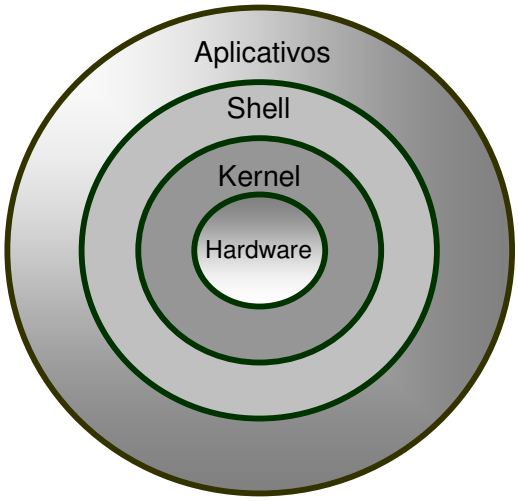


Figura 1*. Arquitectura de UNIX*

Los dos conceptos centrales sobre los que se basa la arquitectura del **UNIX** son los ***archivos*** *y* ***procesos***. El ***Kernel*** está diseñado para facilitar los servicios relacionados con el sistema de archivos y con el control de procesos.

El sistema de archivos o ***filesystem*** de **UNIX** ***(ufs)*** para un usuario final, es un conjunto de archivos y directorios usados para almacenar y organizar la información. Desde este punto de vista en **UNIX** existe una jerarquía de directorios que para un sistema estándar sería:

/	Raíz
/dev	Fichero especiales de dispositivos
/lib	Bibliotecas del sistema
/bin	Órdenes mas empleadas
/etc	Datos y ordenes restringidas al superusuario
/tmp	Ficheros temporales (se borra periódicamente)
/usr	Órdenes, bibliotecas y programas adicionales
/users	Directorios de usuarios

Para el sistema operativo, un ***filesystem*** es una partición que ha sido formateada en bloques de datos y contiene una estructura de tablas que definen la localización de los archivos y directorios. Cuando se crea un ***filesystem***, en ese momento ya se definen bloques que almacenarán información.

Un aspecto importante de **UNIX** es el interprete de comandos llamado ***shell***. Este arranca cuando se inicia el sistema; la función principal es la de ejecutar los comandos dados por el usuario. En el caso de **UNIX** existen tres tipos de ***shell***: dependiendo su uso, de la aplicación que se utilice o el comando a ejecutar; estos son: ***C-shell, Bourne-shell*** y ***Korn-shell.***

1.3 LA SEGURIDAD EN UNIX.

UNIX en sus inicios no fue diseñado para ser un sistema seguro, dado que en un principio fue concebido para uso académico y, por tanto, estaba abierto a todo el mundo. Una vez que se fue extendiendo su utilización, aparecieron distintas marcas que empezaron a ofrecer, en sus distribuciones, el nivel de seguridad necesario para los actuales requerimientos que se le suponen a este sistema. Las implicaciones de seguridad se pueden dividir en tres áreas o dominios dependiendo de la parte del sistema que se vea involucrado: *La seguridad en el sistema de cuentas, la del sistema de archivos, y la seguridad de red*.

1.3.1 Seguridad en el sistema de cuentas: La forma más fácil de un ***cracker*** para llevar a cabo un acceso no autorizado en un sistema, es a través de la cuenta de otra persona. Para evitar esto la medida de seguridad que presenta el sistema **UNIX** es el nombre de usuario (***login***) y la contraseña (***password***). Debido a esto, la protección de cuentas de usuarios es la primera línea de defensa. Es importante, para que la seguridad no se vea comprometida, controlar viejas cuentas, evitar contraseñas fáciles de adivinar, y supervisar todo lo que pueda facilitar un acceso a una cuenta por parte de una persona que no sea su propietario. En el capítulo 2 se tratará todo lo relacionado con este aspecto.

1.3.2 Seguridad en el sistema de archivos: La seguridad en el sistema de archivos es fundamental para la protección de los datos de acceso no autorizados y, todavía más importante, de posibles daños. Las barreras de entrada que se encuentran en el sistema de archivos son los derechos sobre los mismos. Los privilegios de un archivo indican quién o qué puede acceder a ese archivo, y qué es lo que puede hacer con él. Hay tres tipos de privilegios:

- ***Lectura (r)*:** Indica quién puede listar sus contenidos.
- ***Escritura (w)*:** Indica quién puede modificar el contenido de un archivo o borrarlo.
- ***Ejecución (x)*:** Indica si se puede ejecutar como un programa y, en el caso de un directorio, si se puede accesar a ese directorio. En el capitulo 3 se ampliará sobre este tema.

1.3.3 Seguridad en red: Cuando se enlaza una máquina a una red, al mismo tiempo que el sistema recibe y ofrece innumerables fuentes de información y servicios, queda también expuesta a todos los posibles ataques desde el exterior. El control de la red es uno de los aspectos fundamentales de seguridad hoy en día, puesto que la mayoría de los intentos de ataque se llevan a cabo a partir de posibles agujeros existentes en los sistemas y servicios de red. En el capítulo 4 se revisarán los puntos más críticos de seguridad en una red de un sistema abierto.

2 CONTROL DE ACCESO

Este capítulo permite ilustrar los beneficios de la seguridad que pueden ser concedidos en el esquema de control de acceso; mecanismo que utiliza el sistema multiusuario **UNIX**. El objetivo no solo se enfoca en el sistema de archivos, sino también en controlar el uso de los otros recursos tales como ***sockets***, impresoras, y en un caso más especial una máquina puede comprometerse en cualquier llamada al sistema.

El control de acceso es un esquema que ayuda a los grandes sistemas operativos en su flexibilidad y resolución, tal como en el sistema **UNIX**. Este consiste en una lista de diferentes privilegios para diferentes objetos; usualmente estos objetos son usuarios, archivos; que tienen privilegios de lectura, escritura o ejecución.

Inicialmente se cubren aspectos conceptuales de acceso en cualquier sistema, donde el principal elemento es la clave; a continuación se ilustran los mecanismos de control de acceso en un ambiente **UNIX**, y posteriormente se hace revisión de algunas herramientas usadas para atacar y/o proteger estos mecanismos en **UNIX**.

2.1 CONCEPTOS GENERALES

2.1.1 Control de acceso de un sistema: En muchas ocasiones ha sido posible obtener acceso no autorizado a los sistemas de cómputo, permitiéndole a los intrusos en cuestión, utilizar recursos a los que no debería tener acceso, o incluso realizar actos dañinos como robar o destruir información. Por esta razón, es muy importante tener en cuenta que el conectarse a una red, y sobre todo a una red de alcance global como Internet, trae también muchos problemas de seguridad.

En cualquier sistema de red se deben usar ***passwords*** para tener acceso a diversos recursos computacionales. Es de mucha importancia una buena selección y un uso adecuado de ellos; ya que existen una cantidad de riesgos potenciales de los que se debe estar alerta.

Todo sistema cuenta con un archivo de ***passwords***, el cual es el centro de atracción y punto inicial de ataque de "Intrusos". Afortunadamente a través del mecanismo de la ***criptografía***, estos ***passwords*** son almacenados de una manera secreta ó encriptada de tal forma que no sean legibles. Pero aún así, los administradores de los sistemas deben proteger dichos archivos; ya que si el contrincante logra capturar una copia, después se puede tomar todo su tiempo para tratar de romper por lo menos uno de los ***n passwords*** que contiene el archivo, y así lograr accesar a información confidencial, usar recursos computacionales no autorizados o falsificar una autorización.

Una estrategia de ruptura de los ***passwords***, sería usar un mecanismo de "fuerza bruta" a prueba y error en la cual se verifican todas las posibles combinaciones de ***passwords*** a través de un programa automático fácil de realizar. El problema de ésta estrategia es de tiempo, ya que si un sistema cuenta con ***passwords*** de 8 caracteres, esto implicaría $(256)^8$ ó $(2)^{64}$ posibles combinaciones lo cual implicaría un esfuerzo computacional bastante costoso, ya que usando el PC de mayor velocidad, esta tarea podría demorar miles de años.

Una mejor estrategia, la logran los Intrusos usando programas de dominio público llamados ***"crackers"***, los cuales toman como entrada información relevante del usuario al que se le quiere romper el ***password***, tales como: Nombre, apellido, apodo, nombre de los familiares más allegados, números de teléfonos, números de documentos, placa del carro, fechas de cumpleaños y otros. Estos programas generan solo las combinaciones que estén más relacionadas con la información del usuario y una cantidad tal para hacer una corrida en un tiempo limitado (ejemplo: 3 horas). De esta manera se logran mejores resultados, tal como lo muestra un informe de una prueba realizada a un servidor en Internet, en el que se obtuvo una efectividad del 25% (De cada cuatro intentos, un ***password*** fue roto), a sabiendas de que con un único acierto el contrincante logra su objetivo.

Existen otras estrategias menos técnicas pero muy comunes: Observar mientras una persona digita su ***password***, capturar ***passwords*** anotados en alguna parte de la oficina, usar un caballo de Troya (un programa utilizado para engañar

usuarios, ya que lo que muestra es distinto a lo que hace), fingir ser un usuario legítimo y llamar al administrador del sistema para obtener un nuevo ***password***.

2.1.1.1 Técnicas de intrusión: El objetivo de un intruso es ganar acceso al sistema y aumentar la gama de privilegios accesibles sobre el mismo. Generalmente, esto requiere que el intruso adquiera información que debe estar protegida. En la mayoría de los casos, esta información está protegida con un ***password*** de usuario. Con el conocimiento del mismo, el intruso puede entrar al sistema y ejercer todos los privilegios del usuario legítimo.

Típicamente, un sistema debe mantener un archivo que asocia un ***password*** con cada usuario autorizado. Si tal archivo se almacena sin la protección necesaria, entonces es blanco fácil para ganar acceso una vez aprendidos los ***passwords***. El archivo de ***password*** puede protegerse de una o dos maneras:

- Encripción de una sola dirección: El sistema almacena la clave del usuario en forma encriptada. Cuando el usuario presenta un ***password***, el sistema lo encripta y lo compara con el valor almacenado. En la práctica, el sistema comúnmente desempeña una transformación de una sola dirección (no reversible) en la que el ***password*** se usa para generar una clave para la función de encripción y se produce una salida de longitud fija.

- Control de acceso: El acceso al archivo de ***password*** se limita a una o unas muy pocas cuentas.

Si existen una o ambas medidas, se requiere de mucho esfuerzo para que el intruso potencial aprenda los ***passwords***.

2.1.1.2 Autenticación de un usuario: Como se puede notar, pueden existir algunos ataques no técnicos para los cuales su solución es también no técnica.

La primera meta de cualquier sistema de seguridad, es asegurarse de que los usuarios que están en el sistema son los que realmente pueden estar y verificar que no hagan cosas incorrectas.

En un sistema, cada usuario es identificado por una entidad conocida comúnmente como "***clave***" del usuario, y que está formada por dos elementos:

- El nombre del usuario, mediante el cual se conoce públicamente al usuario, tanto por parte de la máquina como por parte de otros usuarios.
- El ***password***, que sirve para que el usuario, cada vez que quiera hacer uso del sistema, le demuestre a la máquina que se trata efectivamente de él y no de alguien que está queriendo tomar su lugar. Por esta razón, obviamente, debe

ser completamente confidencial, pues cualquiera que la conozca puede hacerse pasar por el usuario y utilizar sus recursos.

El programa ***login*** verifica el nombre del usuario y el ***password*** usado. Si el nombre del usuario no está en el archivo de ***passwords*** o el ***password*** no es correcto, el programa niega el acceso a la máquina. Cuando el usuario suministra los datos correctos, el sistema le permite accesar a la máquina.

2.1.2 Control de cuentas y password: La protección de las cuentas de los usuarios es la primera línea de defensa contra Intrusos, puesto que las combinaciones ***login/password***, que son fácilmente adivinables, son los principales puntos de quiebre de un sistema.

Es conveniente que los usuarios del sistema usen buenos ***passwords***, ya que estos se convierten quizás en el elemento individual más importante en la seguridad de un sistema de cómputo.

Usualmente un ataque comienza cuando se captura una cuenta en alguna máquina; ya que a partir de ese punto, es mucho más fácil obtener cualquier tipo de acceso a los recursos.

Por lo anterior es fundamental un buen control de los ***passwords*** y de las posibles cuentas que no estén siendo usadas. Algunas de las recomendaciones que los administradores de seguridad deben dar a los usuarios son las siguientes:

- No utilizar el nombre de la cuenta (***login***), tal cual, al revés, en mayúsculas.
- No utilizar nombres personales, ni apellidos.
- No utilizar nombres de familiares.
- No emplear un ***password*** sólo con números o con la misma letra.
- No emplear palabras que se encuentren en los diccionarios.
- No utilizar palabras de menos de seis letras.
- No escribir los ***passwords*** en ningún sitio, ni en archivos.
- No usar información personal, ni combinaciones de estas: Nombres, números de documentos, teléfonos, placa del carro y fechas de cumpleaños.
- No dar el ***password*** a otra persona

Una buena forma de seguir estos consejos es:

- Tomar una línea de un poema o canción y usar la primera letra de cada palabra.
- Poner alternativamente entre una consonante y una vocal, dos vocales, esto proporciona palabras que no tienen sentido y que normalmente no son pronunciables, como ***"routboo"***, ***"quadpop"*** y así.
- Elegir dos palabras cortas y unirlas por un símbolo de puntuación.

- Mezclar letras mayúsculas, minúsculas y números.
- Utilizar palabras que sean fáciles de recordar para que no haya que escribirlas.
- Emplear un ***password*** que pueda ser tecleado rápidamente sin necesidad de mirar al teclado.
- Usar ***passwords*** mínimo de 8 caracteres.
- Cada vez que digite el ***password***, asegurarse que no lo estén observando.
- Cambiarlo periódicamente (2 meses es un tiempo prudente).
- Cuando se tenga incertidumbre de que alguien obtuvo el ***password***, hay que comunicarse con el administrador del sistema para que éste le saque un registro de los últimos accesos y confrontarlos con la realidad.
- Para cada sistema se debe usar un ***password*** diferente.

El problema de no seguir estas reglas es que existen programas que van probando ***passwords*** hasta que encuentran el correcto. Una vez escogido un ***password*** es importante seguir una política adecuada, unas reglas generales pueden ser:

- El único lugar para almacenarlos es la memoria del dueño.
- No decir nunca el ***password*** a otra persona.
- Establecer un tiempo máximo de uso para cada ***password***.

2.1.3 Estrategias para selección de claves: La mayoría de los usuarios escogen claves que son muy cortas o muy fáciles de adivinar; por otro lado si las claves están conformadas por ocho caracteres seleccionados aleatoriamente, será

casi imposible recordar la clave. Existen cuatro técnicas para eliminar las claves fáciles:

- Educación del usuario: Enseñar a los usuarios los métodos para la selección de claves difíciles de adivinar.

- Claves generadas por computador: Son generalmente difíciles de recordar.

- Chequeo de claves reactivo: Se corre un programa para encontrar claves. El sistema cancela la clave que haya sido adivinada y notifica al usuario.

- Chequeo de claves proactivo: Bajo este esquema el sistema chequea cada nueva clave, verificando su vulnerabilidad y longitud. Si el sistema usa algoritmos simples para declarar claves como aceptadas, esto proveerá una pauta a los intrusos para refinar sus técnicas. Las reglas pueden forzar por ejemplo a que todas las claves sean de ocho caracteres, a que incluyan al menos una mayúscula, una minúscula, dígitos numéricos y signos de puntuación.

2.2 CONTROL DE ACCESO EN UNIX

2.2.1 Mecanismo de control de acceso: Para entender la naturaleza del ataque, considérese un esquema que es ampliamente usado sobre sistemas

UNIX, en el cual las claves nunca son almacenadas en texto claro. En vez de eso, se usan los procedimientos que se esquematizan a continuación:

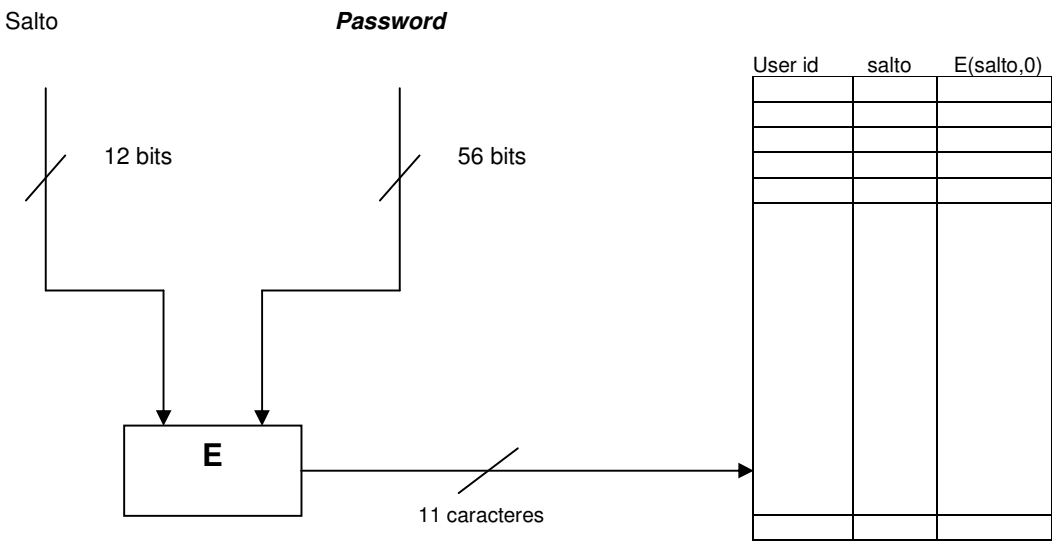
2.2.1.1 Almacenamiento de un nuevo password: Cada usuario selecciona una clave de ocho caracteres, estos caracteres son convertidos a un valor de 56 bits que sirve como clave de entrada a una rutina de encriptamiento, basada en ***DES*** (***ver figura 2***). El "SALT" es un valor de 12 bits. Éste valor se refiere a la fecha de asignación de la clave del usuario. El ***DES*** es ejecutado con una entrada que consta de un bloque de 64 bits de ceros. La salida del algoritmo sirve como entrada a un segundo encriptamiento. Este proceso se repite 25 veces, para un total de 25 encriptamientos. La salida de 64 bits resultantes se traduce a una secuencia de 11 caracteres. Se almacena la clave cifrada junto con una copia del "SALT" (clave adicional) en el archivo de claves del usuario correspondiente.

El "SALT" sirve para evitar duplicación de claves, incrementar la longitud de la clave sin requerir que el usuario recuerde los caracteres adicionales y evitar el uso de una implementación ***DES*** en hardware, lo cual mitiga la dificultad de un ataque de fuerza bruta.

2.2.1.2 Verificación de un password: Cuando un usuario intenta entrar a un sistema **UNIX** el usuario suministra un ID (identificador de usuario) y una clave. El sistema operativo usa el ID para indexar un archivo de claves y recuperar la clave

extendida y la clave encriptada (***ver figura 3***). La clave y su extensión son usadas como entradas en la rutina de encriptamiento. Si el resultado concuerda con la clave encriptada, la clave es aceptada.

La rutina de encripción está diseñada para desalentar los ataques a prueba y error. Las implementaciones del ***DES*** en software son lentas comparadas con las implementaciones en hardware y el uso de 25 iteraciones multiplica el tiempo requerido por 25.



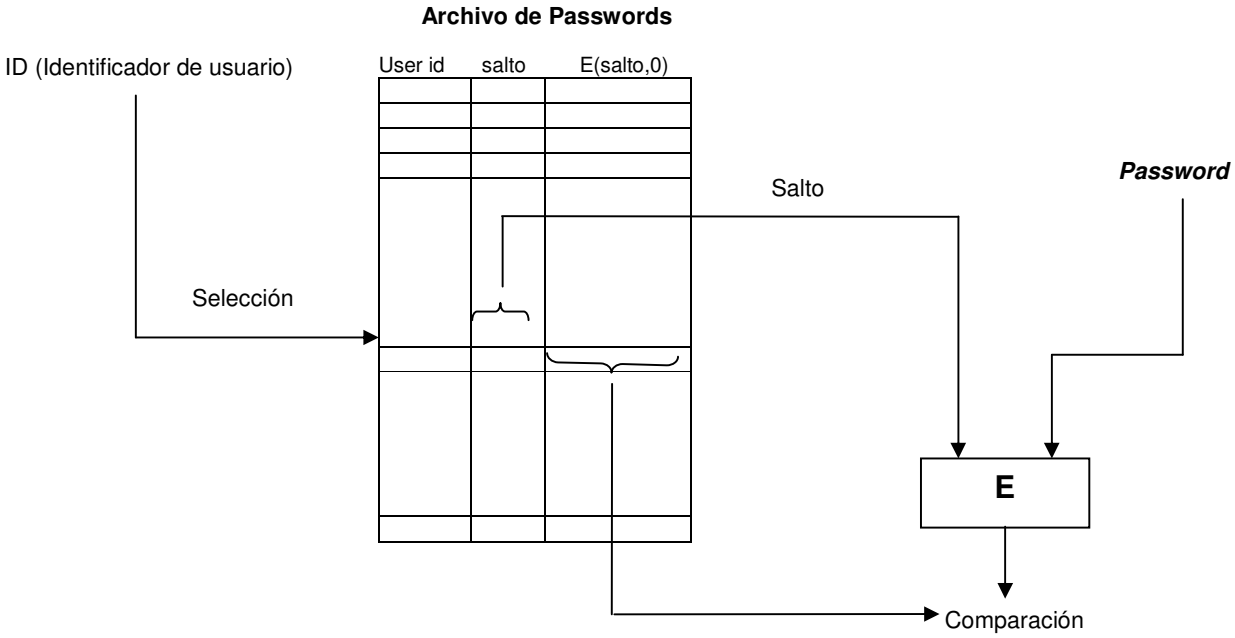


Figura 3. ***Verificación de un password***

2.2.2 Archivos de passwords: La información fundamental de los ***passwords*** se almacena en UNIX en el archivo ***/etc/passwd*****,** el cual contiene una línea por cuenta con el siguiente formato:

<usuario>:<x>:<UID>:<GID>:<comentario>:<directorio home>:<Shell>

- ***Usuario***: Es el nombre de la cuenta o ***login*** del usuario.

- ***x***: Se utiliza para indicar si la cuenta tiene ***password***. Anteriormente, en versiones viejas de **UNIX**, se usaba este campo para colocar el ***password*** encriptado.
- ***UID***: ***User ID*** (Identificador de usuario) Cada usuario tiene este identificador único que sirve para mostrar los dueños de los archivos.
- ***GID***: ***Group ID*** (Identificador de grupo). Cada usuario tiene al menos un grupo de usuarios con similares privilegios. Este campo indica el ***group ID*** primario.
- ***Comentario***: Generalmente se usa para colocar la descripción o nombre completo del usuario.
- ***Directorio home***: Es el directorio en el que es colocado el usuario cuando se conecta.
- ***Shell***: El tipo de comando ***shell*** que usará el usuario cuando se conecte (***BourneShell, KornShell, Cshell***).

El archivo ***/etc/shadow*** es un archivo complementario al ***/etc/passwd***. Su función es almacenar los ***passwords*** de usuario de una manera encriptada. Por cada línea que exista en ***/etc/passwd*** debe existir una línea en ***/etc/shadow***. Este archivo tiene la siguiente estructura:

<usuario>:Password encriptado::::

- ***Usuario***: Es el nombre de la cuenta o ***login.***
- ***Password encriptado***: Indica el ***password*** que ha sido encriptado.

Cabe anotar que el acceso a este archivo es restringido, ya que es peligroso que cualquier usuario vea o copie los ***passwords*** encriptados.

Como se puede ver, el nombre del usuario y la demás información se almacena en el archivo ***/etc/passwd***, mientras que el ***password*** encriptado es autoalmacenado en el archivo ***/etc/shadow***.

Como se explicó en el numeral anterior, con el objeto de autenticar un usuario, el sistema **UNIX** encripta los ***passwords*** utilizando el algoritmo ***DES***, generando una cadena de 11 caracteres, la cual es comparada con la cadena encriptada almacenada en el archivo ***/etc/shadow***. Este hecho hace necesario la utilización del comando ***pwconv***, el cual mueve los ***passwords*** encriptados al archivo ***/etc/shadow*** al que sólo tiene acceso el superusuario; también se puede utilizar el comando ***pwck*** para verificar que todas la cuantas estén configuradas correctamente, además chequea todas las cuentas que tienen ***login*** en blanco en el archivo ***/etc/passwd***, lo cual puede permitir a alguien conectarse sin tener un ***login*** válido.

En versiones primitivas de **UNIX**, de las cuales algunas se encuentran aún en el mercado, el ***password*** encriptado reside únicamente en el campo de contraseña del archivo ***/etc/passwd***. Debido a que este esquema no utiliza el archivo ***/etc/shadow***, esto se convierte en un gran problema de seguridad ya que el archivo ***/etc/passwd*** requiere ser accesible por todos los usuarios del sistema, lo

cual permitiría a un intruso hacer una copia remota de los ***passwords*** encriptados para tratar de descifrar algunos de estos de una manera imposible de detectar, ya que el ataque puede ser realizado utilizando herramientas para romper ***passwords*** en una máquina aislada.

2.2.3 Archivos de grupos: La información de los grupos es almacenada en **UNIX** en el archivo ***/etc/group***. Este archivo contiene la siguiente estructura:

<grupo>:<GID>:<usuarios>

- ***Grupo***: Nombre del grupo.
- ***GID***: ***Group ID*** (Identificador de grupo).
- ***Usuarios***: Es la lista de usuarios para los cuales este grupo sería secundario. La lista debe estar separada por comas.

2.2.4 Cambio de password: Todos los usuarios en **UNIX** tienen asignado un ***password*** o contraseña que puede cambiarse con el comando ***passwd***. Este comando pregunta por el nuevo ***password*** (dos veces, para asegurarse que tecleó el ***password*** correcto). El usuario teclea el nuevo ***password***, pero no se muestra en pantalla por razones de seguridad. El nuevo ***password*** surte efecto desde el momento en que se cambia. Si el usuario entra al sistema de nuevo, se le preguntaría por este nuevo ***password*** para permitir el acceso.

2.2.5 Cuenta del superusuario

Los sistemas **UNIX** proveen una cuenta ***root*** o de superusuario, que permite acceso a todo el sistema, esta cuenta es usada para funciones de administración del sistema, por lo tanto su ***password*** debe ser conocido por el menor número de personas posible.

La cuenta del ***root*** es usada para cumplir funciones básicas y tiene amplio control sobre el sistema operativo. Accesa y ejecuta esenciales programas en el sistema. Por esta razón no hay restricciones de seguridad para cualquier programa que es ejecutado por el ***root***.

Es posible especificar desde qué terminales distintas a la consola de la estación de trabajo puede conectarse el superusuario. Cuando se deshabilitan conexiones remotas, la única forma de conectarse como superusuario desde una estación remota es conectándose inicialmente con el ***login*** de algún usuario distinto del superusuario y después usar el comando ***su*** en caso de que esté disponible.

Distintas versiones de **UNIX** tienen diferentes métodos para deshabilitar conexiones remotas de superusuario. Para DEC OSF/1 y SunOS 4.x se hace modificando el archivo ***/etc/ttys***. En sistemas SunOS 5.x y SGI IRIX en el archivo ***/etc/default/login***.

El directorio ***/etc/default*** contiene archivos ASCII que son usados para controlar y monitorear el acceso a ***root***:

- Cualquier entrada en el archivo ***/etc/default/login***, determina las restricciones de acceso a ***root***.
- Las entradas en el archivo ***/etc/default/su***, determina las condiciones por defecto del comando ***su***.
    - ✓ Una entrada permite o niega una conexión cada vez que el comando ***su*** es usado para cambiar a otro usuario. Un registro de cada vez que se usa el comando ***su***, quién lo usa, y cuándo lo usó, se almacena en el archivo ***/var/adm/sulog***, permitiendo rastrear quien está usando la cuenta del superusuario.
    - ✓ Otra entrada permite o niega una visualización en la pantalla cada vez que se intenta usar el comando ***su*** para ganar privilegios de ***root*** cuando se conecta remotamente.

2.2.6 Otros puntos a tener en cuenta en el control de acceso UNIX

2.2.6.1 Shell restringido: El ***shell*** estándar permite a un usuario abrir archivos, ejecutar comandos y otras operaciones. El ***shell restringido*** puede ser usado para limitar la capacidad de un usuario de cambiar directorios y ejecutar comandos. El ***shell restringido*** **(*rsh*)** se encuentra en el directorio ***/usr/lib*** a

diferencia del ***shell*** remoto ***rsh*** que se encuentra en el directorio ***/usr/sbin/rsh***. El ***shell restringido*** difiere del normal en lo siguiente:

- El usuario es limitado al directorio ***home*** (no puede usar el comando ***cd*** para cambiar de directorio)
- El usuario puede usar solamente comandos que se especifiquen en la variable ***path*** la cual no puede ser cambiada por él.
- El usuario puede accesar solamente a archivos en el directorio ***home*** y en los subdirectorios siguientes.
- No puede redireccionar la salida usando ***>>*** o ***>.***

El ***rsh*** no es completamente seguro, sin embargo, es utilizado para que ciertos usuarios no causen problemas en el sistema

2.2.6.2 Registro de conexiones falsas: Se crea un registro para rastrear los intentos de conexiones no exitosas al sistema. Después de 5 intentos consecutivos sin éxito, se almacena el nombre del usuario en el archivo ***/var/adm/loginlog.***

2.2.6.3 Conexiones especiales: Existen dos formas comunes de accesar a una máquina, usando el ***login*** del usuario o el del ***root***. Un número especial de

conexiones al sistema permite a un usuario ejecutar comandos administrativos sin utilizar cuenta ***root***. La siguiente tabla muestra algunas cuentas:

Tabla 1. Conexiones especiales.

Nombre de la Cuenta	GID	Uso
root	0	Casi no tiene restricciones y sobrepasa a las demás cuentas en protección y permisos. La cuenta ***root*** tiene acceso al sistema entero. La cuenta para el ***root*** debe ser bien protegida.
daemon	1	Controla procesos en segundo plano
bin	2	Posee los principales comandos
sys	3	Posee muchos sistemas de archivos
adm	4	Posee algunos archivos administrativos
lp	71	Posee objetos y archivos de datos para impresión
uucp	5	Posee objetos y archivos de datos para ***uucp***.
nuucp	9	Es usado por maquinas remotas para conectarse al sistema e iniciar transferencia de archivos.

2.2.6.4 Listas de control de acceso: El sistema de archivos de **UNIX** usa un esquema de bases de atributos para implementar la seguridad. Además no es

aceptable obtener acceso al sistema fuera del sistema de archivos, por ejemplo por medio de los ***sockets***.

Muchos demonios son privilegiados con el ***ID*** del ***root*** porque ellos tienen que abrir puertos registrados y ser capaces de usar ciertos recursos del sistema. El esquema es que el proceso comience con los privilegios del ***root*** y entonces cambie el ***ID*** del ***root*** al del usuario que lo inició, esto se puede permitir porque se usan listas de control de acceso.

El propósito fundamental del sistema es controlar el acceso a los recursos; esto no limita a que esos recursos sean archivos y directorios; también las capacidades de procesamiento y funciones de red.

El eje central del sistema control de acceso, son las listas de acceso. Algunos recursos o grupos de recursos pueden ser delegados a controlar sublistas de acceso. Las listas son leídas de principio a fin y si se encuentra una sublista de acceso es inmediatamente procesada.

Existe un método que permite la administración centralizada del control de acceso a los recursos del sistema, y es adicionando reglas a los recursos; en dichas reglas, se especifica quién puede entrar y quien no al recurso solicitado.

Este tipo de esquema hace más robusto al sistema en el aspecto de la seguridad, por lo que hace imposible a los usuarios accesar a datos a los que ellos no tienen

permisos. Es importante hacer énfasis en que el sistema tiene el control de acceso centralizado a todos los recursos.

En **UNIX** ***Solaris 2.6*** existen comandos como ***setfacl*** y ***getfacl*** los cuales configuran y muestran las listas de control de acceso de los diferentes archivos.

2.2 HERRAMIENTAS UNIX PARA EL CONTROL DE ACCESO

Existen herramientas para proteger y/o atacar el mecanismo de control de acceso a sistemas **UNIX**, basadas en la verificación de listas de control de acceso o comandos como ***login*** y ***su***. Algunas de estas se mencionan a continuación:

2.3.1 Crack: Herramienta adivinadora de ***passwords***, desarrollada por ***Alec Muffett*** y que fue diseñada para localizar inseguridades en **UNIX** o en otros archivos de ***passwords***, pero escaneando el contenido del archivo de ***passwords***, buscando usuarios que hayan escogidos ***passwords*** débiles.

2.3.2 Kerberos: Diseñada e implementada por ***Barry Jaspan***, es un sistema de autenticación de red para uso físico en redes inseguras, basado en el modelo de distribución de claves permite comunicación entre entidades de la red para mejorar la identidad de cada una, mientras previene caídas o rechaza ataques; también ayuda a la integridad de los datos por detección y modificación, lecturas no autorizadas.

2.3.3 Permissions: Es un básico ambiente ***BSD*** permite solamente tres utilidades para los usuarios dentro de la máquina, ***login***, ***rshd***, ***ftpd***. Esos tres programas son utilizados para chequear las solicitudes a los recursos los cuales determinan, a donde se le permite entrar a un usuario

2.3.4 Skey: Desarrollada por ***Neil M. Haller*** y ***Philip R. Karn***, ésta herramienta una vez que se ingresa al sistema, ayuda a la autenticación de los ***passwords*** en la red que ha sido atacada.

2.3.5 COPS (Computer Oracle and Password System): Implementada por ***Dan Farmer*****,** herramienta que examina el sistema para cierto número de debilidades y alerta al administrador del sistema de ello; en algunos casos este puede automáticamente corregir el problema.

3 PROTECCIÓN DE LA INTEGRIDAD Y CONFIDENCIALIDAD DE ARCHIVOS

El sistema **UNIX** es un sistema multiusuario, el cual tiene la capacidad de que todos los usuarios conectados a la máquina puedan leer y usar archivos de otros usuarios; debido a esto, estos archivos deben ser protegidos contra cualquier uso indebido.

Es de gran importancia asegurarse de que los archivos están siendo accesados de una forma segura; esto se logra si se tienen las configuraciones apropiadas de estos archivos, contando con una buena política para la protección de la confidencialidad e integridad de los mismos.

Este capítulo describe aspectos fundamentales a tener en cuenta para que los archivos y directorios en un sistema de archivos en una estación de trabajo bajo ambiente **UNIX**, se configuren adecuadamente.

3.1 ESTRUCTURA DE ARCHIVOS Y DIRECTORIOS.

3.1.1 Sistema de archivos: Un ***filesystem*** o sistema de archivos es una estructura de datos o una colección de archivos, que puede referirse a dos aspectos distintos: El árbol de directorios y los arreglos de archivos sobre las particiones del disco.

El ***filesystem físico*** es dividido en particiones de disco; el tamaño de la partición determina el número de bloques que el ***filesystem*** usa. Cada partición tiene un superbloque, un **ínodo** y bloques de datos. El superbloque mantiene la información del sistema; el **ínodo** contiene información similar para archivos individuales y los bloques de datos mantienen la información en los archivos.

El ***filesystem lógico*** se refiere a la arquitectura de los directorios conectados o particiones del disco que son accesibles por el usuario. El **UFS (UNIX FILESYSTEM)** es un arreglo en forma de árbol o pirámide invertida, donde todos los archivos están contenidos en el directorio del superusuario. Hay diferencia entre los tipos de filesystem de **UNIX** pero básicamente conservan la misma filosofía, ya que dentro de cada directorio se encuentran subdirectorios.

3.1.2 Directorios: Los directorios forman una estructura jerárquica, las cuales organizan las utilidades y herramientas del sistema operativo **UNIX**. El directorio de más alto nivel en **UNIX** se llama ***root***, y consiste en un grupo de subdirectorios agrupados según su función.

3.1.2.1 Tipos de directorios: Cada directorio en el sistema operativo **UNIX** se encuentra definido de la siguiente manera:

- /: El directorio ***root*** o raíz. Es la base de la estructura del árbol del sistema de archivos. Todos los demás archivos y directorios, independientemente de su localización física, están contenidos en este directorio.
- /**bin**: Es el directorio de comandos binarios. Incluye los archivos ejecutables públicos que son parte del sistema operativo **UNIX**.
- /**sbin:** Es el directorio de archivos ejecutables de administración y funcionamiento del sistema.
- /**dev**: Directorio de dispositivos, contiene archivos especiales tales como de impresoras, discos, consola, y otras.
- /**etc**: Contiene los archivos de administración y configuración del sistema.
- /**lib**: Este directorio contiene las librerías para C y otros lenguajes.
- /**lost+found**: Directorio de archivos perdidos. Los errores de disco o un apagado incorrecto del sistema puede causar que los archivos se pierdan (archivos perdidos se refiere a sitios en disco que están marcados como

usados en las estructuras de datos de los discos, pero que no están listados en ningún directorio.

- /**mnt**: Directorio de montaje. Este directorio está vacío y se usa para montar temporalmente particiones de discos.
- /**home**: Directorio hogar de los usuarios.
- /**usr**: Contiene directorios para las colas de impresión, correo, programas generados localmente, ejecutables para los comandos de usuarios.
- /**tmp**: Directorio temporal, disponible para todos lo usuarios como directorio para archivos transitorios.
- /**var**: Guarda los directorios volátiles de las colas del sistema (impresión, correo, mensajes).
- La notación "**.**" significa el directorio actual, mientras que la notación "**..**", significa el nivel anterior, es decir el directorio padre.

3.1.3 Ínodos: Los enlaces permiten dar a un único archivo múltiples nombres. Los archivos son realmente identificados por su número de **ínodo**. Un directorio no es otra cosa que una lista de números de **ínodo** con sus correspondientes nombres de archivos. Cada nombre de archivo en un directorio es un enlace a un **ínodo** en particular.

Un **ínodo** contiene los siguientes campos:

- Modo de acceso
- ***Id*** del dueño, ***Id*** del grupo
- Contador de enlace
- Tamaño del archivo
- Números de bloques usados

3.1.4 Propiedades de los archivos: El sistema operativo **UNIX** provee un mecanismo llamado permisos de archivos, que permite a los archivos ser propiedad de un determinado usuario. Generalmente, los usuarios pueden leer los archivos de dueños diferentes, pero no se permite que los modifiquen. En **UNIX** cada archivo tiene asociado los siguientes elementos:

- Tipo de archivo.
- Permisos otorgados al dueño, grupo y cualquier otro usuario del sistema.
- Dueño del archivo.
- Grupo dueño del archivo.

3.1.4.1 Tipos de archivos: En **UNIX** la estructura de directorios se encuentra clasificada de acuerdo con los tipos de archivos que en el sistema se encuentren; existe un caracter que define el tipo de archivo. A continuación se describen los tipos de archivos que existen en **UNIX** conjuntamente con el caracter que los representa:

- Socket: s.
- Enlace simbólico: l.
- Directorio: d.
- Dispositivo de caracteres: c.
- Dispositivo de bloque: b.
- Archivo corriente: -.

3.1.4.2 Permisos de archivos: Cada archivo en **UNIX** tiene asociado una serie de permisos que le permiten ser accesados o no por los usuarios. Los permisos de archivos caen dentro de tres grupos:

- ***Lectura***: Permite a un usuario leer el contenido de un archivo, o si es un directorio, listar su contenido con ***ls***.
- ***Escritura***: Permite escribir y modificar un archivo. Para los directorios, este tipo de permiso permite crear nuevos archivos o borrar archivos dentro de ese directorio.
- ***Ejecución***: Permite al usuario ejecutar un archivo. Para directorios, este permiso permite que el usuario pueda ubicarse en el directorio en cuestión usando el comando ***cd***.

Existen tres clases de usuarios para los cuales se les asignan cada uno de estos permisos:

- El propietario del archivo
- El grupo al cual pertenece el archivo
- Todos los usuarios, independientemente del grupo.

El comando ***ls*** con la opción ***-l*** muestra, entre otra información, los permisos de la siguiente manera:

- r w x r - x r - -

- El primer carácter de la cadena de permisos ("-") representa el tipo de archivo.
- Las tres letras siguientes ***("rwx")***, representan los permisos otorgados al dueño del archivo. La ***"r"*** representa lectura, la "***w***" representa escritura, y la "***x***" representa ejecución.
- Los siguientes tres caracteres ***("r-x")***, representan los permisos del grupo en este archivo. Como solo aparece una ***r*** y una ***x***, cualquier usuario que pertenezca al grupo en cuestión sólo podrá leer el archivo o ejecutarlo, pero no modificarlo o borrarlo.
- Los últimos tres caracteres ***("r--")***, representan los permisos que tienen cualquier otro usuario en el sistema diferente al dueño del archivo o aquellos que pertenezcan al grupo. En este caso, como solo aparece una ***r***, otros usuarios podrán leer el archivo pero no escribir en él o modificarlo.

Un aspecto fundamental a tener en cuenta es que los permisos de un archivo también dependen de los permisos del directorio en el que se encuentra el archivo. Es decir, que para poder acceder a un archivo, primero se debe tener acceso de ejecución a todos los directorios que se encuentren en el ***path*** del archivo y, por supuesto, se debe tener el permiso respectivo en ese archivo.

3.1.4.3 Dueños y grupos dueños de archivos: A cada archivo en **UNIX** se le asigna un dueño individual y un grupo; estos solo pueden ser cambiados por el dueño del archivo y el superusuario.

Cuando se crea un archivo en **UNIX**, el usuario que creó el archivo será su dueño y el grupo principal al que pertenece el usuario que lo creó será el grupo dueño del archivo.

Para cambiar el dueño de un archivo o directorio, **UNIX** cuenta con la utilidad ***chown***. Mediante el comando ***chgrp*** se puede cambiar el grupo dueño de un archivo o directorio.

3.2 PROTECCIÓN DE ARCHIVOS Y DIRECTORIOS

La seguridad en el sistema de archivos es factor fundamental para la protección de la integridad y confidencialidad de los datos. Como se explicó en el numeral anterior, el sistema **UNIX** incorpora tres conjuntos de bits asociado a los permisos: Uno del usuario, dueño del grupo, otro para el grupo al que pertenece y un último

conjunto que indica los permisos que se le conceden a cualquier usuario del sistema. Cada conjunto tiene los tres bits conocidos de permisos; uno de lectura, otro de escritura y otro de ejecución. El significado varía si hace referencia a un directorio o a un archivo. La primera norma de seguridad en el sistema de archivos es dar los permisos adecuados a cada archivo y usuario.

3.2.1 Filosofía de protección de archivos y directorios en UNIX: Debido a que **UNIX** es un sistema operativo multiusuario, es necesario identificar cada directorio y archivo con una serie de permisos, los cuales, como ya se ha mencionado, permiten o no el acceso a la información contenido en ellos.

Cada usuario en **UNIX** tiene privilegios en común con otros usuarios. Estos se agrupan y se les especifica un identificador de grupo (***GID***), que lo designa como **grupo primario**. Cuando un usuario puede accesar a ciertos archivos pertenecientes a otro grupo diferente al de él, éste grupo aparecerá para el usuario en cuestión como **grupo secundario**. El superusuario es el único que tiene acceso a todos los archivos y directorios del sistema, sin importar qué permisos tengan configurados.

Es fundamental tener en cuenta para una correcta configuración de los archivos y directorio los siguientes criterios:

- Si un usuario tiene permiso de lectura puede leer el contenido de un archivo, o si se trata de un directorio, el usuario puede listar el contenido de éste con la utilidad ***ls***.

- Cuando a un usuario se le concede el permiso de escritura a un archivo, se le permite modificarlo. Para el caso de los directorios, este tipo de permiso permite crear o borrar archivos dentro de ese directorio.
- El permiso de ejecución permite al usuario ejecutar un archivo. En el caso de un directorio, este permiso permite al usuario ubicarse en el directorio en cuestión mediante la utilidad ***cd***.

Es de gran importancia recalcar que los permisos que se le conceden a un archivo dependen de los permisos del directorio en donde se encuentra el archivo. Por ejemplo, para accesar a un archivo hay que tener permiso de ejecución en todos los directorios que se encuentren en la dirección del archivo y lógicamente tener el permiso respectivo para ese archivo.

3.2.2 Configuración de los permisos de archivos: Cuando se crea un archivo en **UNIX**, estos se crean sin tener en cuenta los derechos que se les da. Como primera medida de seguridad, es importante dar los permisos adecuados a cada archivo. Estos permisos pueden ser modificados sólo por el propietario del archivo o el superusuario. **UNIX** provee la utilidad ***chmod*** que nos permite alterar los permisos de un archivo. El comando ***chmod*** se usa de la siguiente forma:

chmod uvw <nombre del archivo>

Donde ***u***, ***v*** y ***w*** representan un código numérico entre ***0*** y ***7*** para los permisos del dueño, grupo y otros respectivamente. Los valores de las letras ***r***, ***w*** y ***x*** son: ***r = 4***, ***w = 2***, ***x = 1*** y la suma de dichos valores representan todas las posibles combinaciones de los mencionados permisos. Por ejemplo, si se desea que un

archivo tenga los permisos solamente de escritura para el propietario del archivo, es decir, --w-------, se procede de la siguiente forma:

$ chmod 200 <nombre del archivo>

Para ser más claro, en la siguiente tabla se resume este aspecto:

Tabla 2. Permisos de Archivos

Dueño	Grupo	Todos	Comando
Rwx	rwx	rwx	chmod 777 filename
Rwx	rwx	r-x	chmod 775 filename
Rwx	r-x	r-x	chmod 755 filename
rw-	rw-	r--	chmod 664 filename
rw-	r--	r--	chmod 644 filename

3.2.3 Permisos especiales: Existen tres tipos de permisos adicionales llamados permisos especiales que están disponibles para ejecutar archivos y directorios públicos. Cuando esos permisos son configurados, cualquier usuario puede ejecutar archivos, asumiendo los permisos del propietario para ejecutarlo.

3.2.3.1 Permiso setuid (set-user identification): Cuando el ***setuid*** es configurado sobre un archivo ejecutable, cambia el **user id** de la persona que ejecuta el archivo por los del dueño del archivo. Esto permite a cualquier usuario ejecutar archivos y accesar a directorios que solo son accesados por el propietario; por ejemplo, el permiso ***setuid*** en el comando ***passwd*** hace que un

usuario pueda cambiar su ***password***, asumiendo los permisos del superusuario; es decir, este permiso permite a un usuario normal ejecutar comandos a los cuales no tiene privilegios.

Cuando un usuario ejecuta el comando ***passwd*** (su dueño es el superusuario), este gana privilegios temporalmente para escribir en el archivo ***/etc/passwd*** que en condiciones normales el único que puede escribir es el superusuario. Una vez el proceso termina, el usuario retoma sus privilegios normales.

Esto representa un riesgo de seguridad, porque determinados usuarios pueden encontrar una forma de mantener los permisos obtenidos del proceso ***setuid*** aún cuando el proceso haya terminado su ejecución y así seguir teniendo los privilegios del dueño del archivo; para el caso del comando ***passwd***, seguir teniendo privilegios del superusuario.

3.2.3.2 Permiso setgid (set-group identification):

El permiso ***setgid*** es similar a ***setuid***, a excepción de que el proceso efectivo de identificación de grupo **(GID)**, es cambiado al propietario del grupo del archivo, y un usuario tiene acceso basado sobre los permisos obtenidos para tal grupo.

Cuando ***setgid*** es aplicado a un directorio, los archivos creados en ese directorio pertenecen al grupo del directorio, y no al grupo que pertenezca el proceso que lo creó. Cualquier usuario que tiene permiso de escritura en el directorio puede crear archivos ahí; sin embargo, el archivo no pertenecerá al grupo del usuario pero si pertenecerá al grupo del directorio.

3.2.3.3 Sticky bit:

El ***sticky bit***, es un bit de permiso que protege archivos dentro de un directorio. Si el directorio tiene el ***sticky bit*** configurado, un archivo solo puede ser eliminado por su propietario o por el superusuario. Esto previene a un usuario de borrar archivos de un directorio público (/**tmp**):

3.2.3.4 Configuración de los permisos especiales: La configuración de estos permisos constituyen un riesgo de seguridad, por lo tanto hay que tener mucho cuidado al momento de su configuración. Ejemplo: Si hay una versión ***suid*** para el ***csh***, cualquiera podría ejecutarlo y obtener privilegios del superusuario mientras dure el proceso.

Se debe monitorear el sistema para estar consciente de cualquier uso sin autorización de los permisos ***setuid*** y ***setgid*** que permiten obtener privilegios de superusuario.

Otro problema se puede presentar con el comando ***write***, programa que permite enviar mensajes entre usuarios conectados en otras terminales. Este comando puede correrse únicamente si los bits de permisos ***suid*** y ***sgid*** están activados. El problema radica en que mientras ***write*** está corriendo, el usuario ha aumentado sus privilegios. Se debe chequear que esté instalado así: dueño el grupo ***tty*** y con solamente el bit ***sgid*** activado.

Es conveniente analizar archivos que se encuentren con alguno de los bits de permisos activado y que no pertenezca a alguno de estos directorios (SunOS 5.x): ***/bin, /etc, /include, /usr/bin, /usr/openwin, /usr/sbin, /usr/ucv, /var/adm***.

Formato de configuración:

Permiso Especial	Permiso de Lectura	Permiso de Escritura	Permiso de Ejecución
4 Setuid 2 Setgid 1 Sticky bit	4	2	1

Ejemplo:

Setuid

```
# chmod 4555  herman

# ls –l herman
```

```
-r-sr-xr-x  1  her    staff      12095 May  23 10:15  herman
```

Setgid

```
chmod 2551  herman1

# ls –l herman1

-r-xr-s--x  1  her    herstaff  2500  May  23 10:15  herman1
```

Sticky bit

```
chmod 1777 direc

# ls –ld  direc

-rwxrwxrwt  1  root    root   512    May  23 10:15  direc
```

3.2.4 Variable umask: Cuando se crea un archivo o directorio en el sistema operativo **UNIX**, éste se crea con unos permisos por defecto los cuales se encuentran determinados por el valor de la variable ***umask*** que se encuentra en el directorio ***/etc/profile*** o en los archivos ***.cshrc*** o ***.login***. Por defecto los permisos para un archivo de texto son 666 obteniendo permisos de escritura para el propietario, el grupo y otros, y ***777*** para un directorio o un archivo ejecutable.

El valor asignado por ***umask*** es restado del que viene por defecto, esto tiene el efecto de negar permisos en la misma forma en que ***chmod*** los otorga. Por ejemplo: cuando el comando ***chmod 022*** garantiza permisos de escritura para el grupo y otros, ***umask 022*** niega permiso de escritura para grupo y otros.

3.3 UTILIDADES PARA LA PROTECCIÓN DE INTEGRIDAD DE ARCHIVOS.

3.3.1 Backups: Es necesario realizar copias de seguridad, guardarlas en áreas seguras, y tenerlas en diferentes medios. En **UNIX** existen muchos ventajas para resguardar archivos y escribirlos a cintas, inclusive usando compresión. También existen utilidades que ayudan a optimizar copias de respaldo. Estas utilidades nos permiten programar copias periódicamente.

Algunas utilidades son:

- ***Cpio***: ***Copy in/Copy out***, crea un archivo, de uno o varios archivos, tomando una lista de nombres que se ingresan por teclado. Provee formateo y chequeo de error y permite guardar archivos en múltiples cintas
- ***ufsdump*** o ***dump***: Permiten programar copias de respaldo a máximo 9 niveles (día, semana, quincena, mes, año, ...).
- ***tar***: Permite hacer copias de múltiples archivos en un directorio. Es muy fácil de usar.

3.3.2 Sumas de chequeo: Chequear el tamaño de un archivo no asegura que el archivo no haya sido modificado, por lo tanto existen otras técnicas y algoritmos para saber si el contenido de un archivo ha sido modificado.

Existen programas que sirven para chequear la integridad de los archivos, los cuales revisan el tamaño de un archivo y pueden ser corridos periódicamente para archivos importantes.

El comando ***sum*** calcula suma de chequeos a los archivos. Cuando se instala el sistema es posible calcular sumas de chequeo a archivos que comúnmente no cambian y recalcularlas periódicamente para comparar que los archivos no han sido modificados.

Sin embargo existen dos problemas que se pueden presentar:

- Es posible que un intruso cambie un archivo de tal forma que la suma de chequeo permanezca.
- Es posible que las sumas de chequeo almacenadas para verificación sean modificadas sin el conocimiento del administrador. Para este caso se aconseja almacenarlas en medios diferentes.

Otra forma es calcular números basados en mensajes de verificación o firmas digitales. El programa ***tripwire*** calcula mensajes de verificación para archivos críticos y genera una base de datos basada en firmas digitales dado un conjunto de archivos y directorios. Este esquema es menos vulnerable que las sumas de chequeo debido a que es menos probable generar estos mensajes por un intruso.

3.3.3 Encripción de archivos: En **UNIX *Solaris*** existen dos programas para llevarlo a cabo: el **crypt** y el **DES**. El **crypt** sólo puede prevenirnos de que una persona casual vea el contenido de nuestro archivo, pero no puede protegerlo de un **cracker**. Se conocen varios métodos para desencriptar archivos, bastante largos, en pocas horas sin saber qué es lo que contienen. El ***DES***, sin embargo, es mucho más seguro, de todas formas nunca se ha probado que lo sea totalmente, y en los últimos años han surgido serias dudas sobre su infalibilidad.

4. SEGURIDAD DE SERVICIOS INTERNET EN UNIX

Cuando los computadores se conectan por medio de una red a Internet, ellos llegan a ser mucho más vulnerables de ataques desde el exterior, debido al libre acceso y compartimiento de recursos. El sistema **UNIX** generalmente viene sin las configuraciones adecuadas de los servicios de ***TCP/IP*** para trabajar en red, esas capacidades deben ser configuradas cuidadosamente para que el sistema sea lo más seguro posible.

El presente capítulo abarca los aspectos principales a tener en cuenta para obtener una configuración adecuada cuando un equipo es conectado a Internet, específicamente bajo ambiente **UNIX**, utilizando el protocolo ***TCP/IP***, incluyendo formas de protegerlos contra acceso sin autorización, y aplicaciones distribuidas. Se estudia además los servicios Internet con sus vulnerabilidades y en alguno de ellos, sus posibles soluciones.

4.1 HOST DE CONFIANZA Y USUARIOS

El administrador de una red local puede asignar uno o más ***host*** remotos como "confiables" listándolos en el archivo ***/etc/hosts.equiv***. Un usuario puede conectarse y accesar los recursos de un ***host*** en la red sin requerir un ***password***.

De esa misma manera un administrador puede asignar ***host*** a los usuarios en el archivo ***/etc/.rhosts***.

4.1.1 Archivo /etc/hosts.equiv: Este archivo contiene una lista de ***host*** confiables, uno por línea. Si un usuario intenta conectarse remotamente usando ***rlogin*** o ***rsh*** en uno de los ***hosts*** listados en el archivo /***etc/host.equiv*** y si ese usuario tiene una cuenta con el mismo ***login***, el sistema permite al usuario conectarse sin necesidad de un ***password***.

Cuando sólo se encuentra el nombre del ***host*** en el archivo ***/etc/hosts.equiv*** significa que el ***host*** es confiable y así también lo es cualquier usuario de esa máquina.

Si el nombre del usuario también se encuentra al lado del nombre del ***host*** significa que sólo él es confiable en ese ***host***.

Si al nombre de un grupo lo procede el signo **+** significa que todas las máquinas en ese grupo son confiables.

Si al nombre de un grupo lo procede el signo **-** significa que ninguna de las máquinas son consideradas confiables.

4.1.2 Archivo /etc/.rhosts: El archivo ***/etc/.rhosts*** para el usuario es equivalente al ***/etc/hosts.equiv*** para las máquinas. Este contiene una lista de todas las combinaciones de ***hosts – usuarios***. Si una combinación ***host-usuario*** es listada en este archivo, ese usuario obtiene permiso para conectarse remotamente a ese ***host*** sin suministrar un ***password***.

Los usuarios pueden crear el archivo ***/etc/.rhosts*** en su directorio ***home***. Usando este archivo es otra forma para permitir acceso confiable entre sus propias cuentas sobre diferentes sistemas sin usar el archivo ***/etc//host.equiv***.

4.2 SERVICIOS DE RED

Hay diferentes servicios en **UNIX** que permiten a los usuarios conectarse de un ***host*** a otro y ejecutar comandos en ***host*** remotos (accesos remotos). Esos servicios están basados en el modelo ***Cliente/Servidor***, y se ejecutan en diferentes puertos lógicos del sistema. Cierto números de puertos generalmente son reservados para algunos servicios de red. En **UNIX** solamente el superusuario puede establecer conexiones a través de los puertos numerados del 0 al 1.023, los cuales son conocidos como “puertos confiables”, con el objeto de que un usuario cualquiera no pueda obtener información privilegiada a través de un servicio falso. Debido a que esto no es un estándar de ***TCP/IP*** sino una convención de **UNIX**, es posible que un PC con una tarjeta de red genere conexiones falsas desde puertos

confiables. Los puertos de protocolo disponibles no son conocidos globalmente, sino, cuando un programa necesita un puerto, el software de la red le asigna uno dinámicamente.

Cada servicio de red tiene su propia configuración de archivos el cual debe ser cuidadosamente configurada para proteger el sistema. El archivo ***/etc/services*** muestra todos los servicios de red y los puertos en que se ejecuta y qué protocolo de red ellos están utilizando. Por ejemplo:

```
ftp_21/tcp
smtp_25/tcp_mail
```

Esto indica que el servicio ***ftp*** está ligado con el puerto de protocolo 21 y que se trata de un servicio ***TCP*** y que ***smtp*** está en el puerto 25.

4.2.1 Inetd: El sistema **UNIX** cuenta con programas separados (conocidos como ***daemons***) para proporcionar cada uno de los servicios, como ***telnet***, ***ftp***, ***finger***, entre otros. Sin embargo, con el paso del tiempo, el número de servicios de red ha crecido considerablemente. Por lo tanto, el tener corriendo siempre los ***daemons*** para cada uno de los servicios, en espera de conexiones a dicho servicio, se consideró impráctico, y se decidió utilizar un sistema de "intermediario". En este esquema, utilizado ahora de manera universal en **UNIX**, existe un ***daemon*** llamado ***inetd*** (***Internet services daemon***) que es quien está ejecutándose de

manera constante, en espera de peticiones de servicios. Cuando se recibe una petición, ***inetd*** ejecuta el ***daemon*** correspondiente al servicio solicitado, lo "conecta" al puerto apropiado, y reanuda su tarea de escucha. De esta manera, solamente se ejecutan los servidores que sean necesarios de acuerdo con las solicitudes recibidas. La configuración de ***inetd***, se hace a través del archivo ***/etc/inetd.conf***, el cual tiene el siguiente formato:

servicio socket protocolo wait/nowait usuario programa args

A continuación se específica cada uno de los parámetros del formato anterior:

- ***servicio*****:** Nombre del servicio establecido en esa línea.
- ***socket*****:** Tipo de ***socket*** que utiliza el servicio. Puede ser ***stream*** o ***datagram***.
- ***protocolo*****:** Es el protocolo que utiliza el servicio. Puede ser ***tcp*** o ***udp***.
- ***wait/nowait*****:** ***wait*** Indica que después de realizar un servicio, el servidor queda corriendo por un período de tiempo, en espera de otras conexiones. ***nowait*** indica que el servidor muere en el instante que termina de prestar el servicio.
- ***usuario*****:** Indica el usuario con cuyos privilegios se ejecutará el servidor.
- ***programa args*****:** Es el nombre del programa a ejecutar para prestar el servicio, acompañado de los argumento que el programa requiera.

4.2.2 Http: El protocolo ***Hyper Text Transfer Protocol*** (***http***) se encarga de la transferencia de documentos ***HTML*** (***Hyper Text Markup Language***). En general hay cuatro tipos básicos de riesgo:

1. Los documentos privados almacenados en el sitio ***web*** pueden caer en manos de usuarios no autorizados.

2. La información enviada por el usuario remoto al servidor ***web*** (tal como números de cuenta y tarjetas de crédito), puede ser interceptada por terceros

3. Algunos datos acerca del computador que actúan como servidor ***web*** pueden ser extraídos, dándole la posibilidad al atacante de acceder a información que potencialmente le permita introducirse en el ***host***.

4. La ejecución de comandos en el ***host*** por parte de los usuarios, que les permitan modificar y/o dañar el sistema.

Otro punto vulnerable en los servidores ***www*** está relacionado directamente con el uso de los ***scripts CGI*** (***Common Gateway Interface***). En general un servidor ***WWW*** en el que los usuarios puedan utilizar ***scripts CGI***, puede tener agujeros grandes de seguridad de dos formas distintas:

1. Los ***scripts***, que están diseñados para atender y responder ciertas solicitudes de los usuarios, pueden eventualmente suministrar información delicada relacionada al ***host*** que puede ser útil para los ***hackers***.

2. Los ***scripts*** que procesan entradas del usuario, tales como datos que el usuario llena en un formato ***html***, pueden ser vulnerables a ataques en los que el usuario remoto intenta engañarlos disfrazando comandos en la información suministrada.

Un ***script CGI*** corrupto, o modificado por un atacante, puede permanecer suficiente tiempo con un nivel alto de privilegios, antes de que el administrador se entere y entre tanto, entregar datos como ***passwords***, mapas de la red, o permitir la iniciación de sesiones remotas a través de un puerto dado.

4.2.3 Telnet: El servicio ***telnet*** permite a los usuarios conectarse a un sistema remoto y trabajar en el sistema como si estuviera conectado físicamente a él. Para accesar al sistema los usuarios deben ingresar su ***login*** y ***password***, este es un servicio muy usado porque permite a los usuarios accesar directamente diferentes sistemas desde cualquier lugar.

El protocolo ***telnet***, por el simple hecho de permitir a un usuario obtener acceso a otras máquinas, es fuente de problemas de seguridad. Los dos principales problemas que existen son:

- Normalmente, para establecer una sesión, el usuario tiene que proporcionar un ***password***. Si en su viaje a través de la red este ***password*** es capturado, puede ser reutilizado para establecer sesiones posteriores a la misma cuenta.
- Ha sido descubierta una vulnerabilidad en muchos ***daemons*** de ***telnet***, que puede incluso permitir a cualquier persona obtener privilegios de ***root*** en la máquina a la que se conecta.

  Este problema está documentado en el ***CERT Advisory 95:14*** (disponible en ***ftp://ftp.super.unam.mx/pub/security/doc/cert_advisories/CA*** - ***95:14.Telnetd_Environment_Vulnerability***), y es muy recomendable leer dicho documento y corregir el problema, en caso de que exista.

4.2.4 rlogin y rsh: El servicio ***rlogin*** es similar a ***telnet*** en que provee conexiones de acceso remoto siendo estas menos seguras. ***rsh*** permite ejecutar comandos de un sistema remoto sin actualmente estar conectado al sistema, ambos servicios pueden ser configurados y no requerir un ***login*** para comenzar una nueva sesión.

Para el uso de estos comandos es estrictamente necesario que estén configuradas adecuadamente las equivalencias de usuario y de ***host***. La mayoría de los usuarios tienen dificultades con este tipo de comando, cuando el administrador de red no ha configurado adecuadamente estas equivalencias.

Un sistema puede tener configurados los servidores autorizados con permisos de confianza en los archivos ***/etc/hosts.equiv*** y ***/etc/.rhosts***. A través de ***rlogin/rsh***,

un usuario sobre alguno de esos servidores puede entrar al sistema sin necesidad de dar información de acceso. Es necesario monitorear estos archivos de configuración para garantizar un uso adecuado de los mismos.

4.2.5 Ftp (***protocolo de transferencia de archivo***): Permite enviar y recibir archivos desde y hacia un ***host*** remoto a través de la red; usualmente un ***login*** y un ***password*** son requeridos para establecer una conexión, pero hay excepciones.

Este es el servicio más utilizado para la distribución de archivos de todo tipo en Internet. Los problemas que puede ocasionar un servidor de ***ftp*** con errores o mal configurado son:

- Posibilidad de modificar los archivos existentes en el área de acceso anónimo, con esto se abren posibilidades de acceso interactivo al sistema, así como de reemplazar los archivos ya existentes por copias modificadas
- Distribución de programas o material ilegal (software pirata).
- Ataques de "negación de acceso" al consumir todos los recursos disponibles.
- Posibilidad de ejecutar comandos arbitrarios en el servidor.

4.2.6 Tftp: Significa ***Trivial File Transfer Protocol***, es un protocolo muy sencillo de transferencia de archivos, que no proporciona ninguna característica de seguridad. Utilizando el comando ***tftp***, es posible conectarse a cualquier máquina

que proporcione el servicio, sin necesidad de proporcionar ningún ***password***, y transferir archivos.

Tftp fue diseñado inicialmente para permitirle a las terminales cargar de forma automática su sistema operativo a través de la red, utilizando este protocolo para obtener los archivos apropiados de otra máquina. Como ***tftp*** no proporciona ningún control de acceso, se supone que el ***demonio*** de este servicio (llamado ***tftpd***) debe restringirse para que transfiera solamente ciertos archivos. Muchos ***demonios*** de ***tftp*** en versiones comerciales de **UNIX** no proporcionan ese tipo de control y le permiten a quien se conecte acceder a cualquier archivo del sistema, incluyendo ***/etc/passwd***.

4.2.7 Finger: El servicio ***finger***, que permite obtener información acerca de usuarios, tanto en la máquina local como en máquinas remotas, a través de la red, ofrece peligros tanto para el servidor como para el cliente:

- Para el servidor, puede divulgar información sobre la máquina y los usuarios, que puede ser útil en ataques posteriores. Para el cliente, si el ***demonio*** de ***finger*** del servidor es defectuoso o ha sido modificado maliciosamente, puede hacerle llegar información peligrosa. Por ejemplo, si el servidor genera una salida infinita de texto, el disco local se puede llenar, causando posiblemente una caída del cliente.

- Es posible desactivar completamente el servicio de ***finger*** comentando la línea apropiada en ***/etc/inetd.conf***. Sin embargo, utilizado correctamente, ***finger*** puede ser un servicio muy útil.

4.2.8 Sendmail: Definitivamente, el correo electrónico es el medio de comunicación por excelencia en Internet. Permite enviar mensajes a cualquier lugar de la red, incluyendo sitios donde se manejen otros protocolos de red, otros tipos de máquinas y otros sistemas operativos, diferentes de los de la máquina de origen.

Sin embargo, esta flexibilidad ha llevado a que ***sendmail***, el ***demonio*** que recibe y envía correo electrónico en prácticamente todas las versiones de **UNIX**, sea un programa altamente complejo, con muchísimas opciones de configuración y que, de manera individual, ha sido posiblemente el programa que más problemas de seguridad ha causado a lo largo de la historia de **UNIX**.

En sus primeras versiones, ***sendmail*** tenía huecos tan grandes como para permitir a cualquier persona ejecutar comandos con privilegios de ***root*** de forma remota. Con el paso del tiempo, la gran mayoría de ellos han sido corregidos, pero incluso actualmente, casi todos los años se encuentra (y se corrige) un agujero importante de seguridad en ***sendmail.***

4.2.9 X-windows: Es un sistema gráfico, que permite a los usuarios ejecutar aplicaciones a través de la red de forma casi completamente transparente. Es la base de prácticamente todas las interfaces gráficas existentes actualmente en estaciones de trabajo. Sin embargo, también tiene problemas de seguridad y algunos bastante graves.

En su nivel más bajo, ***X-windows*** es en realidad un protocolo de comunicación, llamado ***protocolo X***, que utiliza un modelo ***cliente/servidor*** para funcionar. Aquí, la máquina en la que trabaja el usuario, es conocida como servidor ***X-windows***. Se le llama así debido a que es la que genera las entradas y maneja las salidas de los clientes. Los clientes ***X-windows*** son aplicaciones que reciben y procesan las entradas asignadas en el servidor y producen salidas dirigidas a él.

X-windows es lo suficientemente flexible como para permitir que los clientes y el servidor estén corriendo en máquinas distintas; en un solo servidor pueden estar desplegando información clientes que están corriendo en muchas máquinas diferentes. Aquí es donde está el problema de seguridad. Si un cliente puede interceptar entradas generadas en el servidor que no le corresponden o enviar un mensaje a otro cliente, puede haber un hueco de seguridad. Los problemas más peligrosos que pueden ocurrir con ***X-Window*** son:

- Modificación de parámetros de la sesión de trabajo del usuario.
- Creación y destrucción de ventanas.

- Captura de eventos ***X-windows*** , como movimientos del ***mouse***.
- Creación de eventos ***X-windows***

Además, El comando ***xhost*** permite diseñar servidores con atributos de confianza, desde los cuales los servidores ***X-windows*** pueden ser accesados. Muchos usuarios con la opción *"**xhost +**",* deshabilitan el control de acceso y permiten que clientes ***X-windows*** se conecten a su sistema. Esto hace fácil y conveniente el acceso remoto mediante una interfaz gráfica, pero de igual manera es posible que los intrusos obtengan acceso por este medio.

4.2.10 NFS: El ***NFS*** se diseñó para permitir a los usuarios compartir los archivos a través de la red. Uno de los usos más comunes de ***NFS*** es para las estaciones sin disco. Bajo el sistema ***NFS*** un servidor mantiene datos y recursos para muchos clientes. El cliente tiene acceso al sistema de archivos que el servidor exporta a los clientes; los usuarios conectados a la máquina cliente pueden accesar al sistema de archivos montándolos desde su servidor. Para el usuario en la máquina cliente estos archivos parecen como si estuvieran en la máquina local.

Los puntos de montaje son directorios del cliente y se utilizan para unir (o montar) otros sistemas de archivos. Cuando esto sucede, cualquier archivo o directorio que pudiera estar almacenado localmente en el directorio que actúa como punto de montaje es ocultado durante el tiempo en el que el sistema de archivos está montado. Estos archivos no están permanentemente afectados por el proceso de montaje y son de nuevo accesibles cuando se produce el desmonte. Sin embargo,

los directorios de montaje suelen estar vacíos para no oscurecer la comprensión del sistema de archivos.

Cada sistema tiene una tabla virtual que especifica qué sistemas son montados por defecto. En este archivo se especifican el sistema de archivos local de **UNIX** y los ***NFS*** que son montados automáticamente cuando un sistema arranca.

A través del comando ***mount*** se obtiene acceso a un sistema de archivos remoto como si estuviese montado físicamente en su sistema. Si un usuario remoto tiene acceso a éste, podría quedar habilitado para borrar o modificar archivos. Para montar un sistema de archivos remotos, primero debe ser exportado usando el comando ***share***. Este comando actualiza el archivo ***/etc/exports*** listando los servidores y usuarios con privilegios para montar cada sistema de archivos exportado. Si se listan los nombres de los servidores sin usuarios ni contraseñas, quiere decir que todos los usuarios tienen todas las autorizaciones. Se debe estar seguro que la información sobre estos archivos debe ser explícita, referenciando a los usuarios y servidores con atributos de confianza.

4.2.11 NIS ***(Network Information System)***: Fue desarrollado por ***Sun Microsystems*** para reducir el esfuerzo necesario para la puesta a punto y mantenimiento de las estaciones en **UNIX**. Se basa en la centralización de los archivos necesarios para la configuración de nuestra red, en una única máquina que sería el servidor ***NIS***. Sólo será necesario actualizar los archivos de configuración en esta máquina para que los cambios tengan efecto en todas las que forman parte del dominio.

A grandes rasgos el ***NIS*** se basa en el protocolo ***RPC*** (***Remote Procedure Calls***). Mediante ***RPC*** se consigue que un proceso, pueda llevar a cabo llamadas a funciones en máquinas remotas. El propósito de ***NIS*** es el establecimiento de una base de datos distribuida. El servidor ***NIS*** se construye de tal forma que pueda leer la base de datos pero no pueda actualizarla. .

NIS se considera seguro puesto que no modifica los mapas directamente y los archivos pueden ser leídos por todo el mundo. Esto desde el punto de vista del servidor; pero cambiando hacia el cliente, la seguridad es bastante pobre. Puesto que ***NIS*** no utiliza autenticación a nivel de ***RPC***, cualquier máquina de la red puede enviar una respuesta falsa, y pasar por el servidor de ***NIS***. Para que el ataque tenga éxito, el paquete con la respuesta falsa debe llegar al cliente antes de que responda el servidor. También debe ser capaz de observar la petición y generar la respuesta. Para esto es preciso que la máquina que lleva a cabo el ataque esté situada en el mismo camino de comunicación del servidor y del cliente.

Un intruso podrá acceder al cliente ***NIS*** a través de los comandos de acceso remoto, cambiando en el mapa ***hosts.byaddr*** la dirección de la máquina que lleva a cabo la intrusión a otra máquina que se encuentre el archivo /**etc**/**hosts.equiv** o en uno de los archivos ***/etc/*.rhosts** de los usuarios.

4.2.12 DNS: El servicio ***DNS*** (***Domain Name System***), es un amplio conjunto de bases de datos distribuida usado a lo largo de la Internet, proporcionando correspondencia entre los nombres de ***host*** y las direcciones de ***IP*** de los mismos. Existe la posibilidad de abusar de este servicio para poder entrar en un sistema. Suposiciones durante la fase de autenticación, pueden conllevar serios huecos de seguridad importantes. El problema de seguridad es similar al que existe en el ***NIS***. La autenticación se lleva a cabo en muchos casos a partir del nombre o la dirección, las aplicaciones de alto nivel, usan en la mayoría de los casos los nombres para la autentificación, puesto que las tablas de direcciones son mucho más difíciles de crear, entender y mantener. Si por ejemplo alguien quiere suplantar una máquina por otra, no tiene más que cambiar una de las entradas de la tabla que relaciona su nombre con su dirección. Este es el problema fundamental de ***DNS***. Para conseguir esto una máquina debe obtener primero el número ***ID*** de la petición ***DNS***. Para ello debe construir el paquete de respuesta y usar la opción de enrutamiento de fuente, para hacerlo llegar al que llevó a cabo la petición.

5 HERRAMIENTAS DE SEGURIDAD

Comúnmente un administrador de un sitio Internet ignora el grado de vulnerabilidad de su sitio. En la mayoría de los casos adolecen de los conocimientos de las plataformas y los servicios que manejan. En otros casos desconocen que existen técnicas de evaluación de seguridad provistas por el mismo sistema operativo que ayudan a mejorar la seguridad, tales como auditorias, técnicas estadísticas y otras. Algunos administradores conocen de la existencia de esas técnicas de seguridad pero no saben aplicarlas, por muy sencillas que ellas sean.

No es suficiente con tener algún conocimiento de estas técnicas si se espera alcanzar un grado elevado de seguridad. Para tal efecto, existen herramientas sofisticadas que se pueden clasificar en dos categorías: Scanners y barreras. Los scanners rastrean y encuentran huecos de seguridad, mientras que las barreras son usadas como filtros con objeto de contrarrestar los ataques.

Es importante rescatar la potencialidad de este tipo de herramientas, ya que mediante su uso adecuado es factible encontrar diversos tipos de vulnerabilidades en los servicios Internet, razón por la cual, es conveniente contar con un procedimiento de evaluación.

5.1 SATAN

Satan es la herramienta de análisis de seguridad para auditoría de redes ***(Security Analysis Tool for Auditing Networws)*** desarrollada por ***Dan Farmer*** y ***Wietse Venema***, que reúne gran cantidad de información sobre servidores remotos, examinando los servicios de red, tales como: ***finger, nfs, nis, ftp***, etc. ***Satan*** es capaz de identificar las siguientes vulnerabilidades de estos servicios:

- Configuraciones incorrectas.
- Configuración inadecuada de los servicios de red.
- Problemas en las utilidades de red o del sistema.
- Decisiones pobres o ignorantes en políticas de seguridad.

Satan puede hacer reportes de datos con presentación para visores ***HTML***, tales como Netscape o Explorer. Además está diseñando específicamente para buscar inseguridades en una red, pero también puede ser utilizado para adquirir información general de la configuración de la red como: Topología, servicios montados, tipo de hardware y software utilizado.

Sin embargo el verdadero poder de esta herramienta viene a ser utilizado cuando se usa en modo exploratorio, ya que es capaz de examinar las rutas de permisos y dependencias entre servidores secundarios basándose en una colección inicial de información y en unas reglas de configuración. Esto no solamente le da al

administrador la información de su red o de su servidor, sino que le da la información del estado de seguridad del sistema incluyendo el entorno y las relaciones (rutas y permisos) entre todos, ayudándole a tomar las decisiones acerca del nivel de seguridad requerido.

Satan tiene un programa de adquisición de blancos que usa ***fping*** para determinar si los servidores de una red están vivos. Luego esta información pasa a un motor que maneja la recolección de datos en el ciclo principal.

Uno de los principales conceptos que maneja este ***scanner*** es el de los permisos en el ambiente de red, los cuales son bastante delicados. Se habla de permisos de confianza ***(Trust)*** cuando un servidor puede tener un servicio local comprometido por un cliente con o sin la autorización adecuada y a su vez el servidor tiene permisos en otra maquina. Esto hace que el usuario pueda acceder al último servidor con los permisos que tiene el primero.

Correr ***Satan*** significa probar la seguridad de un sitio remoto. Este tiene la capacidad de escanear un gran número de servidores dentro de una red; afortunada o desafortunadamente, usted no puede tener la autoridad o permiso para escanear todos los servidores. ***Satan*** nunca debería ser usado para escanear servidores cuando no se tiene permiso explícito del dueño del servidor.

Satan se usa editando el archivo de configuración (***config/Satan.cf***) antes de correrlo. Es fundamental tener en cuenta que para correrlo hay que tener privilegios de superusuario.

Se necesita perl5 así como un compilador de C para lograr que ***Satan*** corra apropiadamente. Esta herramienta crea y usa realmente pocos archivos, pero un usuario típico solo necesita estar relacionado con el archivo de configuración, (***config/Satan.cf***).

Desafortunadamente ***Satan*** no es muy portable como se quisiera, pero aun así puede correr en un numero bastante grande de maquinas **UNIX** y puede trabajar en varios sistemas operativos.

Satan fue construido específicamente para los siguientes propósitos:

- Descubrir si el mapeo de seguridad en grandes redes tenia solución.
- Utilizar el kit de herramientas tradicional de **UNIX** como una aproximación al diseño
- Utilizar en lo posible las herramientas de software disponibles, con el objetivo de reducir el tiempo de diseño a un mínimo.
- Diseñar un paquete de software que fuera a la vez educativo y útil.
- Crear una herramienta disponible libremente para cualquiera que quisiera usarla.

- Descubrir gran cantidad de información de seguridad de la red como fuese posible sin ser destructivo.
- Crear la mejor herramienta disponible de seguridad en redes a cualquier precio.
- Promover la creación de programas comerciales o académicos en el área de seguridad.
- Por ultimo mostrar que tan inseguro es realmente el Internet y que tanto la seguridad de un sitio depende de un gran numero de ***host*** potencialmente inseguros.

Satan puede hacer un barrido a un conjunto de servidores, lo que le da una imagen más clara de lo que es la seguridad de la red en una forma global, examinando todas las redes de confianza y las posibles avenidas de ataque o de aproximación. Puesto que no hay forma de que ***Satan*** pudiese a priori saber donde va a estar barriendo, se decidió que en lugar de colocar unos frenos artificiales en el programa, se permitiera que el administrador del sistema colocara sus propios frenos de donde quiere que corra, a través del archivo de configuración.

Puede ser muy fácil usar ***Satan*** la primera vez; sus capacidades básicas son muy simples para usar y entender. Para arrancar a usarlo, se siguen los siguientes pasos:

1. Realizar una copia del programa
2. Barrer su servidor o red.
3. Analizar la salida.

En general, ***Satan*** es un ***kernel*** genérico relativamente pequeño que sabe poco o nada acerca de los tipos de sistemas, nombre de servicios de red, vulnerabilidades u otros detalles. El conocimiento acerca de estos detalles se construye en una pequeña y dedicada herramienta colectora de datos y reglas básicas. Como mencionamos anteriormente, el comportamiento de ***Satan*** se controla desde el archivo de configuración (***config/Satan.cf***).

La parte más complicada de este ***scanner*** es la interpretación de los resultados obtenidos, debido a que no existe un nivel de seguridad correcto. Una buena seguridad depende mucho de las políticas y precauciones que se tengan del sistema o del sitio involucrado.

Además, existen algunos conceptos utilizados por ***Satan***, tales como, por qué la información de confianza y de red son dañinas, y muchas de las opciones que puede ser utilizadas (proximidad, filtros de ataques, entre otras), que no son familiares o que desconocen muchos administradores de sistemas, es muy importante leer y entender la documentación para utilizar la herramienta adecuadamente.

En los reportes generados por ***Satan***, si hay un servidor listado con un punto rojo esto significa que el servidor tiene una vulnerabilidad que podría comprometerse, un punto negro significa que no hay vulnerabilidades que hayan sido encontradas para ese servidor en particular todavía; el hacer un clic en los hipertextos da mayor información sobre ese servidor o la red, piezas de información o vulnerabilidades.

La mejor manera de comprender lo que ***Satan*** hace, es haciendo el escaneo de las redes y examinando los resultados con las herramientas de reportes y análisis.

Satan puede encontrar problemas de seguridad en estas áreas :

- ***File system*** exportados inadecuadamente
- Acceso desde un hosts arbitrario al archivo de ***password NIS***
- Versión vieja de oldmail
- Control de acceso desabilitado en X-server
- Archivo accesible vía ***tftp***
- ***ftp*** anónimo configurado inadecuadamente.

5.2 ISS

En 1992, ***Christopher Klaus*** desarrolló el ***ISS (Internet Security Scanner)***, creado para ayudar a los administradores a explorar la vulnerabilidad de la seguridad de sus respectivas redes asociadas por los servidores de ***TCP/IP***.

Fue la primera herramienta de seguridad conocida en el mercado a nivel mundial. Pasado tres años el ***ISS***, había sido usado en más de 1500 corporaciones, logrando su objetivo, ayudar a detectar vulnerabilidades en redes de computación.

El ***Internet Scanner*** tiene unos reportes predefinidos que ayudan a recopilar la información necesaria para tomar las decisiones acerca de las políticas de suguridad.

El ***Internet Scanner*** ejecuta una amplia gama de detección de vulnerabilidades en la industria. Encuentra tanto vulnerabilidades como intrusos mediante la examinación de dispositivos, servicios y sus interrelaciones. Informa detalladamente de cada vulnerabilidad encontrada, el ***host*** vulnerable, descripción de la vulnerabilidad y acciones correctivas.

Internet Scanner puede usarse sobre todas las redes basadas en ***TCP/IP***, las redes conectadas a Internet, así como máquinas de redes ***Standalone***.
ISS puede simular más de 120 formas de ataques de vulnerabilidad de la seguridad de la red. Estos ataques incluyen rutinas de diagnóstico sobre cada uno de los siguientes servicios:

- *Sendmail*
- *ftp*
- *NNTP*
- *Telnet*
- *RPC*
- *NFS*

Se pueden ejecutar escaneos desde la interface gráfica o desde la línea de comandos. Además de esto, los resultados se pueden analizar, creando y examinando una variedad de reportes generados por ***Internet Scanner*** en sus aplicaciones personalizadas. Se puede usar la interface gráfica y sus herramientas de visualización de datos para analizar escanneos ejecutados previamente.

La siguiente lista resume los pasos que hay que realizar para la evaluación de su red y el desarrollo de las políticas de seguridad:

- Recopilar información de la red
- Probar la red
- Planear una estrategia de scanneo
- Implementar el plan
- Operar la estrategia de scanneo

Para poder correr el ***ISS***, hay que conectarse como superusuario y además se necesita un visor estándar ***WEB*** para ver los reportes y archivos de ayuda ***HTML***. Es necesario tener el archivo de clave (licencia) que es suministrado por la compañía ***ISS.*** Este archivo habilita las características del ***Internet Scanner***. Sin este, el ***Internet Scanner*** corre como una copia de evaluación, la cual provee únicamente pruebas ***Intranet*** para la computadora local.

Existen políticas que son usadas para sintonizar los escaneos. Ellas determinan cuales vulnerabilidades correrá el ***Internet Scanner*** durante una sesión del escaneo.

Antes de correr un ***scan***, se debe sintonizar la configuración de las políticas de ***Internet Scanner***. Para especificar el número de configuraciones aplicables al ambiente de red y acortar la duración del ***scan***, hay que deshabilitar las configuraciones que no son requeridas.

Hay cuatro políticas por defecto las cuales hacen parte del ***Internet Scanner***:

- **Escaneo suave:** Evalúa rápidamente los servicios y las vulnerabilidades **UNIX** encontradas sobre una red; se debe correr este escaner inicialmente.

- **Escaneo medio:** Además de los chequeos del escaneo suave, prueba los demonios más comunes, configuraciones ***RPC***, ***SMTP*** y ***FTP***.

- **Escaneo pesado:** Además de los chequeos del escaneo mediio, prueba todas las otras vulnerabilidades, con la excepción de algunos chequeos de rechazo de servicios.

- **Identificación OS:** Usa los cheques que se refieren a información de servidores escaneados. Reporta al sistema operativo y versiones del sistemas, también como algunos programas actualizados.

Los métodos de fuerza bruta tratan de encontrar cuentas por defecto abiertas usando pares comunes de nombres de usuarios y contraseña. *Internet Scanner* construye una lista de pares de nombres de usuarios-contraseña para examinar la lista de usuarios por defecto. Al menos una lista de usuarios por defecto debe ser seleccionada desde las siguientes opciones:

Miscellaneous Default Users (Usuarios diversos por defecto)
VMS Default Users (Usuarios por defecto *VMS*)
UNIX Default Users (Usuarios por defecto *UNIX*)

Además de la configuración de estas opciones, usted puede agregar adicionalmente pares de contraseñas usando el archivo de nombre de usuarios por defecto (***username***), .

Si ***Internet Scanner*** encuentra una cuenta por defecto abierta, es importante corregir esta vulnerabilidad. Un ejemplo de una cuenta por defecto es ***sync***, un nombre común de cuenta en ***SunOS*** y otras versiones de **UNIX**. Típicamente,

este acceso revela solamente información adicional acerca del tipo y el número de versión de los sistemas ***OS*** y desplegará el mensaje del día (***MOTD***). Esto normalmente no es un riesgo severo. Sin embargo, ciertos paquetes ***FTP*** pueden permitir que un atacante se conecte como ***sync*** y robe el archivo de contraseñas.

Otros paquetes de software no pueden ejecutar un chequeo apropiado para acceso de cuentas, abriendo las vulnerabilidades de seguridad con cuentas ***nonpassword*** protegidas tales como ***sync***. La cuenta ***sync*** sin contraseña puede llegar a ser una vulnerabilidad de la seguridad del sistema: Un intruso con una cuenta regular sobre el servidor puede desviar la cuenta ***sync*** (sobre sistemas con librerías ***linked*** (enlazadas) al tiempo de corrida). Ellos cambian la *variable* ***LD LIBRARY PATH*** a una librería de su propio resorte, y corren ***login*** o ***su*** para ganar privilegios de otros usuarios hasta finalmente alcanzar o llegar a ser el superusuario.

Todas las cuentas, incluyendo la cuenta ***sync***, no deberían tener contraseñas fáciles. Si el acceso de la conexión de estas cuentas no es necesaria, estas debieran ser removidas o marcadas con un * (asterisco) en el archivo de contraseñas, y la cadena ***/bin/false*** en el campo ***shell*** en ***/etc/passwd***.

Sobre algunos sistemas **UNIX**, si no hay muchas conexiones en un período de tiempo, ***inetd*** apaga un servicio (tal como ***Telnet***) para un período de tiempo. Se puede modificar para que ***inetd*** permita más conexiones por un período de tiempo. ***Internet Scanner*** puede seleccionar cuántas conexiones simultáneas

pueden suceder entre un período dado para disminuir lentamente la posibilidad de ataques de fuerza bruta a un nivel aceptable para ***inetd***.

El ***Internet Scanner*** ofrece muchas opciones para generar sus reportes. El reporte puede generarse en varios formatos, incluyendo formatos que contienen información fija e información mejorada, para reportar las vulnerabilidades. Para proveer un conjunto más útil de reportes que les podrían servir a múltiples usuarios, los reportes detallados se han categorizado como sigue:

- **Reporte Ejecutivo**: Proveer múltiples tendencias, porcentajes, y un número paquetes agregados fácilmente digeribles en gráficas a color para resúmenes administrativos.

- **Reportes de Administración de Línea**: Provee datos de nivel intermedio relativos a vulnerabilidades descubiertas, servicios activos, ***host*** interrogados y de sistemas operativos descubiertos.

- **Reportes Técnicos**: Provee datos de seguridad detallados y comprensibles, relativos a la vulnerabilidad de ***host***, servicios y software presentes para personal técnico que requiere evaluar y/o corregir los datos de seguridad no cubiertos.

5.3 TCP/WRAPPERS

TCP Wrappers es una herramienta simple que nos sirve para monitorear y controlar el tráfico que llega por la red. Esta herramienta ha sido utilizada exitosamente en la protección de sistemas y la detección de actividades ilícitas. Desarrollada por ***Wietse Venema*** de la ***Universidad de Eindhoven***, en ***Holanda***, además de monitorear y controlar, filtra requerimientos de acceso a los servicios de red ofrecidos por **UNIX**, tales como ***systat, ftp, telnet, rsh, exec, tftp, talk*** y otros servicios.

Un ***Wrapper*** es un programa para controlar el acceso a un segundo programa, literalmente cubre la identidad del segundo programa, obteniendo con esto un nivel más alto de seguridad. Estos programas nacieron de la necesidad de modificar el comportamiento del sistema operativo sin tener que modificar su estructura. Un ejemplo sencillo del uso de los ***Wrappers*** es limitar la cantidad y tipo de información que manejen programas a través de la red, como es el caso de ***sendmail*** y ***finger***, entre otros.

El funcionamiento de ***TCP Wrappers*** es muy sencillo; se trata de un programa ***Wrappers*** o ***daemon tcpd*** que puede ser instalado sin necesidad de hacer cambios mayores al software, ni a la configuración. Una vez instalado, se configura el ***inetd***, para que en vez de ejecutar el ***daemon*** del servicio solicitado ejecute al ***tcpd***.

El programa ***Wrappers*** cuenta con un simple pero poderoso mecanismo, en el cual el ***inetd*** es engañado; ya que en lugar de correr el programa servidor deseado, se corre un pequeño programa o demonio ***Wrapper <tcpd>.*** Cuando este demonio es ejecutado a través del ***inetd***, es porque previamente había una solicitud de la cual el ***Wrapper*** recibe el nombre o la dirección del ***host*** cliente, junto con el servicio requerido; esta información es chequeada y monitoreada, y de acuerdo a criterios de acceso establecidos por el administrador, niega u otorga el acceso. Cuando todo está bien, arranca el ***daemon*** del servicio deseado.

El programa ***Wrappers*** se basa sobre la información de la dirección fuente obtenida desde los paquetes de red, la cual no es ciento por ciento confiable, aunque es muy bueno para encontrar falsificaciones.

El ***Wrappers*** se encarga de registrar todo sobre una bitácora, en la cual se reporta el nombre del ***host*** cliente y el servicio requerido, sin intercambiar información con las aplicaciones del servidor o del cliente y sin introducir ***overhead*** en la conversación entre las aplicaciones ***cliente-servidor***. Cualquiera que sea la acción tomada, ***tcpd*** registra el intento de acceso.

El hecho de no tener interacción con las aplicaciones del cliente y del servidor tiene dos ventajas principales:

1) Los ***Wrappers*** son aplicaciones independientes; así que el mismo programa puede proteger muchos tipos diferentes de servicios de red.

2) La no interacción también permite que los ***Wrappers*** sean invisibles en el extremo del cliente (excepto, para usuarios autorizados).

Otra propiedad importante es que estos programas son activados solamente en el contacto inicial entre cliente y servidor. Una vez que un ***Wrapper*** ha realizado este trabajo el deja libre la conexión ***cliente-servidor***, sin introducir ninguna clase de información redundante, de control u otro tipo de ***overhead***.

Debido a que estos demonios liberan la conexión ***cliente-servidor***, una vez ésta es establecida, los ***Wrappers*** solo pueden ver el intento de la conexión.

Existen dos formas de usar los programas ***Wrappers***:

1) La forma fácil: Se mueven los demonios de los servicios de red a algún otro directorio y se llenan los huecos resultantes con copias de los programas ***Wrappers***. Este enfoque no involucra cambios en los archivos de configuración del sistema, así que hay muy poco peligro de una caída del sistema.

2) La forma avanzada: Permite modificar la configuración del archivo ***inetd (/etc/inetd.conf)*** para que, en vez de ejecutar el servidor, ejecute el ***tcpd*** para los servicios que se desee monitorear.

Por ejemplo:

ftp stream tcp nowait root /usr/etc/ftpd ftpd, se convierte en:

ftp stream tcp nowait root /usr/etc/tcpd ftpd

Otro ejemplo:

tftp dgram udp wait root /usr/etc/tcpd in.tftpd -s /tftpboot

Cuando llega una solicitud ***tftp, inetd*** correrá el programa ***Wrapper*** con un proceso llamado ***'in.tftpd'***. Este es el nombre que el ***Wrapper*** usará cuando captura la solicitud y cuando busca las tablas adicionales de control de acceso. ***'in.tftpd'***, es también el nombre del programa servidor que el ***Wrapper*** intentará correr cuando todo este bien. Todos los argumentos (***in.tftpd -s /tftpboot*** en este caso particular) son transferidos transparentemente al programa servidor.

5.4 CRACK

Esta herramienta de seguridad fue desarrollada por ***Alec Muffet***. Es otra herramienta para analizar la seguridad de una plataforma **UNIX**. Consta de un programa adivinador de ***passwords*** que es diseñado para localizar rápidamente inseguridad en el archivo de ***password*** de **UNIX** (***/etc/passwd***), escaneando el contenido del mismo, buscando a usuarios que han escogido una mala combinación ***login-password.***

Básicamente ***crack*** establece su objetivo en lograr obtener un ***password*** del archivo ***/etc/passwd*** de un sistema **UNIX**. Para entrar en sistemas **UNIX** se necesita un nombre de usuario (***login***)y un ***password***; este archivo contiene la lista de los nombres de usuarios y los ***passwords***, así como otra información asociada a cada ***login***. El problema es que los ***passwords*** están ecriptados.

Cada línea del archivo corresponde a un usuario, y contiene, por orden, el nombre de usuario, el ***password*** encriptado, el ***UID***, el ***GID***, el nombre real, el directorio en el que está su cuenta y el ***shell*** que se carga al inicio. Cada campo está separado por "***:***" del anterior y del siguiente. En algunas líneas, el campo ***password*** es "*******". Este ***password*** encriptado es inválido, o sea, no se corresponde con ningún ***password***; por tanto, las cuentas que tienen alguna "*" en el campo ***password*** son cuentas a las que se puede entrar. Otro aspecto importante a tener en cuenta son los usuarios cuyo ***UID*** es "0", como el ***root***. Estos usuarios son ***root*** a todos los efectos, o sea, tienen los mismos derechos que el ***root***. Los usuarios que sin tener el ***UID=0*** tienen el mismo ***GID*** que el ***root***, también tienen algunos privilegios por el hecho de pertenecer al grupo del ***root.***.

El mecanismo de encriptación de **UNIX** no es reversible, esto es, no se puede desencriptar. Para comprobar si el usuario es el correcto lo que hace es encriptar el ***login*** y comparar el ***password*** encriptado con lo que hay en el archivo. Si coincide es que el ***password*** ingresado corresponde al del archivo encriptado.

El mecanismo o funcionamiento que utiliza ***crack*** para atacar un archivo de ***passwords*** de **UNIX** es precisamente el mismo que se describió anteriormente, adicionándole ciertas modificaciones. Lo que ***crack*** hace es encriptar palabras y comprobar cada una de ellas si coincide con el ***password*** encriptado. Si coincide, se obtiene un ***password***, y si no, prueba la siguiente palabra.

Para que ***crack*** realice esta operación necesita de ciertos mecanismos de ayuda que son : una librería, un diccionario de palabras y el archivo de ***passwords*** a probar.

La librería es un programa que tiene una pequeña implementación del algoritmo ***DES*** y permite agilizar y optimizar el proceso de encontrar un ***password***. Esta librería está configurada para que trabaje de acuerdo a las diferentes implementaciones del algoritmo ***cript***, función que utilizan algunos sistemas **UNIX** para encriptar los ***passwords***.

El diccionario es configurado por el usuario de acuerdo con ciertos parámetros definidos por el sistema o por el mismo, o puede ser invocado por comandos externos. Este diccionario generalmente es creado por opciones que tiene ***crack***.

Crack no está destinado a atacar ningún servicio en especial de cualquier sistema **UNIX**, la función primordial es lograr obtener una combinación ***login/password*** que sea correcta para acceder al sistema como un usuario legal. Con una correcta ejecución de ***crack*** y un resultado positivo del mismo, se tiene en cuenta

las vulnerabilidades que hay en el sistema, en lo que hace referencia al archivo de ***passwords***. Esto debe ser primordial ya que es la primera línea de ataque para cualquier usuario no autorizado a un sistema de cómputo

6. CONCLUSIONES Y RECOMENDACIONES

Conectar una máquina a una red, y sobre todo si esa red es ***Internet***, es una decisión que puede traer muchísimas ventajas, pero también grandes problemas. Si la máquina en cuestión va a ofrecer alguna clase de servicios al exterior, estos problemas se pueden multiplicar de manera alarmante.

Esto puede desilusionar al más aventurado administrador de sistemas **UNIX**. Sin embargo, la seguridad no es un objetivo inalcanzable. En este proyecto se han presentado múltiples técnicas y herramientas que, aplicadas correctamente y con los criterios apropiados, pueden permitirle a un sistema conectado a una red bajo plataforma **UNIX**, contar con un nivel de seguridad apropiado para las necesidades de sus usuarios y sus administradores, y aún así proporcionar al exterior servicios útiles e importantes.

Como siempre, sin embargo, es importante recordar que ninguna herramienta, por sofisticada que sea, ningún archivo de configuración, por correcto que parezca; ningún servidor, por depurado que esté, puede evitar todos los problemas. Y estos problemas inesperados solamente pueden ser detectados y solucionados por un administrador bien capacitado, y que esté alerta de lo que sucede en su máquina.

En el proyecto se trató la seguridad de la plataforma **UNIX**, desde el nivel del sistema operativo y con la ayuda de las herramientas ***ISS***, ***Satan*** y ***Crack***, las cuales indudablemente sirven para obtener una configuración adecuada de los servidores. Sin embargo, existen aspectos que se salen del marco técnico y que están íntimamente ligados con la seguridad y deben ser tenidos en cuenta. Como resultado de la investigación se emiten unas recomendaciones que incluso pueden ser generalizadas a cualquier sitio **UNIX** que provee servicios ***Internet***:

- Se deben crear y adherir políticas de seguridad.
- Realizar continuo monitoreo verificando la integridad de archivos importantes, corriendo los programas ***ISS***, ***Satan***, ***Crack*** y otros de acuerdo a las partes descritas en el manual.
- Instalar parches donde se encuentren problemas de seguridad.
- Es importante tener mecanismos de filtrado de paquetes, tales como ***Tcp Wrappers***, lo cual puede ser hecho sobre todos los sistemas críticos, al menos que otros aspectos lo hagan imposible.
- Debe hacerse una evaluación de los servicios que deben correr.
- Solo deben permitirse conexiones a servicios externos cuando sean necesarios para la universidad y al menos que sean tan seguros como el sistema en cuestión.
- El sistema podría asegurarse usando las últimas técnicas y herramientas de software tanto como sea posible.
- La educación continua del sistema y administradores es esencial.

- Se deben hacer regularmente auditorías de seguridad y monitoreo.
- Solo se debe permitir acceso a los recursos propios, financieros y otros importantes de la universidad, al personal justo y necesario.

A continuación se destacan algunos aspectos particulares de vital importancia para una buena administración del sistema **UNIX**:

- Combinar mayúsculas y minúsculas en los ***passwords*** y exigir el cambio periódico de los mismos.
- Tener especial cuidado en dar de baja las cuentas de aquellas personas que ya no pertenecen a la corporación.
- Utilizar el comando ***umask*** (en los archivos ***.cshrc*** y ***.profile***) para controlar los permisos de los archivos que se crean.
- Verificar que se dispone de una versión de ***ftpd*** posterior a diciembre de 1988 ya que un ***bug*** de versiones anteriores permitía entrar como ***root*** vía ***ftp*** anónimo.
- Con algunas excepciones (ejemplo: terminales) todos los dispositivos deben pertenecer al ***root***.
- Deben evitarse correr ***daemons*** que no se utilizan.
- Se deben encriptar las ***passwords*** que circulan por la red.
- Intentar evitar servicios de ***rlogin*** y ***rsh*** pues son inseguros.
- Si se abre la red a ***Internet***, es conveniente tener una sola máquina conectada que puede ser configurada como ***firewall***.

- El administrador debe comprobar que no haya programas inseguros instalados como ***setuid*** y ***setgid***.

Otro aspecto de la seguridad de los datos en un sistema es la verificación de la integridad de los archivos. Es muy común que un intruso (o un error de software, o un administrador descuidado) modifique algún archivo del sistema, y que posteriormente dicha modificación provoque un mal funcionamiento o un hueco de seguridad. **UNIX** es un sistema tan complejo que es imposible, en muchas ocasiones, que el administrador localice esas modificaciones, sobre todo si han sido realizadas cuidadosamente por un intruso para permitirle acceso posterior a la máquina. Por lo tanto, es importante monitorear de forma automática la integridad de los archivos importantes del sistema.

Basados en la investigación realizada, la utilización del servidor "***cóndor***", donde se hicieron las pruebas para verificar la adecuada configuración de archivos del sistema; detallamos los resultados obtenidos, con su respectivo análisis en el ***Anexo A***.

Es muy importante utilizar periódicamente las herramientas ***ISS*** y ***Satan***, ya que permiten conocer la información que un posible atacante obtendría de la red en el caso de que utilizase dichos programas contra el sistema. Una de las conclusiones más relevantes fue que ***Satan*** ya no es una herramienta tan poderosa como parece, debido a que su versión (Satan Versión 1.1) ya está muy obsoleta y no permite encontrar de una forma fiable las vulnerabilidades que se

supone que puede encontrar, en cambio ***ISS*** es más poderosa que ***Satan***, ésta herramienta si permitió encontrar las vulnerabilidades simuladas, pero el inconveniente es que su versión comercial es muy costosa.

Es de gran utilidad el uso de programas de chequeo de la seguridad de los ***passwords*** elegidos por los usuarios. En este aspecto, el mejor programa existente en la red es ***Crack*** , que permite utilizar un gran número de diccionarios y reglas de inferencia a la hora de verificar la calidad de un ***password***.

Para analizar la correcta estructura y configuración del archivo de ***password*** del servidor "***condor***", se corrió ***starcrack***, una versión poderosa del ***crack***. Al ejecutarse ésta herramienta sobre el archivo de ***passwords*** encriptados, reportó a seis (6) usuarios que tenían un ***password*** fácilmente de adivinar. Por lo tanto se recomienda a los administradores en general seguir con las pautas propuestas en este proyecto.

BIBLIOGRAFIA

INTERNET

INSTITUTO COLOMBIANO DE NORMAS TECNICAS Y CERTIFICACION. Normas Colombianas para la presentación de tesis de grado. Bogotá: ICONTEC, 1998. p. 4 - 100.

KARANJIT Siyan, y CHRIS Hare. Internet y la Seguridad en Redes, Domine las complejidades de la seguridad de la red. Editorial Prentice Hall. Segunda Edición. México 1995

MANUAL DE LA MATERIA SEGURIDAD EN REDES. Manual Teórico - Práctico.

MANUAL FUNDAMENTOS EN UNIX. Módulo Teórico - Práctico

STALLINGS, Wiiliam. Network and Internetwork Security, Principles and Practice Internet. Editorial Prentice Hall. Segunda Edición.

ANEXOS

Anexo A. Resultados de las pruebas realizadas en la configuración de los archivos del sistema UNIX del servidor "cóndor"

1. FUNCIONES DE AUDITORIA

<table>
<tr><td colspan="2"><u>Objetivo 1.1:</u>
Asegurarse que la auditoria del subsistema esté activada y esté configurada de una forma segura.</td></tr>
<tr><td>Paso 1</td><td>Ejecute el siguiente comando para verificar que la auditoria este activada:
<code>/usr/sbin/auditconfig -getcond</code></td></tr>
<tr><td colspan="2">Resultados esperados:
La opción –getcond es la condición de auditoría de la máquina. La respuesta es una las siguientes:
<code>Auditing - Auditing is enabled and turned on</code>
<code>No audit - Auditing is enabled but not turned on</code>
<code>Disabled - Auditing is not enabled</code>
Un Mensaje de error con el siguiente formato <code>"auditconfig: error = Invalid argument(22)"</code> indica que la opción BSM no está disponible y no puede ser usada.</td></tr>
<tr><td colspan="2">Comentarios:
El modelo básico de seguridad debe ser instalado y puesto en marcha
EL Script <code>/etc/security/bsmconv</code> permite las características BSM de Solaris.</td></tr>
</table>

RESULTADOS OBTENIDOS

Auditconfig: auditon(2) failed.

Auditconfig: error = Invalid argument(22)

La salida no es la esperada en el servidor cóndor

Objetivo 1.1:

Verificar que la auditoria del ***Kernel*** no haya sido modificada inapropiadamente

Paso 1	Mire los siguientes archivos: `/etc/security/audit_event` Compare su contenido con el archivo por defecto de los eventos de auditoria del ***Kernel*** que vienen equipados con ***Solaris***.

Resultados esperados:

Cualquier modificación en los eventos de auditoría del ***Kernel*** deben ser justificados

Comentarios:

Las acciones del sistema que son auditables, son definidas como eventos de auditoría en el archivo ***`/etc/security/audit_event.`*** Cada evento auditable es definido por un nombre simbólico, y número de evento, etc.

Resultados obtenidos

Formato del archivo: ***event number:event name:event description:event classes (comma separated)***

Se obtuvo una lista grande con este formato ej:

System Adminstrators: Do NOT modify or add events with an event number Less

than 32768. These are reserved by the system.

0 Reserved as an invalid event number.

1 - 2047 Reserved for the SunOS kernel.

2048 - 32767 Reserved for the SunOS TCB programs.
32768 - 65535 Available for third party TCB
applications.
#
6144 - 32767 SunOS 5.X user level audit events
LA SALIDA FUE LA SIGUIENTE:
19:AUE_MCTL:mctl(2):fm
69:AUE_FCHROOT:fchroot(2):pc
150:AUE_AUDITSTAT:auditstat(2):ad
6148:AUE_crontab_create:crontab-crontab created:ad
Hay que compararla con la del kernel equipado por defecto

Objetivo 1.2:

Asegúrese de que la auditoria esté correctamente configurada y que reúna los eventos de auditoría requeridos.

Paso 1	Ejecute el siguiente comando: `/usr/sbin/auditconfig -chkconf`

Resultados esperados:

El comando no debería mostrar ninguna inconsistencia en el mapeo auditado. De lo contrario, este comando no mostraría ningún mensaje.

Comentarios:

El módulo básico de seguridad debería ser instalado y puesto en marcha. La opción ***–chkconf*** chequea la configuración de los eventos de auditoría del Kernel para clasificar el mapeo y reportar cualquier inconsistencia

Objetivo 1.3:	
Verificar que las clases de auditoría no hallan sido modificadas	
Paso 1	Ejecute el siguiente comando: `More /etc/security/audit_class` Compare su contenido con el archivo por defecto de los eventos de auditoria del Kernel que vienen equipados con Solaris.2.5.1
Resultados esperados: Los archivos que coincidan, indican que la auditoría de clases no ha sido modificada inapropiadamente	
Comentarios: # # User Level Class Masks # # Developers: si cambia este archivo también debe cambiar audit.h # # File Format: # mask: name: description # 0x00000000: no: invalid class 0x00000001: fr: file read 0x00000002: fw: file write 0x00000004: fa: file attribute access 0x00000008: fm: file attribute modify 0x00000010: fc: file create 0x00000020: fd: file delete 0x00000040: cl: file close 0x00000080: pc: process 0x00000100: nt: network 0x00000200: ip: ipc 0x00000400: na: non- attribute 0x00000800: ad: administrative 0x00001000: lo: login or logout Cada evento auditado es definido como perteneciente a una clase o clases de auditoría. Por asignación de eventos a clases, un administrador puede más fácilmente tratar con un gran número de eventos. Cuando llamamos una clase, simultáneamente una dirección llama a todos lo eventos de esa clase. En todo caso un evento auditable es grabado en un registro dependiendo si el administrador selecciona una clase de auditoría que incluye el evento específico.	
Resultados obtenidos	

Developers: If you change this file you must also edit audit.h.

#

File Format:

#

mask:name:description

#

0x00000000:no:invalid class

0x00000001:fr:file read

0x00000002:fw:file write

0x00000004:fa:file attribute access

0x00000008:fm:file attribute modify

0x00000010:fc:file create

0x00000020:fd:file delete

0x00000040:cl:file close

0x00000080:pc:process

0x00000100:nt:network

0x00000200:ip:ipc

0x00000400:na:non-attribute

0x00000800:ad:administrative

Compararlos con el que viene equipado con Solaris, aunque la salida del manual es como la que se obtuvo.

Objetivo 1.4:	
Verificar que el intento de conexiones sin éxito aparezcan registradas	
Paso 1	Verifique que los siguientes archivos existan y tengan los permisos aceptables. `ls -l /var/adm/loginlog`

Resultados Esperados: El archivo ***/var/adm/loginlog*** debe existir en el host y tenga los siguientes permisos -rw------- 1 root sys 2073 Aug 23 13:05 /var/adm/loginlog Si el archivo no existe, nadie está conectado.
Resultados obtenidos No existe el archivo ***loginlog***, en el servidor cóndor TIENE OTRO NOMBRE***(sulog)*** -rw------- 1 root root(aquí debería ser sys, según el manual) 2244 Feb 27 10:55 sulog

Objetivo 1.4:	
Verificar que el demonio ***syslogd*** se esté ejecutando y esté configurado seguramente.	
Paso 1	Ejecute el siguiente comando: `ps –eaf | grep syslog`
Resultados Esperados: La salida debe parecerse a la siguiente: ps –eaf | grep syslog root 309 1 0 16:23:24 ? 0:00 /usr/sbin/syslogd	
Paso 2	Ejecute el siguiente comando: `Ls –l /etc/syslog.conf`
Resultados Esperados: La configuración es: no es de todos, ni del grupo y el propietario debe ser el root	
Paso 3	Ejecute el siguiente comando: `/bin/more /etc/syslog.conf`

Resultados Esperados:

La configuración del archivo debe tener lo siguiente:

```
mail.debug              /var/log/syslog
*.info;mail.none        /var/adm/messages
*.alert                 /dev/console
*.alert                 root
*.emerg                 *
```

Esto conectará las entradas de correo al archivo ***/var/log/syslog*** y lo demás al archivo ***/var/adm/messages***

Resultados obtenidos

Paso 1.

```
root   150     1  0   Feb 24 ?        0:00 /usr/sbin/syslogd root  1953  1828  0 11:32:18
pts/2    0:00 grep syslog
```

La salida es la esperada

Paso 2.

```
-rw-r--r--   1 root sys 1317 Oct  3 10:59 etc/syslog.conf
```

La salida es 644.

Paso 3.

```
#ident  "@(#)syslog.conf 1.3 93/12/09 SMI"   /* SunOS 5.0 */
        #
        # Copyright (c) 1991-1993, by Sun Microsystems, Inc.
        #
        # syslog configuration file.
        #
        # This file is processed by m4 so be careful to quote (`') names
# that match m4 reserved words.  Also, within ifdef's, arguments
# containing commas must be quoted.
#
# Note: Have to exclude user from most lines so that user.alert
#       and user.emerg are not included, because old sendmails
```

```
#       will generate them for debugging information.  If you
#       have no 4.2BSD based systems doing network logging, you
#       can remove all the special cases for "user" logging.
#
*.err;kern.notice;auth.notice;user.none         /dev/console
*.err;kern.debug;daemon.notice;mail.crit;user.none     /var/adm/messages

        *.alert;kern.err;daemon.err;user.none           operator
        *.alert;user.none                       root
```

La salida es la esperada.

Objetivo 1.6:

Identificar cualquier usuario para los cuales la auditoría haya sido desactivada

Paso 1	Teclee el siguiente comando: `/bin/more /etc/security/audit_user` Identifique cualquier usuario contenido en el archivo ***/etc/passwd***, que no esté en el archivo ***/etc/security/audit_user***.

Resultados Esperados:

El archivo tuiene una línea al comienzo con el nombre del usuario, para cada usuario autorizado. También las clases no auditadas en el archivo ***audit_control***, están listadas después de la segunda coma, para cualquier línea de usuario en este archivo. El nivel de auditoría del sistema se aplica a todos los usuarios, a menos que el usuario tenga una entrada en el archivo ***/etc/security/audit_users***. El ninel de auditoría del usuario sobrepasa al nivel de auitoría del sistema. Los campos en el archivo están separados por columnas y están definidos como sigue:

Username: always audit flags: never audit flags

Comentarios:

Todos los usuarios deben estar sujetos a auditoría

Resultados obtenidos

username:always:never

#root:lo:no

El único usuario es el root, deberían estar más usuarios o por lo menos los que están en el archivo passwd y estar sujetos a auditoría

Objetivo 1.7:

Verificar los parámetros requeridos identificados para cada evento de auditoría grabado, incluyendo fecha y hora del evento, identificación del usuario, tipo de evento, éxito o fracaso del mismo.

Paso 1	Como un usuario sin privilegio intente ver el archivo `/etc/security/audit_control` Usando el comando: `/usr/bin/more /etc/security/audit_control`

Resultados Esperados:

Los permisos para ver el archivo son denegados para un usuario sin privilegios.

Paso 2	Como root intente ver el archivo `/etc/security/audit_control` Usando el comando: `/usr/bin/more /etc/security/audit_control`

Resultados Esperados:

La configuración de eventos auditados en el sistema, muestran como mínimo los siguientes eventos que son auditados:

(ad) Operación administrativa normal.

(lo) Conexión y desconexión.

(fc) Creación del objeto

(fd) Eliminación de objetos

(fw) Fallas para escribir a un archivo

Comentarios:

El archivo ***audit_control***, muestra los archivos de auditoría del sistema y las configuraciones del demonio de auditoría***(auditd)***: Cada línea consiste de un título y una cadena, separada por una coma. El administrador del sistema define cuatro grupos de líneas en el archivo audit_control:

1. La línea ***audit flags(flags)***, contiene la bandera de auditoría que define que clases de eventos son auditados por los usuarios en la máquina.
2. La línea ***non-attributable flags(naflags)***, contiene la bandera de auditoría que define, qué clases de eventos son auditados cuando una acción específica no puede ser atribuida a un usuario específico.
3. La línea audit ***threshold(minfree)***, define el nivel mínimo de espacio libre para todos los filesystems auditables, el porcentaje del minfree debe ser mayor o igual a cero, por defecto es 20%.
4. La línea ***directory definition (dir)***, define cada filesystem auditado y directorios en la máquina que se usarán para almacenar estos grupos de archivos auditados.

El archivo ***audit_user*** almacena por usuario los datos auditados por preselección. Cada entrada en este archivo tiene la forma:

Username: always-audit-flags: never-audit-flags

Resultados obbtenidos

Paso 1.

naflags:: not found

Los resultados son los esperados

Paso 2.

dir:/var/audit

flags:

minfree:20

naflags:

Los resultados son los esperados

Objetivo 1.8:	
Verifcar que el script ***audit_warn*** no haya sido modificado inapropiadamente.	
Paso 1	Digite el siguiente comando: `/usr/bin/more /etc/security/audit_warn`
Resultados Esperados: El archivo **`/etc/security/audit_warn`** no ha sido cambiado por el que viene por defecto con Solaris 2.5.1.	
Comentarios: Cuando el ***demonio auditd*** encuentra una condición inusual mientras escribe los registrios de auditoría, este invoca el script ***`/etc/security/audit_warn`***. Este script puede ser personalizado por sitios individuales para advertir las condiciones que quizá requerirá intervención manual o que se maneje automáticamente. Para todas las condiciones de error, ***`audit_warn`*** escribe un mensaje un mensaje en la consola y envía uno al alias ***"audit_warn"***.	
Resultados obtenidos La salida hay que redireccionarla a una arhivo porque es muy extensa Presenta un script, hay que compararlo con el equipado solaris	

Objetivo 1.9:	
Verificar que el alias ***"audit_warn"*** haya sido configurado correctamente.	
Paso 1	Digite el siguiente comando: `/usr/bin/more /etc/aliases`
Resultados Esperados:	

Una entrada debe aparecer para el alias ***"audit_warn"***, y el alias debe ser el nombre de una cuenta actual.

```
#ident "@(#) aliases 1.13 92/ 07/ 14 SMI" /* SVr4.0 1.1 */
```

Los alias pueden tener cualquier mezcla de carácter minúscula o mayúscula en la parte izquierda pero en la parte derecha de tener un solo tipo (usualmente minúscula).
El programa ***"newaliases"*** necesitará ser ejecutado después
NOTA: Este archivo es actualizado por cualquier cambio a través de sendmail

 # @(#) aliases 1.8 86/ 07/ 16 SMl ##

Los siguientes alis son requeridos por el protocolo de correo, RFC 822.
Coloque la dirección de una persona que tenga relación con este sistema de correo

audit_ warn: sysadmin, root
#
#

Comentarios:

Resultados obtenidos

dent "@(#)aliases 1.13 92/07/14 SMI" /* SVr4.0 1.1 */
##
Aliases can have any mix of upper and lower case on the left-hand side,
but the right-hand side should be proper case (usually lower)
#
#>>>>>>>>>> The program "newaliases" will need to be run after
>> NOTE >> this file is updated for any changes to
>>>>>>>>>> show through to sendmail.

#
#	@(#)aliases 1.8 86/07/16 SMI
##	# Following alias is required by the mail protocol, RFC 822
Set it to the address of a HUMAN who deals with this system's mail problems.
Postmaster: root
Alias for mailer daemon; returned messages from our MAILER-DAEMON
should be routed to our local Postmaster.
MAILER-DAEMON: postmaster
Aliases to handle mail to programs or files, eg news or vacation
decode: "|/usr/bin/uudecode"
La salida es la esperada.

Objetivo 1.10:

Verificar que los datos auditados estén protegidos por el sistema de manera que el acceso a estos es limitado solamente para los que tienen autorización de ver los datos auditados. En resumen, verificar que los datos auditados están protegidos de cambios o eliminación por usuarios en general

Paso 1	Como root, determine el nombre de los archivos auditados mostrados en una línea que inicia con ”dir:” en el archivo `/etc/security/audit_control`. Para cada nombre de archivo listado en el archivo `audit_control, revise los permisos de archivo usando el siguiente comando` ls -l

Resultados Esperados:

El propietario de todos los archivos debe ser el root y de be estar en el modo 0600.

Resultados obtenidos

Total 0 No se presento ninguna salida, debe haber un error de configuración

2. DISPONIBILIDAD

Objetivo 2.0	
Verificar que el sistema es respaldado en bases regulares.	
Paso 1	Ver el archivo crontab para determinar si el sistema es respaldado automáticamente con base en un plan. Del libro de registros del sistema o del administrador del sistema, determine cuando la última copia fue hecha. Determine si las cintas de backup están correctamente etiquetadas.
Resultados Esperados:	
Los backups son hechos regularmente por procedimientos operacionales.	
Resultados obtenidos	
La salida es ilegible, presenta una serie de caracteres ascii ilegibles No se puede ver el archivo	

Objetivo 3.6:
Identificar todos los archivos modificables y verificar su necesidad. Los archivos modificables por todos, directorios y servicios representan un hueco potencial de seguridad en el sistema.

Paso 1 — Como root, Ejecute los siguientes comandos:
`/usr/bin/find / -type f \( -perm -2 -o -perm -20 \) -exec ls -ldb {} \;`
 	
`/usr/bin/find / -type d \( -perm -2 -o -perm -20 \) -exec ls -ldb {} \;`

Resultados Esperados: No hay palabras inesperadas en archivos o directorios. Los archivos deben ser modificables solamente si hay un requerimiento legítimo.
Comentarios: Los siguientes archivos estuvieran más seguros si permanecieran modificables por todos los usuarios ***/tmp*** ***/var/tmp*** ***/var/preserve*** ***/var/mail***

Resultados obtenidos

La salida es una gran lista archivos, los cuales no deben tener permiso de escritura para todos, solamente para el dueño y su grupo.

Se encontró permiso de escritura en los siguientes archivos:

-rw-rw-rw- 1 root root 0 Oct 3 10:57 /var/sadm/install/.pkg.lock
-rw-rw-rw- 1 root root 5722 Mar 2 21:27 /var/adm/log/asppp.log
-rw-rw-rw- 1 bin bin 0 Oct 3 10:59 /var/adm/spellhist
-rw-rw-rw- 1 root root 0 Oct 3 10:57 /var/sadm/install/.pkg.lock
-rw-rw-rw- 1 root root 5722 Mar 2 21:27 /var/adm/log/asppp.log
-rw-rw-rw- 1 bin bin 0 Oct 3 10:59 /var/adm/spellhist
-rw-rw-rw- 1 root root 5436 Mar 2 21:28 /var/saf/_log
-rw-rw-rw- 1 root root 656 Mar 2 21:27 /var/lp/logs/lpsched
-rw-rw-rw- 1 root root 915 Mar 2 21:27 /var/lp/logs/lpNet
-rw-rw-rw- 1 root other 574 Mar 1 21:01 /etc/group
-rw-rw-rw- 1 root root 0 Feb 27 12:52 /etc/.pwd.lock
-rw-rw-rw- 1 root other 131 Jan 26 21:48 /etc/resolv.conf
-rw-rw-rw- 1 root other 884736 Oct 3 12:09 /pub/satan/satan.tar
-rw-rw-rw- 1 root other 1167835 Oct 7 19:39 /pub/gcc/libg++-2_7_1.gz
-rw-rw-rw- 1 root other 4290560 Oct 3 12:14 /pub/iss/iss-Solaris.tar
-rw-rw-rw- 1 root other 5294080 Oct 3 12:14 /pub/iss/iss-SunOS.tar
-rw-rw-rw- 1 root other 9451520 Oct 3 12:18 /pub/perl/perl5.004_04.tar
-rw-rw-rw- 1 root other 2336469 Oct 5 21:57 /pub/browser/netscape-v30-export_x86-unknown-linux-elf_tar.gz
-rwxrwxrwx 1 root other 6852 Oct 27 21:03 /pub/winnuke/ataque
-rw-rw-rw- 1 root other 71816 Oct 28 21:51 /pub/gzip/gzip
-rw-rw-rw- 1 root other 50697 Feb 27 13:01 /pub/guess.zip
-rw-rw-rw- 1 root other 16011 Feb 27 13:01 /pub/hackunix.zip
-rw-rw-rw- 1 root other 122657 Mar 2 19:06 /pub/jack14.zip

```
-rw-rw-rw-   1 root     other      34014 Feb 27 13:01 /pub/unixhack.zip
-rw-rw-rw-   1 root     other       1477 Feb 27 13:02 /pub/unixpass.zip
-rw-rw-rw-   1 root     other      69671 Mar  3 19:23 /pub/john.zip
-rw-rw-rw-   1 root     other      70798 Mar  3 19:37 /pub/tpudict.zip
-rw-rw-rw-   1 root     other     427520 Mar  3 19:38 /pub/MATUNIX.doc
-rw-rw-rw-   1 root     other      95222 Feb 27 10:52 hagualin/secureSolaris.html
-rw-rw-rw-   1 root     other     292864 Feb 27 11:00 /hagualin/manual.doc
-rw-rw-rw-   1 root     other      50697 Feb 27 13:00 /guess.zip
```

Objetivo 3.8:	
Verifique que los archivos de inicio y apague del sistema son válidos y están protegidos.	
Paso 1	Como root, Ejecute el siguiente comando: `/usr/bin/find /etc \( -perm -2 -o -perm -20 \) -exec ls -ldb {} \;`
Resultados Esperados: No debe haber salida que indique que el directorio /etc y su contenido son modificables por todos los usuarios	
Paso 2	Revise la configuración de los scripts de inicio y apagada del sistema y la configuración de archivos en `/etc/rc[1-3].d`
Resultados Esperados: Caualquier tarea hecha en los scripts de inicio es ejecutada seguramente. Cualquier servicio iniciado o tarea ejecutada es aprobada. Cualquier directorio que contenga un script, ejecutable, o un archivo de configuración que es ejecutado en los ***scripts rc***, durante el boot o el reinicio no es escribible por un usuario diferente al root.	
Resultados obtenidos Paso 1	

```
Drwxrwxr-x  24 root     sys         3072 Mar  2 21:50 /etc
lrwxrwxrwx   1 root     root          14 Oct  3 10:58 /etc/TIMEZONE ->
./default/init
lrwxrwxrwx   1 root     root       14 Oct  3 10:58 /etc/aliases -> ./mail/aliases
lrwxrwxrwx   1 root     root       16 Oct  3 10:58 /etc/autopush -> ../sbin/autopush
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/clri -> ../usr/sbin/clri
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/crash -> ../usr/sbin/crash
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/cron -> ../usr/sbin/cron
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/dcopy -> ../usr/sbin/dcopy
drwxrwxr-x   2 root     sys          512 Oct  3 11:24 /etc/default
drwxrwxr-x   2 root     sys          512 Oct  3 10:58 /etc/dfs
lrwxrwxrwx   1 root     root          14 Oct  3 10:58 /etc/ff -> ../usr/sbin/ff
lrwxrwxrwx   1 root     root          19 Oct  3 10:58 /etc/fmthard ->
../usr/sbin/fmthard
lrwxrwxrwx   1 root     root          18 Oct  3 10:58 /etc/format ->
../usr/sbin/format
drwxrwxr-x   6 root     sys          512 Oct  3 10:58 /etc/fs
drwxrwxr-x   2 root     sys          512 Oct  3 10:58 /etc/fs/proc
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/fsck -> ../usr/sbin/fsck
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/fsdb -> ../usr/sbin/fsdb
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/fstyp -> ../usr/sbin/fstyp
lrwxrwxrwx   1 root     root          21 Oct  3 10:58 /etc/getty ->
../usr/lib/saf/ttymon
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/grpck -> ../usr/sbin/grpck
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/halt -> ../usr/sbin/halt
lrwxrwxrwx   1 root     root          12 Oct  3 10:58 /etc/hosts -> ./inet/hosts
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/inetd.conf ->
./inet/inetd.conf
lrwxrwxrwx   1 root     root          12 Oct  3 10:58 /etc/init -> ../sbin/init
drwxrwxr-x   2 root     sys         1024 Oct  3 11:22 /etc/init.d
```

```
lrwxrwxrwx   1 root     root          19 Oct  3 10:58 /etc/install -> ../usr/sbin/install
lrwxrwxrwx   1 root     root          19 Oct  3 10:58 /etc/killall -> ../usr/sbin/killall
lrwxrwxrwx   1 root     root          19 Oct  3 10:58 /etc/labelit -> ../usr/sbin/labelit
drwxrwxr-x   2 root     sys          512 Oct  3 10:58 /etc/lib
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/lib/pam_authen.so ->
./pam_authen.so.1
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/lib/pam_entry.so ->
./pam_entry.so.1
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/lib/pam_extern.so ->
./pam_extern.so.1
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/lib/pam_pwmgt.so ->
./pam_pwmgt.so.1
lrwxrwxrwx   1 root     root          18 Oct  3 10:58 /etc/lib/pam_session.so ->
./pam_session.so.1
lrwxrwxrwx   1 root     root          14 Oct  3 10:58 /etc/log -> ../var/adm/log
drwxrwxr-x   2 bin      mail         512 Oct  3 11:29 /etc/mail
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/mkfs -> ../usr/sbin/mkfs
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/mknod ->
../usr/sbin/mknod
lrwxrwxrwx   1 root     root          13 Oct  3 10:58 /etc/mount -> ../sbin/mount
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/mountall ->
../sbin/mountall
lrwxrwxrwx   1 root     root          18 Oct  3 10:58 /etc/ncheck ->
../usr/sbin/ncheck
lrwxrwxrwx   1 root     root          15 Oct  3 10:58 /etc/netmasks ->
./inet/netmasks
lrwxrwxrwx   1 root     root          15 Oct  3 10:58 /etc/networks ->
./inet/networks
drwxrwxr-x   4 root     sys          512 Oct  3 11:22 /etc/opt
lrwxrwxrwx   1 root     root          31 Oct  3 10:59
```

```
/etc/opt/PFUaga/bin/ag10config -> ../../../../usr/sbin/ag10config
lrwxrwxrwx   1 root     root          29 Oct  3 10:59
/etc/opt/PFUaga/bin/aga.ucode -> ../../../../usr/lib/aga.ucode
drwxrwxr-x   3 root     sys          512 Oct  3 11:22 /etc/opt/SUNWleo
drwxrwxr-x   2 root     sys          512 Oct  3 11:22 /etc/opt/SUNWleo/bin
lrwxrwxrwx   1 root     root          29 Oct  3 11:22
/etc/opt/SUNWleo/bin/leo.ucode -> ../../../../usr/lib/leo.ucode
lrwxrwxrwx   1 root     root          30 Oct  3 11:22
/etc/opt/SUNWleo/bin/leoconfig -> ../../../../usr/sbin/leoconfig
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/protocols ->
./inet/protocols
lrwxrwxrwx   1 root     root          19 Oct  3 10:58 /etc/prtconf ->
../usr/sbin/prtconf
lrwxrwxrwx   1 root     root          19 Oct  3 10:58 /etc/prtvtoc ->
../usr/sbin/prtvtoc
lrwxrwxrwx   1 root     root          11 Oct  3 10:58 /etc/rc0 -> ../sbin/rc0
drwxrwxr-x   2 root     sys          512 Oct  3 11:16 /etc/rc0.d
lrwxrwxrwx   1 root     root          11 Oct  3 10:58 /etc/rc1 -> ../sbin/rc1
drwxrwxr-x   2 root     sys          512 Oct  3 11:13 /etc/rc1.d
lrwxrwxrwx   1 root     root          11 Oct  3 10:58 /etc/rc2 -> ../sbin/rc2
drwxrwxr-x   2 root     sys         1024 Oct  3 11:29 /etc/rc2.d
lrwxrwxrwx   1 root     root          31 Oct  3 11:21 /etc/rc2.d/S89bdconfig ->
../init.d/buttons_n_dials-setup
lrwxrwxrwx   1 root     root          11 Oct  3 10:58 /etc/rc3 -> ../sbin/rc3
drwxrwxr-x   2 root     sys          512 Oct  3 10:58 /etc/rc3.d
lrwxrwxrwx   1 root     root          11 Oct  3 10:58 /etc/rc5 -> ../sbin/rc5
lrwxrwxrwx   1 root     root          11 Oct  3 10:58 /etc/rc6 -> ../sbin/rc6
lrwxrwxrwx   1 root     root          11 Oct  3 10:58 /etc/rcS -> ../sbin/rcS
drwxrwxr-x   2 root     sys          512 Oct  3 11:17 /etc/rcS.d
lrwxrwxrwx   1 root     root          18 Oct  3 10:58 /etc/reboot ->
```

```
../usr/sbin/reboot
lrwxrwxrwx   1 root     root          15 Oct  3 10:58 /etc/rmt -> ../usr/sbin/rmt
lrwxrwxrwx   1 root     root          21 Oct  3 10:58
/etc/security/audit/localhost/files -> ../../../../var/audit
lrwxrwxrwx   1 root     root          15 Oct  3 10:58 /etc/services -> ./inet/services
lrwxrwxrwx   1 root     root          18 Oct  3 10:58 /etc/setmnt ->
../usr/sbin/setmnt
lrwxrwxrwx   1 root     root          20 Oct  3 10:58 /etc/shutdown ->
../usr/sbin/shutdown
lrwxrwxrwx   1 root     root          15 Oct  3 10:58 /etc/sulogin -> ../sbin/sulogin
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/swap -> ../usr/sbin/swap
lrwxrwxrwx   1 root     root          15 Oct  3 10:58 /etc/swapadd ->
../sbin/swapadd
lrwxrwxrwx   1 root     root          18 Oct  3 10:58 /etc/sysdef ->
../usr/sbin/sysdef
lrwxrwxrwx   1 root     root          15 Oct  3 10:58 /etc/tar -> ../usr/sbin/tar
lrwxrwxrwx   1 root     root          12 Oct  3 10:58 /etc/telinit -> ../sbin/init
drwxrwxr-x   2 root     sys          512 Oct  3 10:58 /etc/tm
lrwxrwxrwx   1 root     root          14 Oct  3 10:58 /etc/uadmin -> ../sbin/uadmin
lrwxrwxrwx   1 root     root          14 Oct  3 10:58 /etc/umount -> ../sbin/umount
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/umountall ->
../sbin/umountall
lrwxrwxrwx   1 root     root          15 Oct  3 10:58 /etc/utmp -> ../var/adm/utmp
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/utmpx -> ../var/adm/utmpx
lrwxrwxrwx   1 root     root          19 Oct  3 10:58 /etc/volcopy ->
../usr/sbin/volcopy
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/wall -> ../usr/sbin/wall
lrwxrwxrwx   1 root     root          17 Oct  3 10:58 /etc/whodo ->
../usr/sbin/whodo
lrwxrwxrwx   1 root     root          15 Oct  3 10:58 /etc/wtmp -> ../var/adm/wtmp
```

```
lrwxrwxrwx   1 root     root          16 Oct  3 10:58 /etc/wtmpx ->
../var/adm/wtmpx
-rw-rw-r--   1 root     sys          881 Jan  1  1970 /etc/inittab
-rw-rw-r--   1 root     sys            0 Oct  3 10:58 /etc/dumpdates
-rw-rw-r--   1 root     sys          540 Oct  3 11:24 /etc/vfstab
-rw-rw-rw-   1 root     other        574 Mar  1 21:01 /etc/group
-rw-rw-rw-   1 root     root           0 Feb 27 12:52 /etc/.pwd.lock
drwxrwxr-x   9 lp       lp           512 Oct  3 11:29 /etc/lp
drwxrwxr-x   2 lp       lp           512 Oct  3 11:04 /etc/lp/alerts
drwxrwxr-x   2 lp       lp           512 Oct  3 11:16 /etc/lp/classes
-rw-rw-r--   1 lp       lp           200 May  2  1996 /etc/lp/fd/catv.fd
-rw-rw-r--   1 lp       lp           243 May  2  1996 /etc/lp/fd/download.fd
-rw-rw-r--   1 lp       lp           514 May  2  1996 /etc/lp/fd/dpost.fd
-rw-rw-r--   1 lp       lp           498 May  2  1996 /etc/lp/fd/postdaisy.fd
-rw-rw-r--   1 lp       lp           516 May  2  1996 /etc/lp/fd/postdmd.fd
-rw-rw-r--   1 lp       lp           236 May  2  1996 /etc/lp/fd/postio.fd
-rw-rw-r--   1 lp       lp           275 May  2  1996 /etc/lp/fd/postior.fd
-rw-rw-r--   1 lp       lp           646 May  2  1996 /etc/lp/fd/postmd.fd
-rw-rw-r--   1 lp       lp           218 May  2  1996 /etc/lp/fd/postpages.fd
-rw-rw-r--   1 lp       lp           330 May  2  1996 /etc/lp/fd/postplot.fd
-rw-rw-r--   1 lp       lp           531 May  2  1996 /etc/lp/fd/postprint.fd
-rw-rw-r--   1 lp       lp           235 May  2  1996 /etc/lp/fd/postreverse.fd
-rw-rw-r--   1 lp       lp           532 May  2  1996 /etc/lp/fd/posttek.fd
-rw-rw-r--   1 lp       lp           270 May  2  1996 /etc/lp/fd/pr.fd
drwxrwxr-x   2 lp       lp           512 Oct  3 11:16 /etc/lp/forms
drwxrwxr-x   2 lp       lp           512 Oct  3 11:16 /etc/lp/interfaces
lrwxrwxrwx   1 root     root          17 Oct  3 11:16 /etc/lp/logs -> ../../var/lp/logs
drwxrwxr-x   2 lp       lp           512 Oct  3 11:16 /etc/lp/printers
drwxrwxr-x   2 lp       lp           512 Oct  3 11:16 /etc/lp/pwheels
-rw-rw-r--   1 lp       lp          2141 Oct  3 11:16 /etc/lp/Systems
```

```
lrwxrwxrwx   1 root     root          17 Oct  3 11:29 /etc/lp/model ->
/usr/lib/lp/model
drwxrwxr-x   2 adm      adm          512 Oct  3 11:13 /etc/acct
-rw-rw-r--   1 bin      bin          289 Oct  3 11:13 /etc/acct/holidays
lrwxrwxrwx   1 root     root          33 Oct  3 11:14 /etc/uucp/remote.unknown ->
../../usr/lib/uucp/remote.unknown
lrwxrwxrwx   1 root     root          18 Oct  3 11:17 /etc/chroot -> ../usr/sbin/chroot
lrwxrwxrwx   1 root     root          17 Oct  3 11:17 /etc/fuser -> ../usr/sbin/fuser
lrwxrwxrwx   1 root     root          16 Oct  3 11:17 /etc/link -> ../usr/sbin/link
lrwxrwxrwx   1 root     root          17 Oct  3 11:17 /etc/mvdir -> ../usr/sbin/mvdir
lrwxrwxrwx   1 root     root          16 Oct  3 11:17 /etc/pwck -> ../usr/sbin/pwck
lrwxrwxrwx   1 root     root          24 Oct  3 11:17 /etc/termcap ->
../usr/share/lib/termcap
lrwxrwxrwx   1 root     root          18 Oct  3 11:17 /etc/unlink -> ../usr/sbin/unlink
-rw-rw-r--   1 root     other        213 Oct  3 11:28 /etc/.sysIDtool.state
-rw-rw-rw-   1 root     other        131 Jan 26 21:48 /etc/resolv.conf
```

En la salida el diredtorio /***etc*** se encuentra en todos, en el paso dos (2), los scripts están bien configurados

Objetivo 3.9:	
Identificar los archivos ***suid*** y ***sgid*** en el sistema y verificar su necesidad.	
Paso 1	Como root, Ejecute el siguiente comando: `/bin/find / -type f \( -perm -4000 -o -perm -2000 \) \-exec ls -ldb {} \;`
Resultados Esperados: Verificar todos los programas listados, estos deben tener activado el bit suid y sgid.	
Resultados obtenidos	

La salida es la siguiente

```
-rwxr-sr-x   1 root     root      729584 Mar 26  1996 /usr/openwin/bin/Xsun
-rwsrwxr-x   1 root     bin        94452 Mar 26  1996 /usr/openwin/bin/xlock
-rwxr-sr-x   1 root     sys        13592 Mar 19  1996 /usr/openwin/bin/wsinfo
-r-sr-sr-x   1 root     bin        12612 Mar 19  1996 /usr/openwin/bin/ff.core
-r-xr-sr-x   1 root     mail      451832 Mar 18  1996 /usr/openwin/bin/mailtool
-rwsr-sr-x   1 root     bin        23236 Feb  7  1996
/usr/openwin/bin/kcms_configure
-rwsr-sr-x   1 root     bin        75912 Feb  7  1996
/usr/openwin/bin/kcms_calibrate
-rwxrwsr-x   1 root     sys         9284 Mar 26  1996 /usr/openwin/bin/xload
-rwsr-xr-x   1 root     bin        42620 Mar 22  1996 /usr/openwin/lib/mkcookie
-r-xr-sr-x   1 bin      sys        9512 May  2  1996
/usr/platform/sun4m/sbin/eeprom
-rwsr-xr-x   1 root     sys        33260 May  2  1996 /usr/bin/at
-rwsr-xr-x   1 root     sys        12080 May  2  1996 /usr/bin/atq
-rwsr-xr-x   1 root     sys        10764 May  2  1996 /usr/bin/atrm
-r-sr-xr-x   1 root     sys        19612 May  2  1996 /usr/bin/chkey
-r-sr-xr-x   1 root     bin        14604 May  2  1996 /usr/bin/crontab
-r-sr-xr-x   1 root     bin         9676 May  2  1996 /usr/bin/eject
-r-sr-xr-x   1 root     bin        26312 May  2  1996 /usr/bin/fdformat
-r-sr-xr-x   1 root     bin        28800 May  2  1996 /usr/bin/login
-r-x--s--x   1 bin      mail       66196 May  2  1996 /usr/bin/mail
-r-x--s--x   1 bin      mail      133448 May  2  1996 /usr/bin/mailx
-r-xr-sr-x   1 bin      sys        31484 May  2  1996 /usr/bin/netstat
-rwsr-xr-x   1 root     sys         9860 May  2  1996 /usr/bin/newgrp
-r-xr-sr-x   1 bin      sys        15872 May  2  1996 /usr/bin/nfsstat
-r-sr-sr-x   3 root     sys        15688 May  2  1996 /usr/bin/passwd
-r-sr-xr-x   1 root     sys        23740 May  2  1996 /usr/bin/ps
-r-sr-xr-x   1 root     bin        18632 May  2  1996 /usr/bin/rcp
```

```
-r-sr-xr-x   1 root     bin        64748 May  2  1996 /usr/bin/rdist
-r-sr-xr-x   1 root     bin        14636 May  2  1996 /usr/bin/rlogin
-r-sr-xr-x   1 root     bin         7940 May  2  1996 /usr/bin/rsh
-r-sr-xr-x   1 root     sys        15820 May  2  1996 /usr/bin/su
-rws--x--x   1 uucp     bin        56500 May  2  1996 /usr/bin/tip
-r-sr-xr-x   2 root     bin        11192 May  2  1996 /usr/bin/uptime
-r-xr-sr-x   1 bin      tty         9836 May  2  1996 /usr/bin/write
-r-sr-xr-x   2 root     bin        11192 May  2  1996 /usr/bin/w
-r-sr-sr-x   3 root     sys        15688 May  2  1996 /usr/bin/yppasswd
-r-sr-xr-x   1 root     bin         4888 May  2  1996 /usr/bin/volcheck
-r-s--x--x   1 root     sys       329360 Apr 25  1996 /usr/bin/admintool
-r-xr-sr-x   1 bin      bin        88256 Apr 25  1996 /usr/bin/solstice
---s--x--x   1 root     uucp       70664 May  2  1996 /usr/bin/ct
---s--x--x   1 uucp     uucp       84524 May  2  1996 /usr/bin/cu
---s--x--x   1 uucp     uucp       66376 May  2  1996 /usr/bin/uucp
---s--x--x   1 uucp     uucp       21428 May  2  1996 /usr/bin/uuglist
---s--x--x   1 uucp     uucp       17772 May  2  1996 /usr/bin/uuname
---s--x--x   1 uucp     uucp       62096 May  2  1996 /usr/bin/uustat
---s--x--x   1 uucp     uucp       70268 May  2  1996 /usr/bin/uux
-r-xr-sr-x   1 bin      sys         9544 May  2  1996 /usr/bin/ipcs
-r-sr-sr-x   3 root     sys        15688 May  2  1996 /usr/bin/nispasswd
-r-sr-xr-x   1 root     bin        12500 May  2  1996 /usr/lib/fs/ufs/quota
-r-sr-sr-x   1 root     tty       156856 May  2  1996 /usr/lib/fs/ufs/ufsdump
-r-sr-xr-x   1 root     bin       641288 May  2  1996 /usr/lib/fs/ufs/ufsrestore
-r-sr-xr-x   1 root     bin        21564 May  2  1996 /usr/lib/exrecover
---s--x--x   1 root     bin         3120 May  2  1996 /usr/lib/pt_chmod
-r-sr-x--x   1 root     bin       235272 May  2  1996 /usr/lib/sendmail
-r-sr-xr-x   1 root     bin         6984 May  2  1996 /usr/lib/utmp_update
-rwsr-xr-x   1 root     adm         4008 May  2  1996 /usr/lib/acct/accton
---s--x--x   1 uucp     uucp        4856 May  2  1996
```

```
/usr/lib/uucp/remote.unknown
---s--x--x   1 uucp     uucp      171432 May  2  1996 /usr/lib/uucp/uucico
---s--x--x   1 uucp     uucp       32288 May  2  1996 /usr/lib/uucp/uusched
---s--x--x   1 uucp     uucp       83152 May  2  1996 /usr/lib/uucp/uuxqt
-rwsr-xr-x   3 root     bin        14080 May  2  1996 /usr/sbin/allocate
-r-xr-sr-x   1 root     bin         6948 May  2  1996 /usr/sbin/arp
-r-xr-sr-x   1 bin      sys         6284 May  2  1996 /usr/sbin/fusage
-rwsr-xr-x   1 root     bin         8680 May  2  1996 /usr/sbin/mkdevalloc
-rwsr-xr-x   1 root     bin         9192 May  2  1996 /usr/sbin/mkdevmaps
-r-sr-xr-x   1 root     bin        18172 May  2  1996 /usr/sbin/ping
-r-xr-sr-x   1 root     sys        16312 May  2  1996 /usr/sbin/prtconf
-rwsr-xr-x   1 root     sys        21600 May  2  1996 /usr/sbin/sacadm
-r-xr-sr-x   1 bin      sys         7192 May  2  1996 /usr/sbin/swap
-r-xr-sr-x   1 root     sys        21572 May  2  1996 /usr/sbin/sysdef
-r-xr-sr-x   1 bin      tty         9364 May  2  1996 /usr/sbin/wall
-r-sr-xr-x   1 root     bin        12512 May  2  1996 /usr/sbin/whodo
-rwsr-xr-x   3 root     bin        14080 May  2  1996 /usr/sbin/deallocate
-rwsr-xr-x   3 root     bin        14080 May  2  1996 /usr/sbin/list_devices
-r-xr-sr-x   1 bin      sys         5492 May  2  1996 /usr/sbin/dmesg
-r-sr-xr-x   1 root     bin       550316 May  2  1996 /usr/sbin/static/rcp
-r-sr-xr-x   1 root     bin        32280 Mar 26  1996 /usr/sbin/ffbconfig
-r-sr-xr-x   1 root     bin         6584 May  2  1996 /usr/proc/bin/ptree
-r-sr-xr-x   1 root     bin         4852 May  2  1996 /usr/proc/bin/pwait
-rwsr-xr-x   1 root     sys        23392 May  2  1996 /usr/ucb/ps
-rwsr-sr-x   1 bin      bin         8448 May  2  1996 /usr/vmsys/bin/chkperm
-r-sr-xr-x   1 lp       lp           203 Apr 25  1996 /etc/lp/alerts/printer
-r-sr-xr-x   1 root     sys       339780 May  2  1996 /sbin/su
```

Todos los archivos listados presentan el bit suid o sgid activado.

Objetivo 3.10:   Identificar los archivos ***suid y sgid*** en el sistema y verificar su necesidad.
Paso 1 — Como root Ejecute el siguiente comando: `/usr/bin/find / \( -type c -o -type b \) -exec ls -ldb {} \; | grep -v "/dev/"`
Resultados Esperados: No debe haber salida en este comando, indicando que hay dispositivos fuera el directorio /**dev**
Comentarios: Cualquier dispositivo fuera del directorio `/dev` directory es sospechoso   El comando ***`/usr/sbin/ncheck -s`*** puede ser usado para mostrar los archivos especiales y archivos con el ***bit suid*** activado.
Paso 2 — Como root, Ejecute el siguiente comando: `/usr/bin/find /dev ! \( -type l -o -type c -o -type b \) \-exec ls -ldb {} \;`
Resultados Esperados: Todos los archivos en el directorio ***/dev*** son archivos especiales
Paso 3 — Como root, Ejecute el siguiente comando: /usr/bin/find / \(-type c -o -type b \) ! -user root -exec ls -ldb {} \;
Resultados Esperados: No hay dispositivos especiales que sean propiedad del root que no deban ser propiedad del root.
Resultados obtenidos   Paso 1   Se obtuvo una grán lista y todos los dispositivos están en el directorio /devices   Ej: crw------- 1 root sys 11,106 May 3 1996 /devices/pseudo/clone@0:be   crw------- 1 root sys 11, 7 May 3 1996 /devices/pseudo/clone@0:hme

```
crw-------   1 root     sys       11, 63 May  3  1996 /devices/pseudo/clone@0:ie
crw-rw----   1 root     sys        11,  3 May  3  1996 /devices/pseudo/clone@0:ip
crw-------   1 root     sys       11, 40 May  3  1996 /devices/pseudo/clone@0:le
crw-------   1 root     sys       11,104 May  3  1996 /devices/pseudo/clone@0:qe
crw-rw-rw-   1 root     sys       11, 41 May  3  1996
/devices/pseudo/clone@0:udp
crw--w----   1 root     tty        0,  0 Mar  3 19:39 /devices/pseudo/cn@0:console
crw--w----   1 root     tty        0,  0 Mar  2 21:26 /devices/pseudo/cn@0:syscon
crw--w----   1 root     tty        0,  0 May  3  1996 /devices/pseudo/cn@0:systty
crw-rw-rw-   1 root     sys       72,  0 May  3  1996
/devices/pseudo/ksyms@0:ksyms
crw-rw-rw-   1 root     sys       21,  0 May  3  1996
/devices/pseudo/log@0:conslog
crw-r-----   1 root     sys       21,  5 May  3  1996 /devices/pseudo/log@0:log
crw-r-----   1 root     sys       13,  1 May  3  1996 /devices/pseudo/mm@0:kmem
crw-r-----   1 root     sys       13,  0 May  3  1996 /devices/pseudo/mm@0:mem
crw-rw-rw-   1 root     sys       13,  2 Mar  3 16:48 /devices/pseudo/mm@0:null
crw-rw-rw-   1 root     sys       13, 12 May  3  1996
/devices/pseudo/mm@0:zero
crw-r-----   1 root     sys       38,  0 May  3  1996
/devices/pseudo/openeepr@0:openprom
crw-------   1 root     sys       12,  1 May  3  1996 /devices/pseudo/sad@0:admin
crw-rw-rw-   1 root     sys       12,  0 May  3  1996 /devices/pseudo/sad@0:user
crw-rw-rw-   1 root     tty       22,  0 May  3  1996 /devices/pseudo/sy@0:tty
crw-------   1 root     sys       15,  0 May  3  1996 /devices/pseudo/wc@0:wscons
crw-------   1 root     sys       11,  5 Oct  2 23:35 /devices/pseudo/clone@0:icmp
crw-rw-rw-   1 root     sys       11, 44 Oct  2 23:35 /devices/pseudo/clone@0:arp
crw-------   1 root     sys       11, 10 Oct  2 23:35 /devices/pseudo/clone@0:sp
crw-rw-rw-   1 root     sys       11, 23 Oct  2 23:35
/devices/pseudo/clone@0:ptmx
```

```
crw-rw-rw-   1 root     sys       11, 71 Oct  2 23:35
/devices/pseudo/clone@0:zsh
crw-rw-rw-   1 root     sys       11,107 Oct  2 23:35
/devices/pseudo/clone@0:llc1

         Paso 2
drwxrwxr-x  16 root     sys         3584 Mar  2 21:27 /dev
drwxrwxr-x   2 root     sys          512 May  3  1996 /dev/sad
drwxrwxr-x   2 root     sys          512 Oct  2 23:35 /dev/dsk
dr-xr-xr-x   2 root     root         800 Mar  3 20:43 /dev/fd
drwxrwxr-x   2 root     sys         1024 Oct  3 11:26 /dev/pts
drwxrwxr-x   2 root     sys          512 Oct  2 23:35 /dev/rdsk
drwxrwxr-x   2 root     sys          512 May  3  1996 /dev/rmt
drwxrwxr-x   2 root     sys          512 May  3  1996 /dev/swap
drwxrwxr-x   2 root     root         512 Oct  3 11:26 /dev/term
drwxr-xr-x   2 root     root         512 Oct  2 23:35 /dev/fbs
drwxr-xr-x   2 root     root         512 Oct  2 23:35 /dev/printers
drwxr-xr-x   4 root     root         512 Oct  2 23:35 /dev/md
drwxr-xr-x   2 root     root         512 Oct  2 23:35 /dev/md/dsk
drwxr-xr-x   2 root     root         512 Oct  2 23:35 /dev/md/rdsk
drwxr-xr-x   2 root     root         512 Oct  3 11:26 /dev/cua
drwxr-xr-x   2 root     root         512 Oct  3 11:26 /dev/isdn
drwxr-xr-x   2 root     root         512 Oct  3 11:26 /dev/sound
Todos los archivos están en /dev

Paso 3
crw--w----   1 aramirez tty       24,  4 Mar  3 20:46 /devices/pseudo/pts@0:4
crw--w----   1 hagualin tty       24,  5 Mar  3 20:43 /devices/pseudo/pts@0:5
crw--w----   1 hagualin tty       24,  6 Mar  3 20:48 /devices/pseudo/pts@0:6
crw--w----   1 hagualin tty       24,  7 Mar  3 20:43 /devices/pseudo/pts@0:7
```

```
crw--w----   1 hagualim tty       24, 13 Nov 20 19:39 /devices/pseudo/pts@0:13
crw-------   1 uucp     uucp      29,131072 Oct  2 23:35
/devices/obio/zs@0,100000:a,cu
crw-------   1 uucp     uucp      29,131073 Oct  2 23:35
/devices/obio/zs@0,100000:b,cu
```

5 AUTENTICACION

Objetivo5.0:

Verificar que no halla cuentas en el sistema que no hayan sido usadas dentro de un tiempo razonable.

Paso 1	Teclee y ejecute el siguiente script para determinar qué usuarios no se han conectado en el último mes: `#!/bin/sh` `date` `uname –a` `PATH=/bin:/usr/bin;export PATH` `umask 077` `THIS_MONTH=`date | /usr/bin/awk '{print $2}'` `/usr/bin/last | /usr/bin/grep $THIS_MONTH |` `/usr/bin/awk '{print $1}' \` `| /usr/bin/sort –u > users1$$` `cat /etc/passwd \| /usr/bin/awk –F: '{print $1}' \|` `/usr/bin/sort –u \` `> users2$$` `/usr/bin/comm –13 users[12]$$` `/usr/bin/rm –f users[12]$$`

Resultados Esperados:

Ninguna cuenta con nombre de usuario debe mostrarse. Si alguna cuenta es retornada etá debe considerarse como cuenta durmiente y debe ser

desactivada o borrada.
Resultados obtenidos aepastran bin cripseg daemon listen lp noaccess nobody nobody4 nuucp smtp sys test uucp Se listaron 14 cuentas que no deben aparecer

Objetivo 5.1:	
Verificar que no haya GIDs duplicados.	
Paso 1	Revise el archivo `/etc/group`.
Resultados Esperados:	
No debe haber GiDs duplicados	
Resultados obtenidos	
No se presentaron GIDs duplicados	

Objetivo 5.2:

Verificar que los Id de los usuarios sumistrados por la instalación no tengan password por defecto.

Muchas cuentas vienen con el sistema operativo UNIX. (Esas cuentas están normalmente al comienzo del archivo de password ***(/etc/passwd)*** y tienen ***nombres como: bin, uucp, y news.)*** Estas cuentas deben ser desactivadas o cambiarles el password.

Paso 1	Inspeccione el archivo `/etc/passwd` para ver los password por defecto.

Resultados Esperados:

Las cuentas suministradas con la instalación deben ser desativadas o asignarles un password.

Resultados obtenidos

root:x:0:1:Super-User:/:/sbin/sh

daemon:x:1:1::/:

bin:x:2:2::/usr/bin:

sys:x:3:3::/:

adm:x:4:4:Admin:/var/adm:

lp:x:71:8:Line Printer Admin:/usr/spool/lp:

smtp:x:0:0:Mail Daemon User:/:

uucp:x:5:5:uucp Admin:/usr/lib/uucp:

nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico

listen:x:37:4:Network Admin:/usr/net/nls:

nobody:x:60001:60001:Nobody:/:

noaccess:x:60002:60002:No Access User:/:

nobody4:x:65534:65534:SunOS 4.x Nobody:/:

jmartine:x:1002:1::/export/home/jmartine:/bin/sh

hagualim:x:1001:1::/export/home/hagualim:/bin/csh

aramirez:x:1003:1::/export/home/aramirez:/bin/sh

jarcila:x:1004:1::/export/home/jarcila:/bin/sh

aepastran:x:1005:1::/export/home/epastran:/bin/sh
cripseg:x:1006:10::/export/home/cripseg:/bin/sh
backup:x:1007:10:Cuenta chequeao:/export/home/backup:/bin/sh
hagualin:x:1008:10::/hagualin:/bin/sh
test:x:1009:10::/export/test:/bin/sh
Existen algunas cuentas que son instaladas con el SO como uucp y bin

Objetivo 5.3:

Verificar que no haya ***GIDs*** duplicados.

Paso 1	Revise el archivo /**etc/passwd.**

Resultados Esperados:

No debe haber ***UIDs*** duplicados. Si los hay, estas cuentas deben ser desactivadas

Resultados obtenidos

No existen cuentas con ***UID*** duplicados

Objetivo 5.4:

Verifique que no haya cuentas ***guest*** o ***grupo*** en el sistema.
Las cuentas guest presentan un hueco de seguridad. Por su naturaleza, esas cuentas son poco usadas, algunas son usadas siempre por personas que deben tener acceso a la maquina por periodos cortos de tiempo. La forma más segura para mantener las cuentas guest es instalarlas según su necesidad y borrarlas tan pronto como los usuarios deje de utilizarlas.

A las cuentas Guest nunca se les debe dar password sencillos como ***"guest o "visitor"*** y nunca deben ser permitidas en el archivo de password cuando no se estén usando.	
Paso 1	Vea el archivo `/etc/passwd` para determinar si hay una cuenta gues. Si la hay, intente con los password "`visitor`" o "`guest`".
Resultados Esperados: La cuenta guest no debe existir.	
Resultados obtenidos No existen cuentas como ***"guest"o "visitor"***	

Objetivo 5.5: verificar que los archivos passwor y shadow tengan los permisos correctos	

Paso 1	Haga lo siguiente:
	`ls –l /etc/passwd`
	`ls –l /etc/shadow`

Resultados Esperados: Los resultados deben parecerse a estos: -r--r--r-- 1 root sys 576 Apr 27 15:31 /etc/passwd -r-------- 1 root sys 328 Apr 27 15:31 /etc/shadow Los archivos password deben ser propiedad del root. El archivo ***/etc/passwd*** debe estar en el modo ***0444***, y el archivo ***/etc/shadow/*** en el modo ***0400***	
Resultados Obtenidos La salida fue la esperada	

MANUAL DE SEGURIDAD SOLARIS 2.6

El presente documento es un manual técnico, el cual contiene básicamente procedimientos para chequear la seguridad del sistema operativo ***Solaris 2.6***. En este manual se muestran las tareas que debe realizar un administrador de sistemas y por cada una se hace una recomendación acerca de cómo llevarlas a cabo, de las consecuencias en caso de encontrar errores y un análisis de la seguridad. Estas tareas deben realizarse mensualmente para así alcanzar un grado elevado de seguridad. Por último describe cómo ejecutar algunas herramientas de seguridad que sirven como apoyo y ayudan a mejorar la seguridad de la red.

Como se puede ver este manual puede ser incorporado dentro de las actividades del administrador, lo cual garantizaría un óptimo funcionamiento de la red y de los servicios que ésta ofrece.

1 CONTROL DE ACCESO

Esta sección describe como controlar el acceso a la red y al filesystem.

1.1 CONTROL DE ACCESO A LA RED

Objetivo 1.1.1 Asegurar el ***password***.	
Paso 1	Como ***root***, teclee `passwd -n 10 -x 7 username`
Resultados Esperados: Como (***-n 10***) es mayor que (***-x 7***) el ***password*** es bloqueado y no puede ser cambiado. El usuario se puede conectar a la máquina	

Objetivo 1.1.2 Mostrar el estado de los ***password***.	
Paso 1	***Como*** root***, teclee*** `passwd -s username`

Resultados Esperados:

Debe mostrar la siguiente información del usuario

Login

Estado del password:

- ***NP***: No tiene ***password***
- ***LK***: El ***login*** está bloqueado
- ***PS***: Cualquier cosa
- La última fecha en que fue cambiado el ***password***
- Número mínimo de días después del último cambio del ***password*** antes de que el usuario pueda cambiarlo
- Máximo número de días entre cambio de números de ***password***
- Número de días de advertencias antes de que el ***password*** deba ser cambiado

Comentarios:

Si se utiliza la opción ***–a*** la información anterior servirá para todos los usuarios.

Objetivo 1.1.3

Cómo forzar a un usuario a usar un nuevo ***password***.

Paso 1	Como ***root***, teclee `passwd -f username`

Resultados Esperados:

En una próxima conexión el usuario introduce un nuevo ***password***.

Objetivo 1.148	
Activar y desactivar un ***password***.	
Paso 1	Como ***root***, teclee `passwd -n 1 -x 90 -w 7 hagualim`
Resultados Esperados:	
Este ejemplo configura la expiración del ***password*** para el usuario ***hagualim***, el cual lo debe cambiar cada 90 días, y debe esperar un día después para poder cambiarlo otra vez. Siete días antes de que el ***password*** expire la máquina le envía un mensaje al usuario cuando éste se conecta.	
Comentario: `# Passwd -n min -x max -w warn username` Este comando configura el mínimo número de días entre cambio de ***password*** (***-n min***) y el número de días en que el ***password*** es válido (***-xmax***) y el número de días antes de que el ***password*** expire (***-w warn***).	

Objetivo 1.1.5	
Mostrar la información de un usuario conectado al sistema	
Paso 1	Ejecute el siguiente comando: `logins -x -l username`
Resultados Esperados	
La información mostrada incluye el ***login***, ***UID***, ***GID***, ***nombre de usuario***, ***directorio home***, ***shell*** y la ***fecha de expiración*** de la cuenta.	
Paso 2	Ejecute el siguiente comando: `logins -p`

Resultados Esperados   Muestra las cuentas que no tienen ***password***.
Comentarios: Cualquiera cuenta mostrada debe desactivarse u obligar al usuario a usar un ***password***, cualquiera puede tener acceso al sistema si utiliza la cuenta mostrada.

Objetivo 1.1.6 Verificar si el archivo de ***passwords*** está configurado correctamente	
Paso 1	Como ***root***, teclee `pwck`
Resultados Esperados: El comando ***pwck*** chequea el archivo ***/etc/passwd*** para asegurarse de que cada línea está correctamente formateada. Analice los resultados arrojados por este comando.	

Objetivo 1.1.7 Revisión del registro del sistema. Se puede revisar si existe algún usuario no autorizado que esté conectado como ***root*** o que trate de capturar el ***password***.	
Paso 1	Como ***root***, teclee `more /var/adm/sulog`
Resultados Esperados: En este archivo puede verse desde la fecha que están conectados los usuarios y se puede verificar si cuentas durmientes están siendo accesadas.	

Objetivo 1.1.8	
Almacenar todos los ***passwords*** de usuario en el archivo ***/etc/shadow***. **UNIX** encripta todos los ***passwords*** de usuarios usando un algoritmo basado en ***DES***, el cual encripta el ***password*** en una cadena de 11 caracteres. Como este archivo puede ser accesado por todos los usuarios un intruso podría capturar ***passwords*** encriptados y mediante un ataque podría capturar un par ***login-password***.	
Paso 1	Como ***root***, teclee `pwconv`
Resultados Esperados:	
Para prevenir que usuarios normales lean todos los ***passwords*** almacenados se usa el comando ***pwconv*** para mover los ***passwords*** encriptados del archivo /***etc/passwd*** al archivo ***/etc/shadow***, el cual debe estar protegido contra lectura para usuarios distintos al superusuario.	

Objetivo 1.1.9	
Asegurarse que no existen cuentas con privilegios de superusuario. Las cuentas que tienen un ***UID*** de cero(0), automáticamente tienen los privilegios de superusuario. En caso de que no se requiera es necesario eliminarlas colocando un ***'*'*** en el campo de password del archivo ***/etc/passwd***.	

Paso 1	Teclee los comandos:
	`more /etc/passwd | grep:0:`
	`more /etc/passwd | grep:1:`

Resultados Esperados:

Algunos sistemas Unix como ***SunOS 4.x*** tienen un grupo de usuarios llamado ***root*** o ***wheel***. A cualquier usuario de este grupo se le habilita el comando ***su*** para dar privilegios de superusuario en el caso de que conozcan el ***password*** del superusuario. Es conveniente limitar la mayor cantidad de usuarios posibles que puedan usar ***su*** al root. La información de los grupos se encuentra en el archivo ***/etc/group***.

Objetivo 1.1.10

Preguntar a usuarios para reportar conexiones sospechosas. Muchos sistemas guardan las últimas conexiones ejecutadas por los usuarios en el archivo ***/var/adm/lastlog*** o ***/usr/adm/lastlog***.

Paso 1	Como ***root***, teclee los comandos:
	`more /usr/adm/lastlog`
	O
	`more /var/adm/lastlog`

Resultados Esperados:

Aparecerán las últimas conexiones de los usuarios. Se puede preguntar a los usuarios para ver si reconocen conexiones efectuadas a sus cuentas.

Objetivo 1.1.11	
Calcular sumas de chequeo a todos los archivos. El comando ***sum*** calcula suma de chequeos a los archivos.	
Paso 1	Teclee el siguienete comando: `sum <nombre del archivo>`
Resultados Esperados: El tamaño del archivo debe permanecer igual al tamaño de la última modificación.	
Comentarios: Cuando se instala el sistema es posible calcular sumas de chequeo a archivos que comúnmente no cambian y recalcular estas sumas periódicamente para comparar que los archivos no han sido modificados.	

Objetivo 1.1.12:	
Desactive las conexiones del ***root*** a la red, pero active la consola en la línea ***/etc/default/login***.	
Paso 1	Para desactivar el uso de ***ftp*** al ***root***, adicione "***root***" a ***/etc/ftpusers***.

Objetivo 1.1.13:	
Elimine o colóquele comentarios a las cuentas innecesarias, incluyendo "***sys***", "***uucp***", "***nuucp***" y "***listen***".	
Paso 1	La forma más clara para dar de baja una cuenta es colocar "***NP***" en el campo ***password*** en el archivo ***/etc/shadow***

Objetivo 1.1.14:	
Desactive el acceso por ***rlogin***, ***rsh.***	
Paso 1	Eliminarlos en ***/etc/host.equiv***, ***./rhost***, y todos los comandos "***r***" en ***/etc/inetd.conf.***

Objetivo 1.1.15:	
Instale ***TCP Wrappers,*** este provee autenticación de ***host*** y conexiones a los servicios iniciados por ***inetd***.	
Paso 1	Instale ***tcpd*** en el archivo binario ***/usr/sbin***. Reemplace ***/etc/inet/inetd.conf*** con lo siguiente # /etc/inet/inetd.conf # # Configuration file for inetd(1M). See inetd.conf(4). # # To re-configure the running inetd process, edit this file, then # send the inetd process a SIGHUP. # # Syntax for socket-based Internet services: # <SERVICE_NAME <SOCKET_TYPE <SERVER_PATHNAME # # Syntax for TLI-based Internet services: # # <SERVICE_NAME tli <SERVER_PATHNAME # ftp stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.ftpd telnet stream tcp nowait root /usr/sbin/tcpd /usr/sbin/in.telnetd.

Resultados Esperados:

El demonio ***/usr/sbin/tcpd*** debe ser ejecutado por ***inetd***, en vez de ser ejecutado directamente, ***TCP Wrappers*** usa dos archivos para el control de los servicios, ***/etc/host.allow*** y ***/etc/host.deny.***

El archivo ***/etc/host.allow*** debe contener los ***host*** permitidos para accesar los servicios administrativos

ALL: 10.0.20.41,10.0.20.56.

El archivo ***/etc/host.deny*** debe contener los ***host*** no permitidos para accesar los servicios administrativos.

Objetivo 1.1.16:

Asegúrese de confirmar los permisos en los archivos de acceso a ***host***.

Paso 1	Como ***root***, ejecute el siguiente comando: `chmod 600 /etc/hosts.allow /etc/hosts.deny`

1.2 CONTROL DE ACCESO AL FILESYSTEM

Esta sección describe como proteger los archivos del sistema

Objetivo 1.2.0:	
Ejecute el ***Automated Security Enhancement Tool*** (***ASET***), que chequea las configuraciones y los contenidos de los sistemas de archivos. Muchos de los programas ***suid*** y ***sgid*** solo son usados por el ***root*** o por otros usuarios que estén en su grupo. La partición ***/usr*** debe ser montada como solo lectura "***ro***" y el software debe ser movido a una partición de escritura.	

Paso 1	Como ***root***, ejecute el siguiente comando:
	`/bin/mv /usr/aset /var`
	`/bin/ln -s /var/aset /usr/aset`

Objetivo 1.2.1:	
Ejecute ***ASET*** usando el nivel más alto de seguridad y revise los reporte generados en el archivo ***/var/aset/reports***.	
Paso 1	Como ***root***, ejecute el siguiente comando: / `/var/aset/aset -l high`

2 LISTAS DE SEGURIDAD

2.1 FUNCIONES DE AUDITORÍA

<table>
<tr><td colspan="2"><u>Objetivo 2.1.0:</u>
Asegurarse de que la auditoría del subsitema esté activada y configurada de una forma segura</td></tr>
<tr><td>Paso 1</td><td>Ejecute el siguiente comando para verificar que la auditoria esté activada:
<code>/usr/sbin/auditconfig -getcond</code></td></tr>
<tr><td colspan="2">Resultados esperados:
La opción –getcond obtiene la condición de auditoría de la máquina. La respuesta es una las siguientes:
<code>Auditing - Auditing is enabled and turned on</code>
<code>no audit - Auditing is enabled but not turned on</code>
<code>disabled - Auditing is not enabled</code>
Un Mensaje de error con el siguiente formato "<code>auditconfig: error = Invalid argument(22)</code>" indica que la opción BSM no está disponible y no puede ser usada.</td></tr>
<tr><td colspan="2">Comentarios:
El modelo básico de seguridad debe ser instalado y puesto en marcha.
EL script /etc/security/bsmconv permite las características BSM de Solaris.</td></tr>
</table>

Objetivo 2.1.1: Verificar que la auditoría del ***Kernel*** no haya sido modificada inapropiadamente	
Paso 1	Mire los siguientes archivos: `/etc/security/audit_event` Compare su contenido con el archivo por defecto de los eventos de auditoría del ***Kernel*** que vienen equipados con ***Solaris***.
Reusltados esperados: Cualquier modificación en los eventos de auditoría del ***Kernel*** deben ser justificados.	
Comentarios: Las acciones del sistema que son auditables, son definidas como eventos de auditoría en el archivo ***/etc/security/audit_event***. Cada evento auditable es definido por un nombre simbólico, y número de evento, etc.	

Objetivo 2.1.2: Asegurarse de que la auditoría esté correctamente configurada y que reúna los eventos de auditoría requeridos.	
Paso 1	Ejecute el siguiente comando: `/usr/sbin/auditconfig –chkconf`
Resultados esperados: El comando no debería mostrar ninguna inconsistencia en el mapeo auditado. De lo contrario, este comando no mostraría ningún mensaje.	
Comentarios: El módulo básico de seguridad debería ser instalado y puesto en marcha. La opción ***–chkconf*** chequea la configuración de los eventos de auditoría del ***Kernel*** para clasificar el mapeo y reportar cualquier inconsistencia	

Objetivo 2.1.3:

Verificar que las clases de auditoría no hayan sido modificadas

Paso 1	Ejecute el siguiente comando: `more /etc/security/audit_class` Compare su contenido con el archivo por defecto de los eventos de auditoria del ***Kernel*** que vienen equipados con ***Solaris 2.2.1***

Resultados esperados:

Los archivos que coincidan, indican que la auditoría de clases no ha sido modificada inapropiadamente.

Comentarios:

```
# # User Level Class Masks
```

Developers: If you change this file you must also edit audit. h.

```
# File Format: # mask: name: description
# 0x00000000: no: invalid class
0x00000001: fr: file read 0x00000002: fw: file write
0x00000004: fa: file attribute access 0x00000008: fm: file
attribute modify
0x00000010: fc: file create 0x00000020: fd: file delete
0x00000040: cl: file close 0x00000080: pc: process
0x00000100: nt: network 0x00000200: ip: ipc
0x00000400: na: non- attribute 0x00000800: ad:
administrative
0x00001000: lo: login or logout
```

Cada evento auditado es definido como perteneciente a una clase o clases de auditoría. Por asignación de eventos a clases, un administrador puede más fácilmente tratar con un gran número de eventos. Cuando llamamos una clase, simultáneamente una dirección llama a todos lo eventos de esa clase. En todo caso un evento auditable es grabado en un registro dependiendo si el administrador selecciona una clase de auditoría que incluye el evento

específico.

Objetivo 2.1.4: Verificar que el intento de conexiones sin éxito aparezcan registradas	
Paso 1	Verifique que los siguientes archivos existan y tengan los permisos aceptables. `ls -l /var/adm/loginlog`
Resultados Esperados: El archivo ***/var/adm/loginlog*** debe existir en el ***host*** y debe tener los siguientes permisos `-rw------- 1 root sys 2073 Aug 23 13:05 /var/adm/loginlog` Si el archivo no existe nada está conectado.	

Objetivo 2.1.5: Verificar que el demonio ***syslogd*** se esté ejecutando y esté configurado de una forma segura.	
Paso 1	Ejecute el siguiente comando: `ps -eaf | grep syslog`

Resultados Esperados:
 La salida debe parecerse a la siguiente:
 `ps -eaf | grep syslog`
 `root 309 1 0 16:23:24 ? 0:00 /usr/sbin/syslogd`	

Paso 2	Ejecute el siguiente comando: `ls -l /etc/syslog.conf`
Resultados Esperados: La configuración es: no es de todos, ni del grupo y el propietario debe ser el ***root***	
Paso 3	Ejecute el siguiente comando:

	`/bin/more /etc/syslog.conf`

Resultados Esperados:

La configuración del archivo debe tener lo siguiente:

```
mail.debug              /var/log/syslog
*.info;mail.none        /var/adm/messages
*.alert                 /dev/console
*.alert                 root
*.emerg                 *
```

Esto conectará las entradas de correo al archivo ***/var/log/syslog*** y lo demás al archivo ***/var/adm/messages***

Objetivo 2.1.6:

Identificar cualquier usuario para los cuales la auditoría haya sido desactivada

Paso 1	Teclee el siguiente comando: `/bin/more /etc/security/audit_user` Identifique cualquier usuario contenido en el archivo ***/etc/passwd***, que no esté en el archivo ***/etc/security/audit_user***.

Resultados Esperados:

El archivo tiene una línea al comienzo con el nombre del usuario, para cada usuario autorizado. También las clases no auditadas en el archivo ***audit_control***, están listadas después de la segunda coma, para cualquier línea de usuario en este archivo. El nivel de auditoría del sistema se aplica a todos los usuarios, a menos que el usuario tenga una entrada en el archivo ***/etc/security/audit_users***. El nivel de auditoría del usuario sobrepasa al nivel de auitoría del sistema. Los campos en el archivo están separados por columnas y están definidos como sigue:

`username: always audit flags: never audit flags`
Comentarios: Todos los usuarios deben estar sujetos a auditoría

Objetivo 2.1.7: Verificar los parámetros requeridos identificados para cada evento de auditoría grabado, incluyendo fecha y hora del evento, identificación del usuario, tipo de evento, éxito o fracaso del mismo.	
Paso 1	Como un usuario sin privilegio intente vearel archivo ***/etc/security/audit_control*** usando el comando: `/usr/bin/more /etc/security/audit_control`
Resultados Esperados: Los permisos para ver el archivo son denegados para un usuario sin privilegios.	
Paso 2	Como ***root*** intenete ver el archivo ***/etc/security/audit_control*** usando el comando: `/usr/bin/more /etc/security/audit_control`
Resultados Esperados: La configuración de eventos auditados en el sistema, muestran como mínimo los siguientes eventos que son auditados: **(ad)** Operación administrativa normal. **(lo)** Conexión y desconexión. **(fc)** Creación del objeto **(fd)** Eliminación de objetos **(fw)** Fallas para escribir a un archivo	
Comentarios: El archivo ***audit_control***, muestra los archivos de auditoría del sistema y las configuraciones del demonio de auditoría (***auditd***): Cada línea consiste de un	

título y una cadena, separada por una coma. El administrador del sistema define cuatro grupos de líneas en el archivo ***audit_control***:

5. La línea ***audit flags*** (***flags***), contiene la bandera de auditoría que define qué clases de eventos son auditados por los usuarios en la máquina.
6. La línea ***non-attributable flags*** (***naflags***), contiene la bandera de auditoría que define, qué clases de eventos son auditados cuando una acción específica no puede ser atribuida a un usuario específico.
7. La línea ***audit threshold*** (***minfree***), define el nivel mínimo de espacio libre para todos los ***filesystems*** auditables, el procentaje del ***minfree*** debe ser mayor o igual a cero, por defecto es 20%.
8. La línea ***directory definition*** (***dir***), define cada ***filesystem*** auditado y directorios en la máquina que se usarán para almacenar estos grupos de archivos auditados.

El archivo ***audit_user*** almacena por usuario los datos auditados por preselección. Cada entrada en este archivo tiene la forma:

```
username: always-audit-flags: never-audit-flags
```

Objetivo 2.1.8:	
Verificar que el script ***audit_warn*** no haya sido modificado inapropiadamente.	
Paso 1	Digite el siguiente comando: `/usr/bin/more /etc/security/audit_warn`
Resultados Esperados: El archivo ***/etc/security/audit_warn*** no ha sido cambiado por el que viene por defecto con ***Solaris 2.5.1.***	
Comentarios: Cuando el demonio ***auditd*** encuentra una condición inusual mientras escribe los registros de auditoría, este invoca el ***script /etc/security/audit_warn***. Este	

script puede ser personalizado por sitios individuales para advertir las condiciones que quizá requerirán intervención manual o que se maneje automáticamente. Para todas las condiciones de error, ***audit_warn*** escribe un mensaje en la consola y envía uno al alias "***audit_warn***".

Objetivo 2.1.9:

Verificar que el alias "***audit_warn***" haya sido configurado correctamente.

Paso 1	Digite el siguiente comando: `/usr/bin/more /etc/aliases`

Resultados Esperados:

Una entrada debe aparecer para el alias "***audit_warn***", y el alias debe ser el nombre de una cuenta actual.

```
#ident "@(#) aliases 1.13 92/ 07/ 14 SMI" /* SVr4.0 1.1 */
## # Aliases can have any mix of upper
and lower case on the left- hand side, # but the right- hand
side should be
proper case (usually lower). #
# The program "newaliases" will need to be run after
# NOTE: this file is updated for any changes to
# show through to sendmail. #
# @(#) aliases 1.8 86/ 07/ 16 SMI ##

# Following alias is required by the mail protocol, RFC 822.
# Set it to the address of a HUMAN who deals with this system's
mail
problems. Postmaster: root
audit_ warn: sysadmin, root
# Alias for mailer daemon; returned messages from our MAILER-
DAEMON
```

should be routed to our local Postmaster.
Objetivo 2.1.10: Verificar que los datos auditados estén protegidos por el sistema de manera que el acceso a estos esté limitado solamente para los que tienen autorización de ver los datos auditados. En resumen, verificar que los datos auditados están protegidos de cambios o eliminación por usuarios en general

Paso 1	Como ***root***, determine el nombre de los archivos auditados mostrados en una línea que inicia con "***dir***:" en el archivo ***/etc/security/audit_control***. Para cada nombre de archivo listado en el archivo ***audit_control***, revise los permisos de archivo usando el siguiente comando: `ls -l`

Resultados Esperados: El propietario de todos los archivos debe ser el ***root*** y d be estar en el modo ***0600***.

2.2 DISPONIBILIDAD

Objetivo 2.2.0: Verificar que el sistema es respaldado en bases regulares.

Paso 1	Ver el archivo ***crontab*** para determinar si el sistema es respaldado automáticamente con base en un plan. Del libro de registros del sistema o del administrador del sistema, determine cuando fue hecha la última copia. Determine si las cintas de ***backup*** están correctamente etiquetadas.

Resultados Esperados: Los ***backups*** son hechos regularmente por procedimientos operacionales.

2.3 PERMISOS

<table>
<tr><td colspan="2"><u>Objetivo 2.3.0:</u>
Verificar que el cron ha sido correctamente configurado.</td></tr>
<tr><td>Paso 1</td><td>Ejecute los siguientes comandos:
<pre>/usr/bin/find /var/spool/cron/crontabs -type f -exec ls -l {} \; \
-exec /usr/bin/more {} \;
/usr/bin/find /var/spool/cron/atjobs -type f -exec ls -l {} \; \
-exec /usr/bin/more {} \;</pre></td></tr>
<tr><td colspan="2">Resultados Esperados:
Todos los usuarios de los archivos crontab son propiedad de los usuarios y grupos correctos. Todos los archivos que hacen referencia a los usuarios en el archivo crontab, o que son referenciados por archivos en el archivo crontab, no son todos ni el grupo, y las tareas del cron son las apropiadas.</td></tr>
<tr><td colspan="2">Comentarios:
Asegúrese que los archivos de tareas del cron del root no tengan fuente de otros archivos donde el propietario no sea el root, o sea de todos, o del grupo.</td></tr>
<tr><td>Paso 2</td><td>Ejecute ls –l y more en cada archivo referenciado en el archivo crontab para verificar que ninguno de estos archivos son escritos por todos (también revise los directorios en el path de los archivos de referencia).</td></tr>
<tr><td colspan="2">Resultados Esperados:
Todos los archivos a los que se hace referencia en el archivo crontab, o que son referenciado por archivos no son usados por todos o el grupo y contienen entradas válidas.</td></tr>
</table>

Paso 3	Ejecute el siguiente comando:
	`ls -l /var/cron/log`
	`/usr/bin/more /var/cron/log`

Resultados Esperados: El registro del ***cron*** no es de todos ni del grupo, y los trabajos registrados tienen que ser aprobados.	

Objetivo 2.3.1: Verificar que los ejecutables ***expreserve*** y ***exrecover*** son seguros.	
Paso 1	Ejecute el siguiente comando: `ls -l /usr/lib/expreserve /usr/lib/exrecover` Revise para ver si ***expreserve*** o ***exrecover*** tienen el ***suid*** del ***root.***
Resultados Esperados: El bit ***suid*** no debería ser colocado en ***expreserve*** o ***exrecover***. Si lo es, ejecute el siguiente comando: `chmod -s /usr/lib/expreserve /usr/lib/exrecover`	

Objetivo 2.3.2: Verificar que la ruta de búsqueda del ***root*** sea la correcta.Una ruta de búsqueda nunca debe contener el directorio actual. Especialmente en la cuanta del superusuario. Generalmente una ruta de búsqueda nunca debe incluir un directorio escribible por otros usuarios.	
Paso 1	Como ***root***, ejecute el siguiente comando: `echo $PATH` Revise la ruta de búsqueda en los archivo ***/.profile***, ***/.cshrc***, y ***/.login***.

Resultados Esperados: La ruta de búsqueda del ***root*** no incluye el directorio actual("**.**").	
Paso 2	Como ***root***, ejecute el siguiente comando: `ls -ldb `echo $PATH | sed 's/:/ /g'``
Resultados Esperados: Ninguno de los directorios en la ruta de búsqueda deben ser escribibles por todos.	

Objetivo 2.3.3: Asegurarse de que los archivos del sistema son configurados de una forma correcta y segura.	
Paso 1	Ejecute el siguiente comando: `/usr/bin/df -t | /usr/bin/more`
Resultados Esperados: Todo los ***filesystem*** deben estar particionados apropiadamente de manera que ningún ***filesystem*** se esté acercando al 100%.	

Objetivo 2.3.4: Verificar que los archivos de inicialización sean sólo escribibles por el ***root***. Todos los archivos de inicialización deben ser protegidos de tal manera que sólo los usuarios pueden escribir en ellos. Es importante que los archivos de inicialización que usa el superusuario no sean escribibles por otros.	
Paso 1	Como ***root***, ejecute los siguientes comandos desde el directorio ***home*** del ***root*** y verifique que las salidas de los archivos listados sean escribibles sólo por el ***root***: `ls -l /.login`

	`ls -l /.profile`
	
	`ls -l /etc/profile`
	`ls -l /.cshrc`
	`ls -l /.kshrc`
	`ls -l /.emacs`
	`ls -l /.exrc`
	`ls -l /.forward`
	`ls -l /.rhosts`
	`ls -l /.dtprofile`
	`ls -l /.Xdefaults`

Resultados Esperados: Los permisos de los archivos existentes es ***600*** o ***400*** y son propios del ***root***.	
Comentarios: Dependiendo de la configuración del sistema, todos los archivos listados anteriormente quizás no existan.	

Paso 2	Como ***root***, ejecute los siguientes comandos desde el directorio ***home*** del ***root***:
	`/usr/bin/more /.login`
	`/usr/bin/more /.profile`
	`/usr/bin/more /etc/profile`
	`/usr/bin/more /.cshrc`
	`/usr/bin/more /.kshrc`
	`/usr/bin/more /.emace`
	`/usr/bin/more /.exrc`
	`/usr/bin/more /.forward`
	`/usr/bin/more /.rhosts`
	`/usr/bin/more /.dtprofile`
	`/usr/bin/more /.Xdefaults`
	y sobre cualquier ejecutable que es referenciado en el archivo anteriormente visto, ejecute el siguiente comando:

	`ls -l`

Resultados Esperados:

Los permisos de todos los archivos referenciados en los archivos listados son ***600*** ó ***400*** y el propietario es el ***root***.

Objetivo 2.3.5:

Verificar que todos los ejecutables del ***root*** sean propiedad del ***root*** y no de nadie más.

Paso 1	Ejecute el siguiente comando: `ls -ldb / /sbin /etc/ /usr /usr/bin /usr/etc /usr/sbin /usr/lib`

Resultados Esperados:

Los archivos listados son propietarios del ***root*** y de nadie más.

Comentarios:

Todos los ejecutables por el ***root*** deben ser propiedad del ***root***, y no deben ser de nadie más, deben estar localizados en un directorio donde cada directorio en la ruta es propiedad del ***root***.

Objetivo 2.3.6:

Identificar todos los archivos escribibles y verificar su necesidad. Los archivos escribibles por todos, directorios y servicios representan un hueco potencial de seguridad en el sistema.

Paso 1	Como ***root***, ejecute los siguientes comandos:
	`/usr/bin/find / -type f \( -perm -2 -o -perm -20 \) -exec ls -ldb {} \;`
	`/usr/bin/find / -type d \( -perm -2 -o -perm -20 \) -exec ls -ldb {} \;`

Resultados Esperados:

No hay palabras inesperadas en archivos o directorios. Los archivos deben ser escribibles solamente si hay un requerimiento legítimo.

Comentarios:
Los siguientes archivos estuvieran más seguros si permanecieran escribibles por todos los usuarios
`/tmp` `/var/tmp`
`/var/preserve` `/var/mail`

Objetivo 2.3.7: Verificar que todo sea legible por los usuarios, pero no los sistemas de archivos que son propiedad del ***root***.	

Paso 1	Como ***root***, ejecute el siguiente comando:
	`/usr/bin/find / -perm -4 ! \( -perm -6022 \) \`
	`\( -type f -o -type d \) \`
	`! -user root -group 0 -exec ls -ldb {} \;`

Resultados Esperados: Cualquier salida de este comando indica que un archivo o directorio debe ser cambiado a ser propiedad del ***root***.	

Objetivo 2.3.8: Verificar que los archivos de inicio y apague del sistema son válidos y están protegidos.	
Paso 1	Como ***root***, ejecute el siguiente comando: `/usr/bin/find /etc \( -perm -2 -o -perm -20 \) -exec ls -ldb {} \;`
Resultados Esperados:	

No debe haber salida que indique que el directorio ***/etc*** y su contenido son escribibles por todos los usuarios	
Paso 2	Revise la configuración de los ***scripts*** de inicio y apagada del sistema y la configuración de archivos en ***/etc/rc[1-3].d***
Resultados Esperados: Cualquier tarea hecha en los ***scripts*** de inicio es ejecutada seguramente. Cualquier servicio iniciado o tarea ejecutada es aprobada. Cualquier directorio que contenga un ***script***, ejecutable, o un archivo de configuración que es ejecutado en los ***scripts rc***, durante el ***boot*** o el reinicio no es escribible por un usuario diferente al ***root***.	
Objetivo 2.3.9: Identificar los archivos suid y sgid en el sistema y verificar su necesidad.	
Paso 1	Como ***root***, ejecute el siguiente comando: `/bin/find / -type f \( -perm -4000 -o -perm -2000 \) \ -exec ls -ldb {} \;`
Resultados Esperados: Verificar todos los programas listados, estos deben tener activado el bit ***suid*** y ***sgid***.	

Objetivo 2.3.10: Identificar los archivos ***suid*** y ***sgid*** en el sistema y verificar su necesidad.	
Paso 1	Como ***root***, ejecute el siguiente comando: `/usr/bin/find / \( -type c -o -type b \) -exec ls -ldb {} \; \ | grep -v "/dev/"`
Resultados Esperados: No debe haber salida en este comando, indicando que hay dispositivos fuera el directorio ***/dev***	
Comentarios: Cualquier dispositivo fuera del directorio ***/dev*** es sospechoso. El comando	

/usr/sbin/ncheck -s puede ser usado para mostrar los archivos especiales y archivos con el bit ***suid*** activado.	
Paso 2	Como ***root***, ejecute el siguiente comando: `/usr/bin/find /dev ! \( -type l -o -type c -o -type b \) \ -exec ls -ldb {} \;`
Resultados Esperados: Todos los archivos en el directorio ***/dev*** son archivos especiales	
Paso 3	Como ***root***, ejecute el siguiente comando: `/usr/bin/find / \( -type c -o -type b \) ! -user root -exec ls -ldb {} \;`
Resultados Esperados: No hay dispositivos especiales que sean propiedad del ***root*** que no deban ser propiedad del ***root***.	

Objetivo 2.3.11: Verificar que los ***filesystem*** sean montados de una forma segura.	
Paso 1	Ejecute el siguiente comando: `/usr/bin/more /etc/vfstab`
Resultados Esperados: Con excepción de */* and ***/tmp***, todos los ***filesystem*** deben ser montados como "***readonly***" o "***nosuid***".	

Objetivo 2.3.12: Verficar si el uso de los comandos privilegiados (ej: ***su***) están registrados y qué mensajes muestra en la pantalla.	
Paso 1	Digite el siguiente comando: `/usr/bin/more /etc/default/su`
Resultados Esperados:	

Las siguientes líneas no deben tener el símbolo de comentario (#) al inicio de ellas.

```
SULOG=/var/adm/sulog
CONSOLE=/dev/console
SYSLOG=YES
```

Comentarios:

Las entradas en el archivo ***/etc/default/su*** determinan las condiciones por defecto en el comando ***su***

```
SULOG=/var/adm/sulog
```

Existe el archivo ***/var/adm/sulog*** que registra los intentos cada vez que se usa el comando ***su***, contine la información de quién lo usa, y cuando lo usa. Esto permite al administrador del sistema rastrear quién está utilizando la cuenta de superusuario.

```
CONSOLE=/dev/console
```

Permite mostrar en la pantalla los intentos de ganar acceso remotamente como superusuario.

```
SYSLOG=YES
```

Determina si ***syslog(3) LOG_AUTH*** debe ser usado para registrar todos los intentos de acceso con el comando ***su***. Los mensajes ***LOG_NOTICE*** son generados por el comando ***su*** para el ***root***, los mensajes ***LOG_INFO*** son generados por comando ***su*** para otros usuarios, y los mensajes ***LOG_CRIT*** son generados por intentos fallidos del comando ***su***

Objetivo 2.3.13:

Verificar que las Herramientas de Administración del Sistema esten configuradas correctamente y que su uso esté limitado a usuarios autorizados.

Paso 1	Ejecute el siguiente comando: `ls -l /usr/bin/admintool`

Resultados Esperados:

Los siguientes permisos son mostrados:

```
-r- xr- x--- bin bin /usr/bin/admintool
```

Nota: Existe un ***patch*** para corregir fallas en el programa ***admintool*** y es el siguiente ***Sun patch #103558- 0?***

Paso 2	Haga lo siguiente `/usr/bin/more /etc/init.d/conf`

Resultados Esperados:

El servicio ***admind*** no dede estar activado.

Paso 3	Ejecute el siguiente comando: `grep '^ sysadmin' /etc/group`

Resultados Esperados:

Solo los usuarios autorizados para ejecutar "***admintool***" deben ser miembros del grupo ***sysadmin***.

Comentarios:

Los permisos de ***admintool*** son garantizados para los usuarios que son miembros del grupo ***sysadmin***. Esto significa que un usuario que ejecuta una tarea que modifica, los datos de administración en un sistema usando ***admintool***, debe ser un miembro del grupo ***sysadmin*** en el sistema donde la tarea se está ejecuntando.

Objetivo 2.3.14:

Verificar el control de acceso a la información para los mapas de los dispositivos que son apropiados para cada dipositivo específico.	
Paso 1	Revise el siguiente archivo: `/etc/security/device_maps`

Resultados Esperados:

En el archivo ***/etc/security/device_maps***, sólo los archivos de dispositivos especiales que vienen con ***Solaris 2.5.1*** son identificados por cada dispositivo físico.

Comentarios:

El archivo ***device_ maps*** contiene el control de acceso a la información acerca de cada dispositivo físico. Cada dipositivo está representado por una línea que tiene la siguiente forma:

```
device-name: device-type: device-list
```

Donde,

`Device-name` es una cadena de caracteres ASCII denominada dispositivo físico.

`device-type` es una cadena de caracteres ASCII denominada tipo genérico de dispositivo.

`device-list` es una lista de dispositivos de archivos especiales asociada con dispositivos físicos. Este campo contiene un dispositivos especiales de archivos, ruta y nombres separados por un espacio en blanco.

2.4 SERVICIOS

Objetivo 2.4.0:

Verificar que solo los archivos que requieran servicios de red sean suministrados por el sistema.	

Paso 1	Ejecute los siguientes comandos:
	`ls –l /etc/inetd.conf`
	`/usr/bin/more /etc/inetd.conf`

Resultados Esperados: No se debe permitir a nadie a diferencia del ***root*** para escribir en este archivo.	
Comentarios: Colocarles comentarios o eliminar servicios de red innecesarios, en los principales sistemas se requiere solamente ***telnet y ftp***. ***TCP Wrappers*** debe ser instalado en el sistema para mejorar la autenticación de ***host*** y formas de conexión.	

Objetivo 2.4.1: Verificar que los ***logins*** estén configurados correctamente. Al único que se le debe permitir acceso desde el servidor es al ***root***.	

Paso 1	Ejecute los siguientes comandos:
	`ls –l /etc/default/login`
	`more /etc/default/login`

Resultados Esperados: No se debe permitir a nadie a diferencia del ***root*** para escribir en este archivo.	

Objetivo 2.4.2: Verfificar que el archivo de usuarios de ***ftp*** tengan las cuentas apropiadas. El archivo ***/etc/ftpusers*** contiene una lista de usuarios a los que no se permite

acceso ***ftp*** a ningún archivo, este archivo debe tener todas las cuantas que no son usadas por los usuarios actuales.	

Paso 1	Ejecute los siguientes comandos:
	`ls –l /etc/ftpusers`
	`/usr/bin/more /etc/ftpusers`

Resultados Esperados: No se debe permitir a nadie a diferencia del ***root*** para escribir en este archivo. Las cuentas que deben estar incluidas son: ***root***, ***uucp***, ***news***, ***bin***, ***ingress***, ***nobody***, y ***daemon***.	
Comentarios: Si este archivo falla, todos los usuarios son capaces de accesar al sistema via ***ftp***.	

Objetivo 2.4.3: Determinar si ***finger*** está disponible en el sistema. Si lo está verificar que esté configurado correctamente. El servicio ***finger*** no debe estar disponible a menos que sea absolutamente necesario.	
Paso 1	Ejecute el siguiente comando: `finger @localhost`
Resultados Esperados: Un mensaje de error indica que el servicio ***finger*** no está disponible.	
Paso 2	Si lo está, ejecute el siguiente comando: `finger 232341237821983792198732 8@localhost`
Resultados Esperados: La información de los usuarios que están conectados actualmente en el sistema es suministrada.	
Comentarios:	

Hay un ***bug*** en algunos casos que permite a ***finger*** descargar todo lo que se conoce del usuario al que lo solicita.

Objetivo 2.4.4: Determinar si ***tftp*** está disponible en el sistema y si lo está, verificar que haya sido configurado correctamente. ***tftp*** es usado para permitir a los usuarios conectarse sin ***password*** de forma automática a la red, si la máquina no está configurada como un servidor ***BOOT***, ***tftp*** deberá desactivado.	

Paso 1	Como un usuario normal, ejecute los siguientes comandos:
	`% tfpt`
	`tftp> connect localhost`
	`tftp> get /etc/passwd testfile`
	`tftp> quit`
	`% ls -l testfile`
	`% more testfile`
	`% rm testfile`

Resultados Esperados: Si ***tftp*** no responde con "***Error Code 2:Access Violation***" y transfiere el archivo, la versión debe ser reemplazada o configurada correctamente.	
Comentarios: La ruta (***/tftpboot***) debe ser incluida en la línea de comandos pasada al demonio ***tftpd*** en el archivo ***/etc/inetd.conf*** si ***tftpd*** está disponible.	

Objetivo 2.4.5: Veriificar que ***tftp*** no se ejecuta con privilegios.	
Paso 1	Ejecute el siguiente comando: `ls -l /usr/bin/tftp`

Resultados Esperados: Este programa no debe tener los bits ***suid*** y ***sgid*** activados.

Objetivo 2.4.6: Determine si el ***ftp anónimo*** esta disponoble en el sistema, si lo está verifique que esté configurado correctamente.	

Paso 1	Para comprobar si usted está ejecutando ***ftp anónimos*** intente conectarse a ***localhost*** usando ***ftp anónimo***. Asegúrese de dar una ***RFC 822***, como ***password***. Digite el siguiente comando para comprobar si ***ftp anónimo*** está disponible
	`% ftp`
	`name (localhost:idname): anonymous`

Resultados Esperados: Si el mensaje de error "***530 User anonymous unknown***" (o uno similar) es retornado es porque no está disponible. Si el sistema contesta con "***331 Guest login ok***" y pregunta por un ***password***, el ***ftp anonimo*** está disponible.	
Comentarios: El ***ftp anónimo*** no debe estar disponible a menos que halla una necesidad legítima.	
Paso 2	Si está disponible haga lo siguiente 1. Verifique que la cuenta ***ftp*** está creada y ha sido desactivada para remplazarla por ***NP*** en el campo de ***password***. 2. Verifique que la cuenta esté en un directorio especial, como ***/local_ftp***. 3. Verifique que el usuario propietario de ***ftp*** esté en el directorio ***home*** y que no sea escribible por nadie. 4. Verifique que el directorio ***ftp/bin*** sea propiedad del

	superusuario y de nadie más. Verifique que una copia del programa esté en este directorio. 5. Verifique que el directorio ***ftp/bin*** sea propiedad del superusuario y de nadie má. Verifique que una copia del ***password*** y del grupo de archivo esté en este directorio, con todos los campos con (*), la única cuenta que debe estar presente es la de ***ftp***.
Comentarios: Si ***ftp*** anónimo es requerido, sugiere que ***wu-ftp***, debe ser usado. ***WU-FTP*** adiciona seguridad a ***ftp***.	

Objetivo 2.4.7: Verificar que los alias "***decode***" y "***uudecode***" hayan sido eliminadas del archivo ***/etc/mail/aliases.***	
Paso 1	Digite el siguiente comando: `vi /etc/mail/aliases` Busque ***decode*** digitando "***/decode***" y después enter.

Resultados Esperados:
Un mensaje debe ser impreso en la parte superior de la pantalla como sigue
`Pattern not found`
`#decode: "|/usr/bin/uudecode"`	

Comentarios: Despúes de modificar el archivo ***/etc/mail/aliases*** se necesita ejecutar ***/usr/bin/newaliases*** para que los cambios tengan efecto.	

Objetivo 2.4.8: Verificar que la versión de ***sendmail*** sea actual. Se han descubierto recientes vulnerabilidades en ***sendmail***, instale un ***patche*** que corrija esta deficiencia o

instale un versión reciente de ***sendmail***.	
Paso 1	Ejecute el siguiente comando: `telnet localhost 25`
Resultados Esperados: La versión de ***sendmail*** debe ser mostrada `220 hostname Sendmail SMI-8.6/SMI-SVR4 ready at Fri, 22 Aug 1997 14:50:21`	

Paso 2	Intente los siguientes comandos, mientras esté conectado a ***sendmail***:
	`Wiz`
	`Debug`
	`Kill`

Resultados Esperados: Cada comando debe dar el siguiente resultado: `500 Command unrecognized` Si cualquiera da una respuesta diferente, debe ser cambiado por una versión actualizada.	
Paso 3	Desde el intérprete de comandos ***shell***, ejecute el siguiente comando: `/usr/lib/sendmail -d23937476383`
Resultados Esperados: Este comando no causa una ***segmentation fault***.	
Comentarios: En algunas versiones de ***sendmail***, es posible ganar acceso como ***root*** suministrando cantidades mayores, que el rango de espacio de direcciones normales que son usados en este arreglo, indexado por la bandera ***–d***. Si esto causa una ***segmentation fault***, entonces posiblemente aparecerá un ***bug*** en la versión de ***sendmail***. El problema es que los números en este rango quizás omitan el rango revisado un acceso negativo dentro del arreglo. Por lo tanto es posible escribir en lugares de memoria antes del arreglo	

Objetivo 2.4.9:	
Verificar que el archivo ***.netrc*** no sea usado.	

Paso 1	Como ***root***, ejecute el siguiente comando:
	`/usr/bin/find / -name .netrc -exec ls -ld {} \;`
	`-exec more {} \;`

Resultados Esperados: No debe haber ninguna salida a este comando. Cualquier salida indica la existencia de cualquier archivo en el sistema. La ruta, los permisos y los contenidos son listados.	
Comentarios: El archivo ***.netrc***, no debe existir en un sistema seguro. Si el responsable de la seguridad del sistema aprueba el uso del archivo ***.netrc***, para un propósito específico, no guarde la información de los ***passwords*** en el archivo ***.netrc***. Coloque los permisos del archivo ***.netrc*** en desactivado para escritura y lectura, para el grupo y todos (ej: ***600***).	

Objetivo 2.4.10:	
Determinar si cualquier archivo ***.rhost***, se está ejecutando en el sistema. La única forma segura de administrar los archivo ***.rhost*** es desactivándolos completamente del sistema. El administrador del sistema debe revisar a menudo, para ver si han ocurrido violaciones a esta política.	

Paso 1	Como ***root***, ejecute el siguiente comando:
	`/usr/bin/find / -name .rhosts -exec ls -ldb {}`
	`\; -exec more {} \;`

Resultados Esperados:	

No debe haber salida a este comando. Una salida significa que existe un archivo ***.rhost***, los usuarios no deben tener archivos ***.rhost***

Comentarios:

Cron debe ser usado periodicamente para revisar, reportar los contenidos y eliminar los archivos ***.rhost***

Si son necesarios los archivos ***.rhost*** (ej: ejecutar ***backups*** en una red) y su uso ha sido aprobado por el responsable de la seguridad, tenga en cuenta lo siguiente:

1. El primer carácter de cualquier archivo ***.rhost*** no es "-".
2. Los permisos de todos los archivos ***.rhost*** debe ser ***600***.
3. El propietario de cada archivo ***.rhost*** es la cuenta del propietario.
4. Los archivos ***.rhost*** no contienen ***"+"*** en ninguna línea.
5. El uso de grupos de red dentro de los archivos ***.rhost*** no permiten acceso a esta cuenta.
6. Los archivos ***.rhost*** no usan ***"!"*** o ***"#".***

Objetivo 2.4.11:

Verificar que los archivos en el servidor no sean escribibles por el grupo ni por todos. Debido a que el servidor ***NFS*** mapea del ***root*** a ninguno, se puede proteger los archivos y directorios en nuestro servidor, configurando los archivos del ***root*** y haciéndolos propiedad sólo de él.

Paso 1	Vea el archivo ***/etc/dfs/dfstab*** usando el siguiente comando : `/usr/bin/more /etc/dfs/dfstab`

	y por cada partición del ***filesystem*** ejecute el siguiente comando: `/usr/bin/find filesystem \( -perm -2 -o -perm -20 \) -exec ls -l {} \;`
Resultados Esperados: Ningún archivo debe ser mostrado	
Comentarios: ***NFS*** debe ser evitado si es posible	

Objetivo 2.4.12: Verificar las entradas apropiadas en ***dfstab***. La clave del ***root*** especifíca la lista de ***host*** a los que se les permite tener acceso de superusuario, en determinado ***filesystem***. La clave del acceso especifíca la lista de ***host*** (separados por coma) que son permitidos para montarlos en determinado ***filesystem***. Si la clave de acceso no es especificada por un ***filesystem*** en cualquier ***host*** de alguna parte de la red se puede montar un ***filesystem*** por ***NFS***.	
Paso 1	Vea el archivo ***/etc/dfs/dfstab*** usando el siguiente comando `/usr/bin/more /etc/dfs/dfstab`
Resultados Esperados: Las entradas en el archivo el archivo ***/etc/dfs/dfstab*** deben ser similares a las entradas que siguen: `share -F nfs -o ro=bear:dog,anon=-65534 /opt` Y siga los siguiente criterios: 1. Solamente son exportados los ***filesystems*** necesarios. 2. Solamente los ***host*** autorizados se les da acceso para exportar los	

filesystems.

3. Todas las entradas son habilitadas en los ***hostnames*** (preferiblemente una dirección ***IP***)
4. Los ***filesystem*** son particionados usando "***anon=-1***" para desactivar los accesos que no vienen acompañados por la ***ID*** del usuario.
5. El servidor ***NFS*** no es autoreferenciado, por nombre o especificación en la entrada de un "***localhost***".
6. Los ***filesytem*** para ser exportados son particionados como "***read-only***" excepto cuando es aprobado por el responsable de la seguridad.
7. Sólo los accesos necesarios son dados en la exportación de un ***filesystem***.
8. Los ***filesystem*** para ser exportados son particionados sin el bit ***suid***.
9. La opción "***root=***" no de ser usada.
1. El acceso debe ser garantizado por el grupo de red o ***host***.

Paso 2	Ejecute el siguiente comando y asegúrese que los propietarios y los permisos del archivo exportado sean correctos: `/usr/bin/ls -l /etc/dfs/dfstab`

Resultados Esperados:

El archivo ***/etc/dfs/dfstab*** tiene permiso ***644*** y es propiedad del ***root***.

Objetivo 2.4.13:

Verificar el símbolo ***"+"*** en la línea del archivo ***/etc/passwd*** para cada cliente ***NIS*** que haya sido correctamente ingresado. El signo más indica a los programas en **UNIX** que busquen un sistema de base de datos (como ***etc/passwd*** o ***/etc/group***) para preguntarle al servidor ***NIS*** por el resto del archivo. Verifique que el símbolo ***"+"*** en cada línea del archivo ***/etc/passwd*** para cada cliente ***NIS***, haya sido correctamente ingresado. Verifique que el símbolo ***"+"*** no haya sido adicionado al archivo ***/etc/passwd*** en el servidor

maestro ***NIS***.	

Paso 1	Regístrese como cada cliente ***NIS***,y visualice el archivo ***/etc/passwd***, la siguiente línea debe ser encontrada:
	`+:*:0:0:::`
	Verifique que no se haya intentado adicionar accidentalmente en la siguiente línea, si fue así, cualquiera puede ingresar al sistema colocando un símbolo ***"+"*** en el ***prompt*** del ***login***.
	`+::0:0:::`

Resultados Esperados: El servidor ***NIS*** y clientes, deben estar correctamente configurados.	
Comentarios: ***NIS*** debe ser anulado si es posible	

Objetivo 2.4.14: Revise el archivo ***/etc/host.equiv*** para verificar que las configuraciones por defecto, de "todos los host confiables", hayan sido cambiadas. Si hay entradas individuales en estos archivos, verifique que sean apropiadas.	
Paso 1	Ejecute el siguiente comando: `ls -l /etc/hosts.equiv; /usr/bin/more /etc/hosts.equiv`
Resultados Esperados: Muestra la siguiente respuesta `/etc/hosts.equiv: No such file or directory` `/etc/hosts.equiv: No such file or directory`	
Comentarios: Revise la presencia de ***/etc/host.equiv*** después de cada instalación del sistema opreativo o instalación de un ***patch***.	
Paso 2	Si el responsable de la seguridad aprueba el uso del archivo

	/etc/host.equiv para un propósito específico, ejecute el siguiente comando: `ls -l /etc/hosts.equiv; /usr/bin/more /etc/hosts.equiv`

Resultados Esperados:

1. El propietario del archivo ***/etc/hosts.equiv*** es el ***root***.
2. Los permisos del archivo ***/etc/hosts.equiv*** son ***600***
3. El primer carácter del archivo ***/etc/hosts.equiv*** no es ***"-".***
4. El archivo ***/etc/hosts.equiv*** no contiene una línea con el símbolo ***"+"***.
5. El archivo ***/etc/hosts.equiv*** lista solamente un pequeño número de host confiables y los listados están dentro de su dominio o bajo su administración.
6. El archivo ***/etc/hosts.equiv*** no incluye ***'!'*** o ***'#'***.
7. Todos los host en el archivo ***/etc/hosts.equiv*** son especificados unsando una dirección ***IP*** para disminiur ataques engañosos ***DNS spoof attacks***.
8. Use grupos de red en el archivo ***/etc/hosts.equiv*** para fácil administración.

2.5 AUTENTICACIÓN

Objetivo 2.5.0:

Verificar que no existan cuentas en el sistema que no hayan sido usadas dentro de un tiempo razonable.

Paso 1	Teclee y ejecute el siguiente ***script*** para determinar qué usuarios no se han conectado en el último mes: `#!/bin/sh` `date` `uname –a` `PATH=/bin:/usr/bin;export PATH` `umask 077` `THIS_MONTH=`date | /usr/bin/awk '{print $2}'` `/usr/bin/last | /usr/bin/grep $THIS_MONTH |` `/usr/bin/awk '{print $1}' \` `| /usr/bin/sort –u > users1$$` `cat /etc/passwd \| /usr/bin/awk –F: '{print $1}'` `\| /usr/bin/sort –u \` `> users2$$` `/usr/bin/comm –13 users[12]$$` `/usr/bin/rm –f users[12]$$`
Resultados Esperados: Ninguna cuenta con nombre de usuario debe mostrarse. Si alguna cuenta es retornada esta debe considerarse como cuenta durmiente y debe ser desactivada o borrada.	

Objetivo 2.5.1: Verificar que no haya ***GID*** duplicados.	
Paso 1	Revise el archivo ***/etc/group***. Teclee `cat /etc/group`
Resultados Esperados:	

No debe haber ***GID*** duplicados

Objetivo 2.5.2: Verificar que los ***ID*** de los usuarios sumistrados por la instalación no tengan password por defecto. Muchas cuentas vienen con el sistema operativo **UNIX** (esas cuentas están normalmente al comienzo del archivo de ***password*** (***/etc/passwd***) y tienen nombres como: ***bin***, ***uucp***, y ***news***.) estas cuentas deben ser desactivadas o se les debe cambiar el ***password***.	
Paso 1	Inspeccione el archivo ***/etc/passwd*** para ver los ***password*** por defecto. `cat /etc/passwd`
Resultados Esperados: Las cuentas suministradas con la instalación debe ser desativadas o asignarles un ***password***.	

Objetivo 2.5.3: Verificar que no haya ***UID*** duplicados.	
Paso 1	Revise el archivo ***/etc/passwd***. `cat /etc/passwd`
Resultados Esperados: No debe haber ***UID*** duplicados. Si los hay, estas cuentas deben ser desactivadas	

Objetivo 2.5.4:

Verificar que no haya cuentas ***guest*** o ***demos*** en el sistema. Ciertos sistemas vienen instalados con estas cuentas conocidas y ***passwords*** default. Se trata de chequear cuales cuentas son válidas y en caso de que no lo sean, es conveniente borrarlas o cambiar el ***password***. Ejemplo: ***guest/no password***, ***demos/no password***. Las cuentas ***guest*** presentan un hueco de seguridad. Por su naturaleza, esas cuentas son poco usadas, algunas son usadas siempre por personas que deben tener acceso a la maquina por periodos cortos de tiempo. La forma más segura para mantener las cuentas ***guest*** es instalarlas según su necesidad y borrarlas tan pronto como la gente deje de usarlas. Las cuentas ***guest*** nunca se les debe dar ***password*** sencillos como "***guest***" o "***visitor***" y nunca deben ser permitidas en el archivo de ***password*** cuando no se estén usando.

Paso 1	Como ***root*** , ejecute los siguientes comandos
	`cat /etc/shadow | grep guest`
	`cat /etc/shadow | grep demo`

Resultados Esperados:

Las cuentas ***guest*** o ***demo*** no deben existir. Si existen observe que estén bien configuradas.

Objetivo 2.5.5:

verificar que los archivos ***password*** y ***shadow*** tengan los permisos correctos

Paso 1	Haga lo siguiente:
	`ls -l /etc/passwd`
	`ls -l /etc/shadow`

Resultados Esperados:

Los resultados deben parecerse a estos:

```
-r--r--r--   1 root  sys     576 Apr 27 15:31 /etc/passwd

-r--------   1 root  sys     328 Apr 27 15:31 /etc/shadow
```

Los archivos ***password*** deben ser propiedad del ***root***. El archivo ***/etc/passwd*** debe estar en el modo ***0444***, y el archivo ***/etc/shadow/*** en el modo ***0400***

3 EJECUTANDO HERRAMIENTAS DE SEGURIDAD

3.1 SATAN

Correr ***Satan*** significa probar la seguridad de un sitio remoto. ***Satan*** tiene la capacidad de escanear un gran número de servidores dentro de una red; afortunadamente o desafortunadamente, usted no puede tener la autoridad o permiso para escanear todos los servidores. ***Satan*** nunca debería ser usado para escanear servidores que usted no tiene permiso explícito del dueño del servidor .

Recuerde – usted solo debe correr SATAN como "superusuario"!

Asumiendo que usted tiene la autoridad para hacerlo, es muy importante para arrancar con el escaneo lo siguiente:

1. Desde el panel de control en la interface ***HTML*** seleccione ***Run SATAN***. Este le pedirá la selección del objeto principal; teclee el servidor al cual quiere correrle ***Satan*** en caso de que este no esté ya en la caja de peticiones.
2. Seleccione, ***Scan the target host only***, o, si usted prefiere y tiene la autorización y la disponibilidad de tiempo (***Satan*** puede tomar varios minutos

para escanear un simple servidor al mas alto nivel de escaneo) seleccione ***Scan all hosts in the primary subnet***.

3. Seleccione un ***Normal Scan*** antes de arrancar. Mientras mas intensivo sea el escaneo este tomará mas tiempo para completarse.
4. Seleccione ***Start the scan***

Después de que la prueba es hecha, en la interface ***HTML***, ir a la sección de análisis de datos y reportes de ***Satan*** (***Reporting & Data Analysis***), observar primero la sección de vulnerabilidades (***Vulnerabilities***) y después examinar los demás métodos de información y permisos de confianza (***Information*** y ***Trust***).

OTRO ASPECTO IMPORTANTE

Recuerde que si tiene ***tcpd wrappers*** o algún otro mecanismo que sea capaz de hacer un ***finger*** inverso, apague esta opción antes de correr SATAN!. Existe una posibilidad razonable de que alguien mas en la red tenga la misma opción activada y si no se quiere entrar en una guerra de ***finger*** o en un ciclo infinito de ***finger*** o ***fingers*** yendo y viniendo entre usted y sus blancos. Asegúrese de activar la opción nuevamente después de finalizar la recolección de datos, por supuesto!.

3.1 REQUERIMENTOS DEL SISTEMA

Desafortunadamente ***Satán*** no es tan portátil como se quisiera, pero aún así puede correr en un número bastante grande de máquinas Unix. Uno de los principales problemas que se han tenido para que pueda ejecutar todas las tareas en un tiempo razonable es que se tenia que confiar en muchas herramientas disponibles al publico y olvirdase de muchas metodologías habituales de prueba. Se espera ajustar estos diseños a medida que se reciba mas información de los usuarios con el fin de hacerlo mas portátil y robusto para correr en muchas más plataformas.

3.1.1 Sistemas operativos

Actualmente se conoce que trabaja en los siguientes Sistemas Operativos :

- SunOS 4.1.3_UI
- SunOS 5.3
- Irix 5.3

3.1.2 Plataformas hardware

Satan ha sido probado en las siguientes maquinas:

- SPARCstation 4/75
- SPARCstation 5
- Indigo 2

Sin embargo debería correr en algunas más; se esta probando en (Aix, Bsd, Irix5, Hp-ux 9.x, SunOS 4 & 5 Sysv-r4, ultrix 4.x y posiblemente ser en Linux.).

3.1.3 Espacio en disco

Aproximadamente se requiere 20 mb de espacio en el disco para una instalación total con todos los paquetes suplementarios. El grueso de esto es debido a los otros paquetes de software que es necesario tener como ***mosaic*** o ***netscape*** (5.5 mb o 2.5 mb en un ***Sun***) y ***perl5*** (10mb) ; ***Satan*** en si mismo ocupa aproximadamente 2mb de espacio en disco incluyendo la documentación. Si los programas suplementarios están ya instalados no es necesario volverlos a instalar.

3.1.4 Memoria para la corrida

En cuanto a memoria este es otro asunto es bastante dependiente de cuantos ***host*** está barriendo o cuantos están en su base de datos, para tener una idea con aproximadamente 1500 ***host*** y 18000 ***facts*** (lista de todos los registros de salida emitidos por las herramientas ****.satan***. Estos registros son los que se han procesado por ***Satan*** para generar los reportes) se ocupó acerca de 14 megabytes de memoria en un ***sparc 4-75*** corriendo en un ***SunOS 4.1.3***. con aproximadamente 4700 ***host*** escaniados con cerca de 150000 ***facts*** ocupó cerca de 35 megabytes de memoria en una ***Indigo 2***. Sobra decir que el ***swaping*** es bastante costoso si no se tiene bastante memoria

3.1.5 Otras herramientas de software requeridos y donde se pueden conseguir

Es probable que el administrador se de cuenta de que puede no tener todo el software requerido para correr ***Satan*** ya en el sistema. Todo está ampliamente disponible en ***Internet*** tanto los ***browsers*** como también el ***perl versión 5.001*** o últimamente la ***version 5.002***.

3.2 ISS

3.2.1 Requerimientos del sistema

Materiales que usted necesitará

Estos ítems son requeridos o lo ayudarán cuando se vaya a instalar el ***Internet Scanner***:

- Máquina dedicada para correr el ***Internet Scanner*** versión 5.
- CDROM del ***Internet Scanner*** o archivo de instalación.
- ***Archivo de clave***. Suministrado por ***ISS***. Este archivo habilita las características del ***Internet Scanner***. Sin este, el ***Internet Scanner*** corre como una copia de evaluación, la cual provee únicamente pruebas ***Intranet*** para la computadora local.

3.2.1.1 Requerimientos mínimos

A continuación se muestran los requerimientos mínimos de recursos computacionales para correr el ***Internet Scanner*** para **UNIX**. Además de estos requerimientos usted debería tener un sistema dedicado donde corra el ***Internet Scanner***. No solamente con un sistema dedicado se maximiza el desempeño, sino que también se evita la posibilidad de acceso no autorizado a la máquina.

Item	**Requerimiento Mínimos**
Procesador	Estación de trabajo Unix con sistema operativo soportado.
Sistema Operacional	***SUNOS 4.1.3* o superior** ***Solaris 2.3* o superior** ***HP/UX 10.10* o superior.** ***IBM AIX 4.1.5* o superior** Para ***Linux***: ***ISS*** recomienda ***Linux kernel level 2.0.24*** o más alto. Algunas pruebas, tales como una bomba de ***ping*** (también conocida como *"**The Ping of Death**"*), se conocen que causan en versiones anteriores al ***Linux kernel*** el deterioro o el re-arranque de la máquina. Esto puede alterar un escaneo. Las estaciones de trabajo ***Solaris*** requieren ***Motif version 1.2.2*** o posteriores y ***X11R5*** o posteriores para correr los programas ***GUI***. ***Solaris 2.3***, ***Motif*** tercera-parte tal como ***Panorama version*** *1.2.2* o posteriores. ***Solaris 2.4***, opcional ***Sun Motif*** *version 1.2.2* o posteriores. ***Solaris 2.5*** y posteriores, ***Sun DE/Bundled Motif***.
Memoria (***RAM***)	Importante! ***ISS*** recomienda 64 MB de ***RAM*** por 1,000 servidores activos.

	Internet Scanner para UNIX es capaz de correr sobre sistemas con tan poco como 16 MB de memoria. Sin embargo, para un desempeño óptimo, corra ***Internet Scanner*** con 32 MB de memoria o más. 30 MB para instalación desde archivo. 40 MB para instalación desde ***CD***. 40 MB más 2.5 MB por cada 100 servidores**.**
Disco Duro	El mínimo espacio en disco requerido para instalar ***Internet Scanner*** es 30 MB para **UNIX**. Disponga de 300 MB inicialmente y monitoree el espacio utilizado en disco cuando el escaneo finaliza.
Privilegios de Usuarios	Debe conectarse como Super-usuario.
Red	***Ethernet*** o ***Token Ring*** (ver nota mostrada abajo) conectados a una red activa.
Protocolo	***TCP/IP****.*
Despliegue	Requiere un monitor que soporte resolución de 800x600 con un mínimo de 256 colores.
Visor del *WEB*	Necesitará un visor estándar ***WEB*** para ver los reportes y archivos de ayuda ***HTML***.

Nota*: *Internet Scanner sobre una red ***Token Ring*** no ejecuta las siguientes pruebas: ***Stealth Scan***, ***IP Spoofing/TCP sequence prediction***, ***UDP bomb***, y ***SYN flood***.

3.2.2 Arrancando internet scanner desde el GUI

3.2.2.1 Procedimiento

Para arrancar el ***Internet Scanner*** desde el escritorio, ejecute los siguientes pasos:

1. Conéctese como ***root***.

2. Cambie al directorio donde fue instalado el ***Internet Scanner*** en su sistema.

3. En la línea de comandos, corra ***bin/xiss*.**

Internet Scanner desplegará el siguiente menú:

Figura 1. ***Ventana del menú del Internet Scanner***

Donde

Item	**Descripción**
Scan	
Arrancar *Scan*	Iniciar el escaneo
Caja de descripción	Especifica la descripción del escaneo que aparecerá en la lista de Resultados del Scan y en la ventana de diálogos del reporte. Si esta caja esta en blanco, entonces el Scanner usa una descripción de la política.
Host a Scanear	
Archivo de Host	Muestra el nombre del archivo de host que usted cargó.
Botón *Load*	Reemplaza la lista de host o agrega los host a la lista

	de host.
Botón *Save*	Salva la lista de host que usted carga en un archivo de host.
Botón *Clear*	Limpia todos los host de la lista de host. En el prompt se mostrará la confirmación de que usted desearía para limpiar la lista de host.
Lista de Host	Mantiene la dirección IP del dispositivo a ser Scaneado.
Especificación del Host o Rangos	Permite agregar un simple host en un rango de hosts mediante la digitación de sus direcciones IP. Use un guión (-) para denotar el rango de direcciones. Separe las entradas con comas. 127..0.0..1—127.0..005,127.0.10,127.0.2.0
Botón *Add*	Agrega los host en la especificación de Host o en el rango de la lista de host.
Botón de agregar host activos	Agrega un host en la especificación de Host o en el rango que responde a un *ping* ***ICMP*** para la lista de host.
Botón *Remove*	Remueve los hosts en la lista especificada o en el rango de la lista de host.
Botón *Add From Key*	Agrega los hosts que especifica su clave para la lista de Host.
Botón *Add From Key status frame*	Agrega los hosts que especifica la clave para la lista de host. Muestra una información resumida acerca de las operaciones y el escaneo.

Iniciando un Escaneo

El botón ***Start Scan*** se oscurece momentáneamente antes que su etiqueta se cambie; esto pasa cuando el scan empieza y cuando termina.

Usted podrá cambiar el directorio donde se almacenarán los resultados del scan. Cuando se completan los cambios, seleccione el botón ***Ok***. Para continuar el escaneo o ***Cancel*** para abortar el escaneo.

Si la información del escaneo es válida(host, listas, claves, políticas, etc), la etiqueta del botón cambia a ***stop scan*** y el botón se muestra normalmente. Al mismo tiempo el escaneo continua.

Una vez que la etiqueta ***Stop scan*** aparece en el botón, el usuario está en diposición para editar las políticas, listas de host, aún cuando crea un nuevo reporte o un reporte existente, sin embargo, la ventana de estado no ajusta el tamaño.

Nota: No se puede empezar un escaneo, mientras que algún otro está en progreso o no puede generar un nuevo reporte de escaneo mientras hay alguno en progreso.

Cuando se completa el Escaneo, el botón cambia a ***Star scan***.

La ventana de estado mostrará el siguiente mensaje una vez el escaneo ha terminado.

**Scan Completed

Una ventana de diálogo aparecerá con un mensaje similar. Seleccione ***Ok***. Para continuar.

Ahora, usted puede generar nuevamente reportes a partir del último escaneo.

Arrancar el ***Internet Scanner*** desde una línea de comandos, es de utilidad cuando usted quiere programar los escaner que vaya hacer en un tiempo especificado o en algún intérvalo. La sintáxis de la línea de comando es:

iss/bin/iss [options] scan_range

Opciones de la línea de comandos

Las siguientes opciones pueden ser usadas desde la línea de comando del ***Internet Scanner***:

Opción	***Descripción***
-a *[hosts]*	Muestra la información de la licencia. Incluye los servidores que están corrientemente registrados.
-b	Corre el escaner en modo *background* (en el fondo de la aplicación)

-c *config_file*	Especifica el archivo de configuración en uso. (Por defecto es ***iss.config***).
-d *dir_path*	Defectos: YYYYMMDDHHMM. Especifica el nombre del directorio donde el resultado de scan será almacenado.
-D <desc>	Si no es configurado, ninguna descripción es usada. Configure la descripción para el escaner.
-k key file	Especifica el archivo de clave que va usar (por defecto es ***iss.key***)
-h help	Despliega la información de uso
-o outout_file	Por defecto el valor está ***iss.config***. Si ninguno es especificado, el defecto es ***iss.log***. El cual dirige la salida al archivo especificado.
-t host_file	Por defecto el valor está en ***iss.config***. Escanea usando el archivo del servidor especificado.
-V	Despliega la versión del programa ***Internet Scanner***.
-v	Activa el modo ***verbose***.
Scan_range	Una cadena especifica el rango para el escaner en la forma: A, B-C, D, E-F ,G-H donde las letras son las direcciones o nombres de los servidores.

Para la información detallada acerca de las vulnerabilidades individuales y sus arreglos, que se pueden encontrar en un escaneo, ver el catalago de vulnerabilidades de ***X-force*** disponible en

http://www.iss.net/xforce/.

3.3 CRACK

Crack es un programa adiivinador de ***passwords*** que es diseñado para localizar rápidamente inseguridad en el archivo de ***password*** de **UNIX**, escaneando el contenido del mismo, buscando a usuarios que han escogido una mala combinación ***login-password***.

3.3.1 REQUERIMIENTOS

- ✓ **UNIX** como sistema operativo.
- ✓ Compilador de **C**.
- ✓ Moderada cantidad de espacio en disco.
- ✓ Considerado tiempo de ***CPU***.
- ✓ Permiso del administrador del sistema.
- ✓ Privilegios del ***root***.
- ✓ Descompresor ***gzip***.
- ✓ Si trabaja en red o multiprocesamiento, necesitará "***perl***"

3.3.2 CONFIGURANDO *CRACK*

1. Desempaquete ***Crack***.
2. Edite el script de ***Crack***, configurando los valores de **CRACK_PATH**, ***C5FLAGS***, ***CC***, ***CFLAGS*** y ***LIBS*** para adaptarlo a su sistema operativo.

3. Si su sistema no usa la función ***cript(),*** haga lo siguiente:
   - ✓ Debe averiguear e investigar sobre como funciona ***Crack***
   - ✓ En otros sistemas, puede ser ***netBSD***, ***freeBSD***, algunas versiones de digital **UNIX** y ***OSF***.
4. Si usa la función ***cript(),*** haga lo siguiente:
   - ✓ Cambie el directorio a "***src/libdes***" y configurar el código de "***libdes***", para que éste compile su sistema cuando se le de la opción "***make***".

Libdes no es parte de ***Crack***, es una rápida y elegante implementación de ***DES***, la cual incluye una rápida versión del tradicional algoritmo ***cript()***.

Los usuarios de ***Crack*** deben usar ***libdes*** algunas veces para optimizar la velocidad, lea el archivo ***install***, y trabaje sin las banderas para compliar la librería ***libdes***.

3.3.3 ADVERTENCIA

- ✓ Lo primero que se debe hacer es editar la librería ***libdes*** en el orden que utilicen los compiladores de ***C***, puede ser ***gcc***.
- ✓ Los usuarios de máquinas de 64 bits, como ***DEC Alpha***, deben usar la opción ***DES_LONG***, que se encuantra documentada con la distribución de ***Crack***.

- ✓ ***FreeBSD*** y otros sistemas necesitan ***-DTERMIOS***, al remmplazar esta opción al comienzo de "***src/libdes/makefile***", no es funcionalmente crítico, pero ***libdes*** no compilará totalmente sin él.
- ✓ Los usuarios de ***gcc 2.7.10*** o anteriores generalmente obtienen mejores resultados especificando:

  CFLAGS=-O4 -fomit-frame-pointer -funroll-loops

  En el archivo "***Makefile***" o "***Makefile.uni***" (para ***GNU*** "***make***") es apropriado.

- ✓ Regrese al directorio ***Crack*** y haga lo siguiente

  Crack **-makeonly**

  Esto crea los archivos binarios y los construye en el directorio "***run/bin***". Si está usando ***Crack*** en modo de red, que construya los binarios en cada máquina con "***Crack -makeonly***", y no tendrá inconveniente cuando lo esté ejecutando.

3.3.4 EJECUTANDO *CRACK*.

La forma general de invocar ***Crack*** es la siguiente

Crack [optiones] [-fmt formato] [archivo ...]

Una vez ***Crack*** haya sido configurado usted está en capacidad de hacer lo siguiente:

"*Crack* -makeonly" y despúes "*Crack* -makedict"

Lo que debe crear y comprimir los diccionarios, en este punto haga una copia del archivo "***F.merged***" y ubíquelo en el directorio "***run***", esto preservará la información del los ***password*** que has obtenido.

Ejecute ***Crack*** [filename]

Ejemplo: ***Crack*** **-nice 10 /etc/passwd**

LISTA DE OPCIONES

-debug: Permite ver lo que hace el ***script***.

-recover: Usado cuando se reinicia de una ejecución terminada anormalmente, suprima ***rebuild*** del diccionario

-fgnd: Ejecuta ***Crack*** en segundo plano

-fmt format: Especifíca el formato de entrada

-from N: Inicia ***Crack*** con un número "***N***" determinado de reglas

-keep: Previene el borrado de archivos temporales usados para almacenar las entradas de ***Crack***

-mail: Envía un correo de advertencia a cualquiera que su ***password*** es crakeado, ***ver "scripts/nastygram".***

-network: Ejecuta ***Crack*** en modo de red.

-nice N: Ejecuta ***Crack*** en modo de prioridad reducida, asi que otros trabajos pueden tomar prioridad en la red

-makeonly: Usados para construir archivos binarios y diccionarios.

-makedict: Usados para construir archivos binarios y diccionarios.

-kill filename: Opciones internas usadas para soporte en la red.

-remote: Opciones internas usadas para soporte en la red. (ojo)

3.3.5 SISTEMA DE PASSWORDS SHADOW, NIS/YP

Si está ejecutando ***NIS***, la forma más simple de obtener algunos datos para ***Crackear*** los ***passwords*** es hacer lo siguiente:

* ***ypcat passwd > ypfile***
* ***Crack [optiones] ypfile***

Si su sistema usa archivos de ***passwords***, entonces la mejor forma de obtener la información del archivo de ***passwords*** es por medio de estos dos ***scripts***:

* ***shadmrg.aix***
* ***shadmrg.sv***

3.3.6 SALIDA DE *CRACK*

Crack genera salidas ilegibles, por lo tanto, para leer los resultados de una ejecución de ***Crack*** haga lo siguiente:

./Reporter [-quiet] [-html]

Este reporte muestra tanto los ***passwords*** que han sido rotos como los errores que han sido detectados en el archivo de ***passwords***. Los aciertos son listados cronológicamente.

La opción ***-quiet*** suprime la salida de errores en el reporte de ***Crack***