ATAQUES A CONTRASEÑAS CRACK

Marta Benítez González

José Gutiérrez Benítez

INDICE

- ¿Qué es un ataque a contraseña?
- Cómo genera y almacena Linux las contraseñas
- Data Encryption Standar (DES)
- Ruptura de contraseñas de Linux a través de ataque a diccionario
- Shadowing de contraseñas y la suite shadow
- Tras la instalación de suite shadow
- Otros aspectos de la seguridad de contraseñas
- Módulos de autebtificación de contraseña
- Otras soluciones para laseguridad de las contraseñas

Ataques a contraseña

- Son las diversas técnicas que incluyen cualquier acción dirigida a romper, descifrar o borrar contraseñas o cualquier mecanismo de seguridad de las mismas.

- En cuanto a seguridad el ataque a contraseña son lo primero porque pone en peligro a todo el sistema.

- Exige un enfoque a dos niveles:
  - Aplicar herramientas avanzadas para reforzar las contraseñas.
  - Educar a los usuarios en políticas de contraseña básicas.

Contraseñas

En Linux las contraseñas se almacenan en **/etc/passwd** de forma que no resulta seguro porque el fichero es legible.

Las contraseñas ocupan el segundo campo y están alteradas de forma que resulten incompresibles, han sido sometido a criptografía.

Conceptos:

Criptografía. Es la ciencia de escribir de forma secreta.

DES. Algoritmo de cifrado que utiliza linux cuando se crean las contraseñas. Se apoya en una clave y es un cifrado de bloque de 64 bits. Efectúa tres operaciones: permutación inicial (cambia de posición los caracteres), creación de un bloque de entrada que se reordena (transformación) obteniendo un bloque de pre-salida y por último, se aplica una permutación más y el resultado es el texto cifrado.

Ataques a diccionario

- El **DES** se puede romper fácilmente, por:
  - *El factor humano*: se eligen contraseñas débiles.
  - *Longitud limitada*: Las contraseñas son cortas. El número de transformaciones para cifrarlas es pequeño.

- Los atacantes toman *diccionarios* (grandes listas de palabras) y los codifican utilizando **DES**. Con el paso del tiempo pueden codificar cada palabra de 4096 formas distintas.

  Se compara el texto con las contraseñas hasta que se encuentre una coincidencia, por lo que, se ha roto la contraseña.

Ejemplo: Ataque a diccionario (I)

- Se necesita:
  - *Aplicación Crack*. Programa para encontrar contraseñas cifradas mediante DES.
  - *C* y hay que ser *root*
  - Los archivos de configuración: *dictgrps.conf*, *dictrun.conf* y *network.conf*

- Se ejecuta en varias fases:
  - Descomprimir Crack
  - Crear Crack
  - Ejecutar Crack
  - Observar los resultados

Ejemplo: Ataque a diccionario (II)

Descomprimir Crack

Se descomprime en el directorio root utilizando gunzip:

gunzip crack5.0.tar.gz

Se descomprime a un archivo de tipo tar llamado crack5.0.tar. Este archivo se descomprimirá usando el comando tar:

tar –xvf crack5.0.tar

Se creará un directorio llamado c50a, al que habrá que cambiarse para crear el crack.

Nota: Podrá leerse las notas específicas de configuración en manual.txt

Ejemplo: Ataque a diccionario (III)

Crear Crack

Se escribe las siguiente línea de comando:

./Crack –makeonly

Si compila correctamente, observará el siguiente mensaje:

```
all made in util
make[1]: Leaving directory '/root/c50a/src/util'
Crack: makeonly done
```

A continuación, se debe compilar los diccionarios:

Crack –makedit

Cuando acabe mostrará el siguiente mensaje:

```
Crack: Created new dictionary...
Crack: makedict done
```

Ejemplo: Ataque a diccionario (IV)

Ejecutar Crack

Se puede romper directamente el archivo **/etc/passwd** pero para mayor comodidad y entendimiento se ha copiado el archivo en *passwords.txt*.

Para ejecutar *Crack* se introduce el siguiente comando:

Crack passwords.txt

Se ejecutará como un proceso que se puede seguir utilizando el comando ps.
Se aplican muchas reglas a cada palabra. Se pueden observar las reglas actuales a través de los archivos de **/c50a/run** y examinar el conjunto de reglas básicas mediante el archivo **c50a/conf/rules.basic**.
Algunos ejemplos de reglas son los siguientes:

Regla	Resultado
append: $X	Se añade el carácter X al principio de la palabra
reverse: r	Escribe al revés la palabra actual
uppercase: u	Poneen mayúscula la palabra actual

Ejemplo: Ataque a diccionario (V)

Observar los resultados

Para saber si *Crack* ha adivinado correctamente las contraseñas, se utiliza la herramienta *Report* que se encuentra en el directorio **/c50a**:

./Reporter

Ejemplo. Se han encontrado 4 contraseñas en 2 minutos que eran muy débiles:

```
Guessed marty    [marty]      Marty Rush                [passwords.txt /bin/bash]
Guessed Nicole   [alexalex]   Caldera OpenLinux User   [passwords.txt /bin/bash]
Guessed manny    [willow]     Caldera OpenLinux User   [passwords.txt /bin/bash]
Guessed moe      [solace]     Caldera OpenLinux User   [passwords.txt /bin/bash]
```

Opciones de Crack

- Las opciones más comunes que admite Crack son las siguientes:

Opción	Propósito
-debug	Proporciona información estadística e informes de los procesos en tiempo real.
-fgnd	Se utiliza para ejecutar Crack en primer plano, de forma que se pueda observar lo que hace el proceso.
-from N	Para ejecutar Crack a partir de un número de regla determinada representado por el número N.
-mail	Se utiliza para que Crack envíe un e-mail a todos los usuarios cuyas contraseñas se han forzado.
-network	Para ejecutar Crack en modo de red, donde se pueden auditar utilizando varias máquinas a la vez.
-nice	Para designar Crack como un proceso de baja prioridad, lo que permite a otros procesos utilizar la CPU siempre que la necesiten.
-recover	Se utiliza cuando estamos reiniciando el proceso Crack debido a un fallo o una terminación anormal.

Alternativas a Crack

- **Accesorios de Crack: listas de palabras**

  Cuanto mayores sean las listas de palabras se incrementarán las posibilidades de dar con una contraseña.

- **Herramientas alternativas a Crack**

Herramienta	Descripción y Localización
John the Ripper	Herramienta de auditoria de propósito general para windows y Unix. Utiliza algoritmos propios. Admite muchas reglas y opciones y está bien documentada.
Lard	Herramienta de auditoria de contraseñas para Linux y otras versiones de Unix. Suficientemente pequeña para caber en un disket, lo que es útil para auditar equipos no conectados en red.
Xcrack	Un scrpit en Perl para romper contraseñas de Linux. Ejecuta un cifrado completo del archivo de diccionarios. Para entornos donde se espera que hayan contraseñas excepcionalmente malas.

Shadowing de contraseñas y la suite shadow

- El **shadowing** de contraseñas es una técnica mediante la que el archivo **/etc/passwd** sigue siendo legible pero ya no contiene las contraseñas. En su lugar, se almacenan en **/etc/shadow**.

- La herramienta más popular que realiza el *shadowing* es **Linux Password Shadow Suite**. Tras instalar el paquete de shadow se debe de examinar la base de dados de contraseñas de shadow **/etc/shadow**.
- Ejemplo:

```
root: 1LOTWOUA.YC2o:10713:0::7:7::
bin: *:10713:0::7:7::
ftp: *:10713:0::7:7::
Man: *:10713:0::7:7::
```

Shadowing de contraseñas y la suite shadow

- **/etc/shadow** consta de un registro por línea con nueve campos:
  - El nombre de usuario
  - La contraseña de usuario
  - Fecha en la que se cambió la contraseña por última vez
  - Número de días que queda para permitir al usuario cambiar su contraseña.
  - Número de días de antelación con que se avisa al usuario de que tendrá que cambiar su contraseña.
  - Número de días que queda para que el usuario cambie su contraseña antes de que su cuenta sea cancelada.
  - Número de días desde que la cuenta ha sido cancelada.
  - Está reservado

- Implementa dos nuevos conceptos: *Vencimiento de la contraseña* (tiempo de vida limitado) y *Bloqueo automático de cuenta* (se bloquean las cuentas sino cambian las contraseñas o se han caducado).

Utilidades de suite shadow y sus funciones

Utilidad	Función
Chage	Se utiliza para cambiar la información de expiración de contraseña de los usuarios.
Gpaswwd	Se utiliza para añadir nuevos usuarios a los grupos.
Groupadd	Se utiliza para añadir nuevos grupos
Id	Un sustituto de la suite shadow para el comando id. Es una utilidad que muestra el UID y la información asociada.
Passwd	Un sustituto de la suite shadow para el comando passwd. Sirve para crear nuevas contraseñas de usuario o para cambiar las existentes.
Userdel	Se utiliza para borrar usuarios. Este comando borrará al usuario y su directorio de origen.
usermod	Se utiliza para cambiar la información de un usuario(shell, tiempo de expiración de contraseña,etc.).

Añadir usuarios con shadowing: useradd

- Para añadir un usuario con **shadowing**, se utiliza **useradd**, que gestiona las entradas de */etc/passwd*, */etc/group* y */etc/shadow*.
- Opciones de la línea de comandos de **useradd**:

Opción	Propósito
-c	Para especificar el nombre real del usuario o un comentario.
-d	Para especificar el directorio de inicio del nuevo usuario.
-g	Para asignar el usuario a un grupo específico.
-m	Se utiliza para que useradd cree el directorio de inicio del nuevo usuario.
-s	Para especificar la shell predeterminada del nuevo usuario
u	Para especificar el UID del nuevo usuario

Añadir usuarios con shadowing: useradd

- Si se llama al **useradd** sin argumentos, aparece un resumen de uso:

```
usage: useradd [-u uid] [-g group] [-m] [-d home][-s shell][-r rootdir]
        [-e expire dd/mm/yyyy][-f inactive] name
useradd –D
useradd –v
```

Ejemplo: línea de comando que creará una entrada de usuario en */etc/passwd*, */etc/group* y */etc/shadow*:

/usr/sbin/useradd jsprat –m –c ”JackSprat” –u510 –g100 –ss/bin/bash

En */etc/passwd* se añade: **jsprat:x:510:100:Jack Sprat:/home/jsprat/:/bin/bash**

En */etc/shadow* se añade: **jsprat:*not set*:10715:0:-1:7:-1:-1:**

Nota: **useradd** no genera contraseñas,para ello se utiliza el comando passwd:

Ejemplo: **passwd jsprat**

Una vez, actualizado en */etc/shadow* aparecerá la información de la contraseña:

Ejemplo:**jsprat:cALtUMRf40VbU:10715:0:-1:7:-1:-1:1073897392**

Borrar usuarios en sistemas con shadowing: userdel

- Se utiliza **userdel**. Suprime la información del usuario de */etc/shadow /etc/passwd* y */etc/group* y la borra del todo.

- Para borrar un usuario se introduce el siguiente comando:

  userdel –r username

  La opción *–r* borra el directorio de inicio del usuario

Comandos útiles en sistemas con shadowing

- *Modificar el registro de un usuario*: **usermod**.
- **Usermod** puede modificar uno, varios o todos los campos del registro de un usuario.
- Las opciones son:

Opción	Propósito
-d [directorio inicio]	Modifica el directorio de inicio del usuario.
-e [fecha expiración]	Modifica la fecha de expiración de la contraseña de usuario.
-g [grupo inicial]	Modifica los datos de pertenencia al grupo inicial.
-I [nombre usuario]	Modifica el nombre de inicio de sesión del usuario.
-s [shell]	Para modificar la shell predeterminada del usuario.
-u [UID]	Para modificar el UID del usuario.

Comandos útiles en sistemas con shadowing

- *Verificar la base de datos de contraseñas*: **pwchk**
- Con el tiempo se realizarán numerosos cambios en la base de datos de contraseñas.
- Se debe verificar periódicamente la integridad de la base de datos de contraseñas, dado que con el tiempo se incrementa el riesgo.
- **Pwchk** verifica que toda la información de */etc/passwd* y de */etc/shadow* es válida. Asegura que el usuario y los grupos son válidos y que tienen *shells* válidos, que todos los campos están presentes y justificados y que todos los usuarios tienen un grupo apropiado y un *UID* único.

Comandos útiles en sistemas con shadowing

- *Añadir un grupo*: se utiliza **groupadd** con 2 opciones:

Opción	Propósito
-g [id grupo]	Se utiliza para especificar el GID.
-o	Se utiliza cuando se desea crear un GID que no sea único.

- *Modificar la información de un grupo*: se utiliza **groupmod** con 3 opciones:

Opción	Propósito
-g [id grupo]	Modifica el GID.
-n [nombre grupo]	Modifica el nombre de grupo.
-o	Se utiliza cuando se desea crear un GID no único.

Comandos útiles en sistemas con shadowing

- *Borrado de grupos*: **groupdel**

- *Gestionar el acceso a grupos*: **gpasswd**
  - Se utiliza para asignar administradores de grupos a grupos de usuarios. El administrador puede añadir o eliminar usuarios al grupo o limitar el acceso a éste, mediante contraseñas. Las opciones son:

Opción	Propósito
-A [nombre de usuario de administrador]	Especifica un administrador de grupo que se identifica por su nombre de usuario.
-a [nombre de usuario]	Se utiliza para añadir un usuario a un grupo.
-d [nombre de usuario]	Se utiliza para borrar un usuario de un grupo.
-r [grupo]	Se utiliza para quitar una contraseña de grupo.

Comandos útiles en sistemas con shadowing

- *Verificación de datos de grupos*: **grpchk**
  - Con el paso del tiempo se pueden realizar numerosos cambios en los datos de los grupos, por lo que se debe verificar periódicamente la integridad de la información de los grupos, debido al riesgo que se incrementa con el tiempo.
  - **grpchk** examina los datos de los grupos buscando posibles errores en el número de campos y en la validez de sus nombres, sus usuarios y sus administradores.
  - Si **grpchk** encuentra dichos errores, solicita corregirlos.

Más allá de la creación y borrado de usuarios y grupos

- La *suite shadow* cuenta con diversas utilidades para el mantenimiento general de las cuentas y de las bases de datos de autentificación.

- Cambiar los datos de expiración de la contraseña de un usuario: **chage**
  - **Chage** permite cambiar una, varias o todas las reglas utilizando estas opciones:

Opción	Propósito
-E [fecha expiración]	Modifica la fecha en que la cuenta de usuario expirará y será bloqueada.
-I [días antes de bloqueo]	Especifica cuántos días puede permanecer inactiva una cuenta con una contraseña expirada antes de ser bloqueada.
-M [nº máximo días]	Modifica el nº máximo de días durante los que es válida la contraseña del usuario.
-W [días de advertencia]	Modifica el número de días durante los que el sistema avisará al usuario de que hay que cambiar las contraseñas.

Más allá de la creación y borrado de usuarios y grupos

- *Mezclar y emparejar las bases de datos de /etc/passwd y /etc/shadow*:
  - Es posible que se tenga que migrar los datos de */etc/passwd* al formato de *shadow*. Para ello se utiliza **pwconv**.
  - **pwconv** no sólo lo permite sino que además permite integrar simultáneamente información con *shadowing* desde una base datos *shadow* existente.
  - **pwconv** tiene varios mecanismos de seguridad:
    - Si se introducen entradas que no tengan ninguna contraseña asignada, **pwconv** no las migra a */etc/shadow*
    - Utiliza la configuración predeterminada de expiración, aviso y bloqueo de cuentas que viene definida en */etc/login/defs*.
  - Para volver a convertir datos de *shadow* a */etc/passwd* se utiliza **pwunconv.**

Posibles ataques a un sistema con shadowing

- **Suite shadow** solo esconde las contraseñas.
- Los atacante conocen la **suite shadow** y trasladan su interés a */etc/shadow*. La diferencia es que */etc/shadow* es más difícil de alcanzar.
- La **suite shadow** es bastante segura en sí misma, pero su seguridad depende de la seguridad del sistema, ya que otras aplicaciones tienen agujeros de seguridad, que permiten leer y escribir en */etc/shadow*.
- Ataques dirigidos al acceso a */etc/shadow*:
  - *deshadow.c* = código fuente intruso para desproteger las entradas de */etc/shadow*
  - *telnet hole* = se puede provocar un error utilizando *telnet*, que revelará las contraseñas ocultas.
  - *shadowyank* = Aprovechando un agujero de *FTP*, captura las contraseñas ocultas de los fallos de *FTP*

Tras la instalación de suite shadow

- El **shadowing** de contraseñas es un excelente comienzo, pero no puede garantizar la seguridad de las contraseñas del sistema.

- Se debe ampliar el concepto de seguridad:
  - **Elección humana de contraseñas y seguridad del sistema**
  - **Comprobación proactiva de las contraseñas**
  - **Seguridad auxiliar de las contraseñas**

Elección humana de contraseñas y seguridad del sistema

- Se deben elegir contraseñas fuertes y no a partir de datos personales del usuario.
- *Crack* rompería cualquier contraseña de este tipo en segundos.
- Además, las computadoras actuales tienen una gran potencia de proceso, y las herramientas de ataque a diccionarios se han vuelto muy avanzadas.
- Por ello, las herramientas como *crack* son valiosas. Mediante la comprobación regular de la fortaleza de las contraseñas, es posible asegurarse de que ningún intruso pueda penetrar en la red aprovechando una mala elección de la contraseña.

Comprobación proactiva de contraseñas

- Hace que se eliminen las contraseñas débiles antes de su envío a la base de datos de contraseñas.
- Cuando un usuario crea una contraseña, ésta se comparar en primer lugar con una lista de palabras y una serie de reglas.
- Si la contraseña no cumple los requisitos de este proceso se obliga al usuario a elegir otra.
- Actualmente existen 3 comprobadores proactivos de contraseñas que prevalecen:
  - **Passwd+**
  - **Anlpasswd**
  - **npasswd**

Comprobador proactivo de contraseñas: passwd+

- Grandes capacidades de registro, de sesiones, errores, usuarios, reglas y éxito o fracaso en el cambio de una contraseña.
- Especificación del nº de caracteres significativos de la contraseña.
- Algunas reglas proporcionadas:
  - El nº oficina, tlfno, nombre de host y dominio prohibidas
  - Las contraseñas deben tener como mínimo n caracteres de longitud
  - Las contraseñas deben mezclar mayúsculas y minúsculas
  - El nombre y apellidos prohibidos (al derecho o al revés)
  - El nombre de conexión prohibido (al derecho y al revés)

Comprobador proactivo de contraseñas: anlpasswd

- Escrito en código *Perl*, bien documentado, utiliza el archivo de diccionarios que se elija y permite crear reglas personalizadas.

- Las reglas predeterminadas son:
  - Números con espacios
  - Mayúsculas y minúsculas con espacios
  - Todo en mayúscula o en minúscula
  - Todo números
  - La 1º letra mayúscula y números
  - Todas las combinaciones de las anteriores.

Comprobador proactivo de contraseñas: npasswd

- **npasswd** es un sustituto del comando *passwd* de UNIX y de los SO similares.
- Somete a las contraseñas de usuario a estrictas pruebas de capacidad de adivinación.
- **npasswd** está diseñado para complementar o reemplazar los programas estándar de cambio de contraseñas: *passwd*, *chfn* y *chsh*.
- Su distribución cuenta con un conjunto de herramientas de desarrollo para poder ampliarlo o incorporarlo a otras aplicaciones.

Otros aspectos de la seguridad de contraseñas

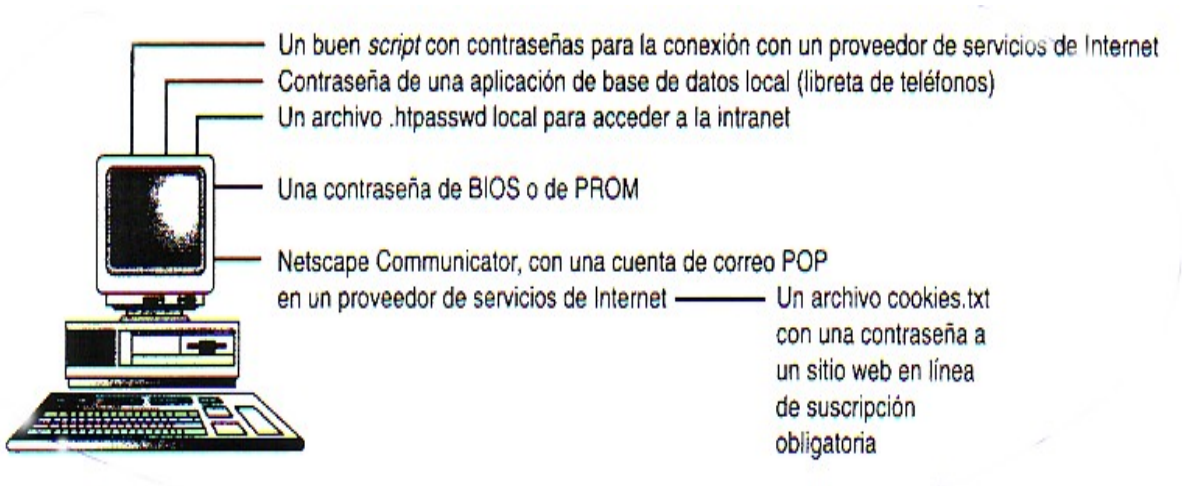
- El **shadowing** no garantiza la seguridad completa de las contraseñas, ya que una red Linux media existen muchos otros mecanismos de contraseña, muchos de los cuales no utilizan */etc/passwd* o */etc/shadow* para su autentificación.

- Examinaremos otras posibilidades de ataques y cómo se pueden resolver:

  - **Proliferación de contraseñas y seguridad**
  - **Módulos de autentificación que pueden conectarse**
  - **Otras soluciones para la seguridad de contraseñas**

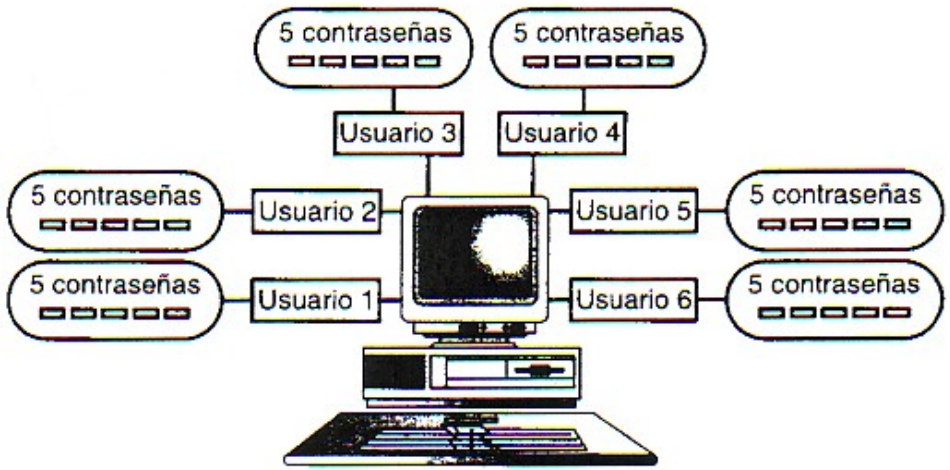
Proliferación de contraseñas y seguridad (I)

- Hasta el momento nos hemos centrado principalmente en las contraseñas de inicio de sesión.
- Sin embargo dentro de un esquema mayor, éstas son solo el principio.
- Puede existir la posibilidad de tener al menos 5 contraseñas:



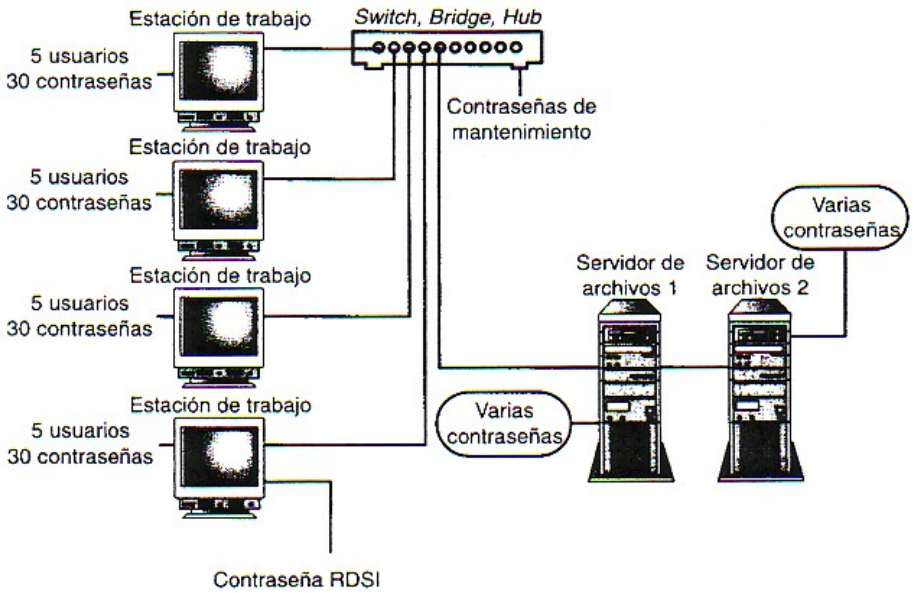
Proliferación de contraseñas y seguridad (II)

- Pero Linux es un sistema multiusuario y sabemos que es posible que tenga la intención de tener unos cuantos usuarios, por ejemplo 5:



Proliferación de contraseñas y seguridad (III)

- Para tener un sistema completo, supongamos que utiliza Linux en un entorno empresarial:



Proliferación de contraseñas y seguridad (IV)

- La red podría tener 200 contraseñas con 2 posibilidades:
  - **La mayoría de contraseñas iguales** = si los usuarios crean contraseñas idénticas para varias aplicaciones, si alguna de ellas tiene agujeros de seguridad, el sistema podría ser invadido.
  - **La mayoría de las contraseñas diferentes** = hace que los usuarios escojan contraseñas fáciles para no olvidarse, y por lo tanto, muy débiles.
- Demás, las aplicaciones no suelen implementar un almacenamiento de contraseñas completamente seguro.

Proliferación de contraseñas y seguridad (V): Consecuencias

- Incremento de las demandas del público de nuevas herramientas de red.
- Los desarrolladores siguen generando aplicaciones innovadoras y lanzándolas rápidamente al mercado, a menudo sin un control estricto de seguridad.
- El mercado de consumo se llena de aplicaciones que almacenan o transmiten contraseñas de forma insegura.

Proliferación de contraseñas y seguridad (VI): Consejos

- Limitar a los usuarios a un conjunto de aplicaciones establecidas y probadas que se conozcan a la perfección.
- En cada aplicación probada, verificar el almacenamiento de contraseñas y los procedimientos de transmisión.
- Para evaluar los procedimientos de transmisión de contraseñas, probar a espiar una conexión entre dos *hosts*, utilizando la aplicación bajo sospecha.
- Eliminar cualquier aplicación que emplee un mal almacenamiento de contraseñas y unos procedimientos de transmisión deficientes.
- Respecto a las aplicaciones probadas, estar atento a avisos urgentes de sus distribuidores.
- Probar la fortaleza del sistema de contraseñas una vez al mes, aunque se utilice la comprobación proactiva de las contraseñas.
- Educar al usuario a comprender la importancia de la seguridad de las contraseñas.

Módulos de autentificación que pueden conectarse (PAM)

- **PAM**, permiten cambiar la forma en que las aplicaciones de Linux ejecutan la autentificación sin tener que reescribirlas ni compilarlas.

- Algunos módulos **PAM** típicos son:
  - *Pam_cracklib* = un comprobador proactivo de contraseñas que puede conectarse
  - *Pam_deny* = fuerza la autentificación y deniega cualquier sesión en la que no se haya proporcionado una autentificación o ésta haya resultado fallida
  - *Pam_pwdb* = un módulo de base de datos de contraseñas que pueden conectarse, que proporciona expiración de contraseñas, avisos, etc.
  - *Pam_group* = asigna y rastrea la pertenencia a un grupo de los usuarios y de sus sesiones terminales

- Los **PAM** proporcionan muchas opciones de gestión de autentificación, de cuentas, de sesiones y de contraseñas, y se han utilizado para desarrollar operaciones de autentificación como la firma única (es cuando un usuario se autentifica una sola vez en una red de máquinas de confianza).

Otras soluciones para la seguridad de contraseñas

- Otras soluciones exóticas para la seguridad de las contraseñas:
  - **Controles de acceso biométrico** (seguridad física) = estas herramientas autentifican al usuario basándose en el olor corporal, estructura facial, huellas dactilares, patrones del iris o retina, voz, etc.
    - Son poco realistas debido a su alto coste
  - **Contraseñas que se utilizan una sola vez**: las contraseñas desechables no se transmiten por la red. En su lugar, el servidor reta al cliente con un valor numérico, que el cliente puede utilizar para generar un valor secreto adecuado para la transmisión de retorno.

FIN