2. CAPÍTULO DOS - PROYECTOS DESARROLLADOS EN LA SSI/ UNAM-CERT

En este capítulo se explican algunos de los proyectos que realicé durante el tiempo en el que me desempeñé en el área de respuesta a incidentes de seguridad de la información de UNAM-CERT. Los proyectos descritos tienen el propósito de:

a) Incrementar la capacidad de detección de incidentes en RED-UNAM.
b) Facilitar a los administradores de redes de datos de la UNAM la implementación de medidas para prevenir los incidentes de seguridad de la información de mayor frecuencia.

2.1. Manual de notificación y prevención de correo fraudulento

2.1.1. Introducción

Uno de los incidentes más comunes durante el tiempo que me desempeñé en el área de respuesta a incidentes en UNAM-CERT fue el robo de identidad que sufrieron miembros de la comunidad universitaria como resultado de correos electrónicos fraudulentos que recibieron en los que se les solicitó que proporcionaran su cuenta de correo y contraseña. Después de que las víctimas enviaron sus datos personales, los usuarios no autorizados las usaron para enviar más correos fraudulentos, correos no deseados (SPAM) y realizar otras acciones no autorizadas.

Con el fin de prevenir este incidente desarrollé un manual dirigido a usuarios del servidor de correo de la UNAM., la primera versión la realicé en agosto de 2010 y la he actualizado constantemente para ajustarla a las necesidades actuales del servidor de correo y de las amenazas. En el momento de elaboración de este informe, el manual se ajusta a la interfaz más reciente de Correo UNAM.

En el manual indiqué los pasos a seguir en caso de que los usuarios hubieran sido víctimas de correos fraudulentos, con el fin de que el impacto, por el robo de identidad fuera mínimo. Señalé la forma de acceder a las cabeceras de correo fraudulento para que pudieran reportarlas al equipo de respuesta a incidentes de UNAM-CERT y así rastrearan la computadora desde la que se envió el correo fraudulento y se notificara a las organizaciones correspondientes. Esto permitió que la respuesta de UNAM-CERT fuera menor, pues anteriormente se perdía mucho tiempo solicitando al usuario que enviara las cabeceras del correo fraudulento.

2.1.2. Objetivo del manual

Dar a conocer el problema de los correos fraudulentos a los usuarios de correo electrónico de los dominios unam.mx y servidor.unam.mx con el fin de que sean menos vulnerables a esta amenaza, así como facilitar a los usuarios la notificación adecuada de estos incidentes a UNAM-CERT.

2.1.3. Desarrollo

La gran mayoría las víctimas de correos fraudulentos tienen un nivel de conocimiento bajo sobre la seguridad informática y sistemas de cómputo en general. En razón de lo anterior, realicé el manual con un nivel muy elemental, a fin de que pudiera ser entendido por todo tipo de usuarios.

La difusión del manual se hizo a través del envío de un correo electrónico a todos los usuarios del servidor de correo y también mediante la página de UNAM-CERT, disponible en la página *http://www.seguridad.unam.mx/doc/?ap=manual&id=215*

Además, se colocó una referencia en la página principal de UNAM-CERT *www.seguridad.unam.mx* para que los usuarios pudieran acceder fácilmente a él, como se puede observar en la figura 2.1.

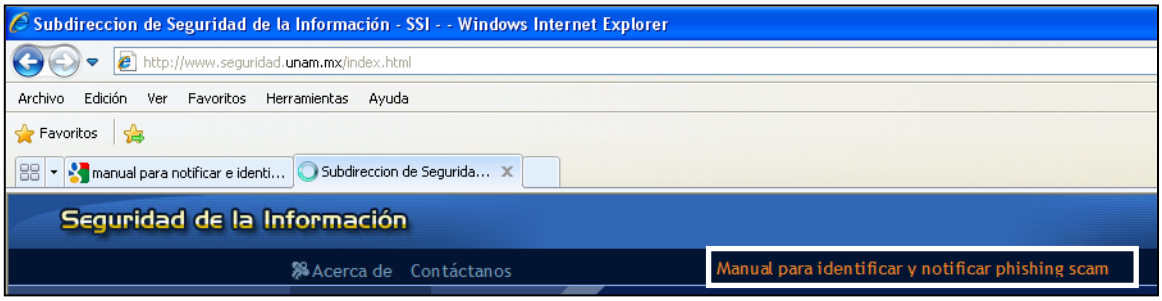

Fig. 2.1 – Referencia para manual para identificar y notificar phishing scam des el sitio de UNAM-CERT

Es importante señalar que la administración del servidor de correo de la DGTIC corresponde exclusivamente al Departamento de Servidores.

En el mes de enero de 2011 se realizaron cambios en el servidor de correo, debido a esto, el Departamento de Servidores de la DGTIC modificó la interfaz horde por una de *Squirrelmail*, haciendo necesario adaptar el manual a la nueva interfaz.

Además se aprovechó el cambio como una oportunidad para hacer más difusión al manual, por medio de un aviso en la página principal en el servidor de correo, el cual se muestra en la figura 2.2. Dicha acción se realizó en colaboración con el Departamento de Servidores de la DGTIC.

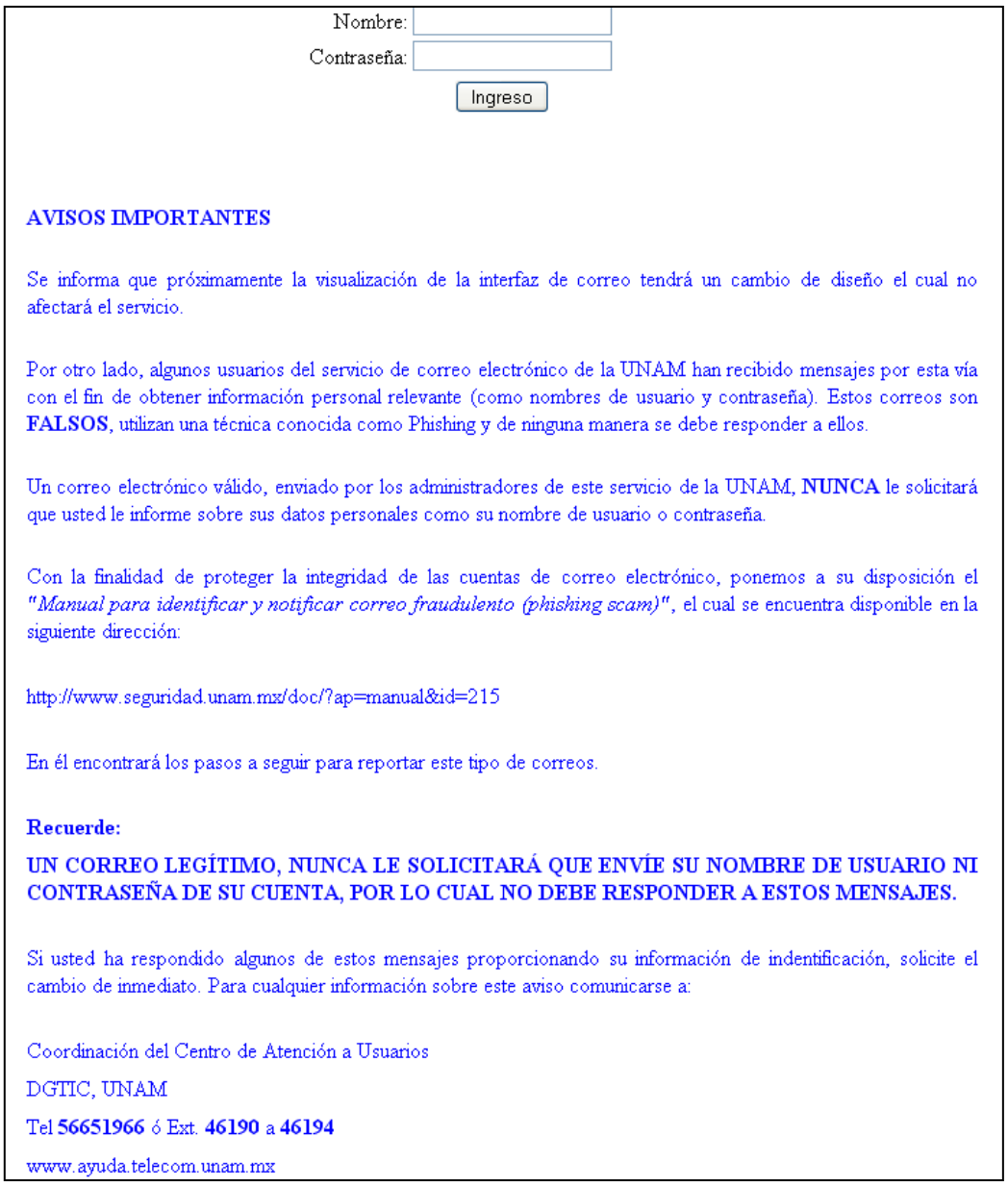

Fig. 2.2 Aviso en la página de Correo UNAM

2.1.4. Conclusión del manual

El manual de identificación y notificación de correos fraudulentos ayudó a prevenir que usuarios de *Correo UNAM* proporcionaran sus credenciales de acceso a usuarios no autorizados y que la notificación de este incidente por parte de los usuarios a UNAM-CERT mejorara.

La retroalimentación de la utilidad del manual fue inmediata. Rápidamente UNAM-CERT recibió reportes de usuarios que indicaban haber proporcionado sus credenciales de acceso a defraudadores, lo que permitió proporcionarles la asesoría adecuada para que cambiaran su contraseña, previniendo así acciones no autorizadas.

2.2. Manual de implementación de cracklib en sistemas LINUX

2.2.1. Introducción

Durante mi desempeño en el área de respuesta a incidentes en UNAM-CERT, observé que la causa más común de accesos no autorizados a servidores de RED-UNAM era el uso de contraseñas débiles.

A pesar de que muchos administradores indican correctamente a los usuarios cuáles son las reglas a seguir para escoger contraseñas robustas, esto no es suficiente. Se deben de aplicar las medidas necesarias para obligar el uso de contraseñas seguras. Para mitigar este problema desde el origen, me di a la tarea de buscar algún software que fuera gratuito, fácil de instalar y configurar. Si bien estas dos últimas características no son un requisito obligatorio, sí son deseables, pues muchos administradores prefieren no instalar herramientas de difícil configuración por temor a desestabilizar los sistemas de cómputo que están bajo su responsabilidad.

Ubuntu es una distribución derivada de Debian. Debido a la amplia adopción de la distribución Ubuntu en RED-UNAM, desarrollé este manual para Debian y sus distribuciones derivadas. Cracklib es una biblioteca que permite implementar una política de contraseñas más robustas en los sistemas LINUX/UNIX, comparar contraseñas contra diccionarios personalizados y almacenar un historial de contraseñas de los usuarios del sistema. De este modo, ayuda a mitigar, en cierta medida, una de las causas más comunes de intrusiones informáticas, las contraseñas débiles... Actualmente se está trabajando en un ambiente virtual de pruebas que permitirá tener múltiples sistemas operativos y crear manuales más generales que tengan un mayor impacto en la comunidad universitaria.

En LINUX/UNIX existe un software llamado cracklib que me sirvió para este propósito.

De acuerdo a la RAE un diccionario es un libro en el que se recogen y explican de forma ordenada voces de una o más lenguas, de una ciencia o de una materia determinada [1]. En términos de seguridad informática, sin embargo, sólo es relevante la lista de palabras, no su explicación.

Los diccionarios pueden ser modificados de acuerdo a las necesidades específicas de cada organización, con el propósito de evitar que los usuarios elijan alguna contraseña que se encuentre en éste. Así, un administrador podría crear diccionarios personalizados con las fechas de cumpleaños, apodos, proyectos o nombres específicos de la organización u otros datos que no se encuentran en los diccionarios estándar y que los intrusos podrían usar para tratar de vulnerar

2.2.2. Objetivo del manual

Proveer, mediante el módulo de PAM cracklib, una alternativa para incrementar la seguridad relacionada al manejo de contraseñas en sistemas operativos GNU/LINUX.

2.2.3. Desarrollo

La biblioteca cracklib tiene integración con PAM (Password Authentication Modules, que en inglés significa Módulos de Autenticación de contraseñas), el cual provee un sistema modular de autenticación de contraseñas y actúa como una interfaz entre aplicaciones de alto nivel y módulos de bajo nivel. Algunas tareas donde se utiliza PAM, son en procesos de autenticación, cambio de contraseñas o almacenamiento de éstas.

PAM puede ser configurado para soportar autenticación local, en la que se usa el archivo de usuarios y contraseñas locales, o autenticación remota, por medio de un protocolo de directorios como LDAP o Active Directory. En este manual se usa la autenticación local.

La biblioteca realiza varias pruebas en las contraseñas para determinar si cumplen con determinadas características de seguridad. El objetivo es reducir el riesgo de que los usuarios escojan contraseñas débiles y fáciles de vulnerar.

Cracklib asigna a cada tipo de carácter de la contraseña un valor en créditos. La suma de créditos para la contraseña debe ser superior al parámetro minlen para que ésta sea aceptada, además de pasar la validación de historial y diccionario.

Cracklib también permite consultar el historial de contraseñas de cada usuario para evitar que usen una que ya habían usado anteriormente. Las contraseñas se guardan por medio del módulo pam_unix y el número de contraseñas que se pueden almacenar es configurable.

2.2.4. Instalación

En los sistemas basados en Debian, como Ubuntu, es posible instalar paquetes pre compilados desde almacenes en servidores remotos, conocidos como repositorios. Esta práctica fue desarrollada como una manera fácil, rápida y eficiente de instalar aplicaciones.

Se puede instalar cracklib por repositorios de paquetes de la siguiente manera:

- Debian y derivados(Ubuntu)

```
# apt-get install libpam-cracklib
```

Con esto se instalarán los siguientes paquetes extras:

```
libcrack2 libpam-cracklib
```

Confirmar y completar la instalación.

En la figura 2.3 se observa la ejecución de la instalación de cracklib por medio de repositorios en el sistema operativo GNU/LINUX Debian versión 5. Se observa que de manera predeterminada el sistema incluye en la instalación las bibliotecas libcrack2 y libpam-cracklib.

```
chacmol:/# apt-get install libpam-cracklib
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalaron de forma automática los siguientes paquetes y ya no son necesarios
.
  libsilc-1.1-2
Utilice «apt-get autoremove» para eliminarlos.
Se instalarán los siguientes paquetes extras:
  cracklib-runtime libcrack2
Se instalarán los siguientes paquetes NUEVOS:
  cracklib-runtime libcrack2 libpam-cracklib
0 actualizados, 3 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 280kB de archivos.
Se utilizarán 901kB de espacio de disco adicional después de esta operación.
¿Desea continuar [S/n]? S
Des:1 http://mmc.geofisica.unam.mx lenny/main libcrack2 2.8.12-8lenny1 [190kB]
Des:2 http://mmc.geofisica.unam.mx lenny/main cracklib-runtime 2.8.12-8lenny1 [2
4.2kB]
Des:3 http://mmc.geofisica.unam.mx lenny/main libpam-cracklib 1.0.1-5+lenny1 [65
.4kB]
Descargados 280kB en 0s (1296kB/s)
Seleccionando el paquete libcrack2 previamente no seleccionado.
(Leyendo la base de datos ...
90336 ficheros y directorios instalados actualmente.)
Desempaquetando libcrack2 (de .../libcrack2_2.8.12-8lenny1_i386.deb) ...
Seleccionando el paquete cracklib-runtime previamente no seleccionado.
Desempaquetando cracklib-runtime (de .../cracklib-runtime_2.8.12-8lenny1_i386.de
b) ...
Seleccionando el paquete libpam-cracklib previamente no seleccionado.
Desempaquetando libpam-cracklib (de .../libpam-cracklib_1.0.1-5+lenny1_i386.deb)
 ...
Procesando disparadores para man-db ...
Configurando libcrack2 (2.8.12-8lenny1) ...
Configurando cracklib-runtime (2.8.12-8lenny1) ...
Configurando libpam-cracklib (1.0.1-5+lenny1) ...
```

Fig. 2.3 – Instalación de cracklib por repositorios en GNU/LINUX Debian versión 5

2.2.5. Configuración

A continuación se edita el archivo de configuración de autenticación de PAM para reemplazar el módulo `pam_unix` por `pam_cracklib` como el primer módulo de autenticación de PAM.

```
# vi /etc/pam.d/common-password
```

Comentar la siguiente línea:

```
password required pam_unix.so nullok obscure md5
```

Con lo anterior se deshabilita el uso del módulo pam_ y se agrega la siguiente línea para habilitar cracklib (una vez que se ha asignado un valor a cada parámetro):

```
password required pam_cracklib.so retry=int1 minlen=int2 dcredit=int3 ucredit=int4
lcredit=int5 ocredit=int6 difok=int7
```

En la línea anterior es necesario sustituir los valores int_i por enteros definidos con base en las necesidades propias de cada organización. Más adelante se muestran algunos ejemplos.

En la tabla 2.1 se muestran los parámetros de configuración de cracklib.

Parámetro	Explicación
retry	Número de intentos de autenticación antes de abortar el proceso y regresar error.
minlen	Número de créditos mínimo de créditos de la contraseña.
Difok	Número de caracteres diferentes entre sí debe ser 6 dentro de la contraseña escogida.
Lcredit	Créditos adicionales si la contraseña tiene minúsculas.
Dcredit	Créditos adicionales si la contraseña tiene dígitos.
Ucredit	Créditos adicionales si la contraseña tiene mayúsculas.
Ocredit	Créditos adicionales si la contraseña tiene caracteres que no son minúsculas, mayúsculas ni dígitos.

Tabla 2.1 Parámetros de configuración de cracklib

Los parámetros `dcredit`, `ucredit`, `lcredit` y `ocredit` son opcionales, si no se especifican en el archivo de configuración, tienen el valor predeterminado de 1.

Los valores de créditos se agregan a la longitud de la contraseña, en la tabla 2.2 se muestra un ejemplo de cómo cracklib determinar los créditos de la contraseña "*Yiulda4=*".

Contraseña	Valor de parámetros	Explicación
Yiulda4=	ocredit=2	Longitud = 8
	ucredit=1	Crédito por mayúsculas= 1
	dcredit=1	Crédito por dígito = 1
	lcredit=0	Crédito por símbolos especiales = 2
	minlen=12	(No se agrega nada por usar minúsculas, puesto que lcredit=0)
		Valor de la contraseña = 8+1+1+2 = 12
		El valor de la contraseña es igual o mayor que minlen, por lo que la contraseña es acepta.

Tabla 2.2 Ejemplo de funcionamiento de cracklib

Después de la línea que se agregó anteriormente, se añade la siguiente línea:

```
password required pam_unix.so remember=n use_authtok md5
```

Esta línea indica que después de que cracklib realicé la verificación de la contraseña, le cederá el control al módulo pam_unix.so, el cual guarda las últimas contraseñas de cada usuario, definidas por el valor de n.

En la figura 2.4 se puede observar cómo luce el archivo `/etc/pam.d/common-password` después de instalar y configurar la biblioteca de PAM cracklib.

```
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords.  The default is pam_unix.

# Explanation of pam_unix options:
#
# The "nullok" option allows users to change an empty password, else
# empty passwords are treated as locked accounts.
#
# The "md5" option enables MD5 passwords.  Without this option, the
# default is Unix crypt.
#
# The "obscure" option replaces the old `OBSCURE_CHECKS_ENAB' option in
# login.defs.
#
# You can also use the "min" option to enforce the length of the new
# password.
#
# See the pam_unix manpage for other options.

#password   required   pam_unix.so nullok obscure md5

# Alternate strength checking for password. Note that this
# requires the libpam-cracklib package to be installed.
# You will need to comment out the password line above and
# uncomment the next two in order to use this.
# (Replaces the `OBSCURE_CHECKS_ENAB', `CRACKLIB_DICTPATH')
#
 password required        pam_cracklib.so retry=3 minlen=12 dcredit=1 ucredit=1 ocredit=1 difok=3
 password required        pam_unix.so remember=12 use_authtok md5
```

Fig. 2.4 Configuración de cracklib en el archivo de configuración de PAM

2.2.6. Diccionarios

Hay dos tipos, los comunes, que contienen palabras frecuentes y que los intrusos informáticos suelen utilizar para obtener accesos no autorizados, y los personalizados, los cuales contienen palabras definidas por el usuario. Cracklib verifica que las contraseñas no se encuentren en alguno de estos diccionarios.

2.2.6.1. Diccionarios comunes

Pueden ser agregados por repositorios o manualmente en el directorio de diccionarios. Para agregarlos mediante repositorios se ejecuta el siguiente comando:

```
# apt-get install cracklib-runtime <lista de diccionarios>
```

Para ver la lista completa de idiomas se puede consultar la descripción del paquete wordlist en la página *http://packages.debian.org/sid/wordlist*

En la fecha de elaboración de este manual, algunos de los diccionarios disponibles por repositorios son los siguientes:

```
wspanish 1.0.19
wfrench 1.2.3-1
wbritish 6-2.1
wamerican 6-2.1
```

Para agregar diccionarios manualmente basta colocar el diccionario en el directorio de diccionarios de cracklib y ejecutar el comando de actualización de diccionarios.

Es posible descargar un diccionario común más grande que los diccionarios disponibles por repositorios, directamente de la página del proyecto cracklib, mediante el siguiente comando:

```
wget http://sourceforge.net/projects/cracklib/files/cracklib-words/2008-05-
   07/cracklib-words-20080507.gz
```

Se mueve el diccionario comprimido al directorio de diccionarios

```
mv cracklib-words-20080507.gz /usr/share/dict
```

Se descomprime el diccionario

```
gzip –d cracklib-words-20080507.gz
```

Para agregar el diccionario descargado manualmente, se ejecuta el siguiente comando:

```
# update-cracklib
```

En la figura 2.5 se muestra cómo agregar los diccionarios predeterminados en español, inglés americano e inglés británico mediante repositorios.

```
chacmol:/# apt-get install cracklib-runtime wspanish wamerican wbritish wfrench
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
cracklib-runtime ya está en su versión más reciente.
fijado cracklib-runtime como instalado manualmente.
wspanish ya está en su versión más reciente.
wamerican ya está en su versión más reciente.
Se instalaron de forma automática los siguientes paquetes y ya no son necesarios.
  libsilc-1.1-2
Utilice «apt-get autoremove» para eliminarlos.
Se instalarán los siguientes paquetes NUEVOS:
  wbritish wfrench
0 actualizados, 2 se instalarán, 0 para eliminar y 0 no actualizados.
Necesito descargar 625kB de archivos.
Se utilizarán 2679kB de espacio de disco adicional después de esta operación.
Des:1 http://mmc.geofisica.unam.mx lenny/main wbritish 6-2.3 [267kB]
Des:2 http://mmc.geofisica.unam.mx lenny/main wfrench 1.2.3-6 [357kB]
Descargados 625kB en 0s (2102kB/s)
Preconfigurando paquetes ...
Seleccionando el paquete wbritish previamente no seleccionado.
(Leyendo la base de datos ...
91838 ficheros y directorios instalados actualmente.)
Desempaquetando wbritish (de .../wbritish_6-2.3_all.deb) ...
Seleccionando el paquete wfrench previamente no seleccionado.
Desempaquetando wfrench (de .../wfrench_1.2.3-6_all.deb) ...
Procesando disparadores para man-db ...
Configurando wbritish (6-2.3) ...
Configurando wfrench (1.2.3-6) ...
```

Fig.2.5 Instalación de diccionarios comunes mediante repositorios

2.2.6.2. Diccionarios personalizados

Se pueden agregar diccionarios personalizados en el directorio */usr/share/dict/*.

Se crea el diccionario llamado independencia, con nombres de personajes históricos mexicanos de la este periodo histórico.

```
# vi /usr/share/dict/independencia
```

En este archivo agregamos las siguientes líneas:

```
MiguelHidalgo

MariaMorelos

JosefaOrtiz
```

2.2.6.3. Comprimir los diccionarios

Cracklib no revisa secuencialmente los diccionarios, en su lugar los comprime en un solo archivo binario. Esta funcionalidad le permite usar grandes diccionarios en un tiempo menor al que tardaría de no usarse compresión. Para indicar a cracklib que comprima los archivos de texto y genere el archivo binario, se ejecuta el siguiente comando:

```
# update-cracklib
```

La ejecución del comando se observa en la figura 2.6, el programa imprime dos números, el primero es el número de palabras leídas de los archivos de texto y el segundo es el número de palabras eventualmente escritas en el diccionario. Esto permite detectar el grade de redundancia en las palabras presentes en los diccionarios.

```
chacmol:/usr/share/dict# update-cracklib
293509 293508
```

Fig. 2.6 Ejecución del comando update-cracklib

2.2.7. Historial de contraseñas

La opción remember del módulo pam_unix guarda las viejas contraseñas en un formato similar al archivo shadow. Es decir, almacena los hashes de las últimas contraseñas de cada usuario. La ubicación de este archivo es /etc/security/opasswd, el cual debe ser protegido como el archivo shadow, verificando que tenga permisos de lectura y escritura sólo para root, si no es así, se deben asignar mediante el siguiente comando:

```
# chown 0600 /etc/security/opasswd
```

Cuando el módulo pam_cracklib es ejecutado, se verifica que la contraseña no sea igual a alguna otra que el usuario haya utilizado anteriormente.

2.2.7.1. Prueba de concepto del historial de contraseñas

En la figura 2.7 se muestra la terminal del usuario "prueba2", quien cambia su contraseña mediante el comando `passwd`. Se observa que cracklib rechaza aquellas contraseñas que no cumplen los criterios establecidos.

```
prueba2@chacmol:~$ passwd
Cambiando la contraseña de prueba2.
(actual) contraseña de UNIX:
Nueva UNIX contraseña:
CONTRASEÑA INCORRECTA: es demasiado similar a la antigua
Nueva UNIX contraseña:
Vuelva a escribir la nueva UNIX contraseña:
passwd: contraseña actualizada correctamente
```

Fig. 2.7 Prueba de concepto de robustez de contraseñas

En la figura 2.8 se observa el contenido del archivo `/etc/security/opasswd`, en donde se almacena el hash de la contraseña anterior del usuario "prueba2", tras modificar su contraseña mediante el comando passwd.

```
chacmol:/# cat /etc/security/opasswd
prueba2:1002:1:$1$fkwPKCO3$3LglixYqnmEZ.itZEdGkI/
```

Fig. 2.8 Prueba de historial de contraseñas

2.2.8. Prueba de concepto de cracklib

Se probó cracklib en un sistema GNU/LINUX Debian versión 5 con configuración predeterminada, el cual utiliza el módulo `pam_unix` para la verificación de las contraseñas. Se dio de alta un usuario y se escogió una contraseña débil. Luego se instaló cracklib y se mostró que ya no era posible que el usuario escogiera una contraseña débil, asimismo se probó la funcionalidad de diccionarios.

La configuración de seguridad proporcionada por el módulo `pam_unix`, rechaza algunas contraseñas débiles, como "123456" y "qwerty", pero permite otras contraseñas triviales como "password", "administrator" o "pruebas1". Un problema importante es que no verifica que la contraseña no se base en el nombre de usuario, esto representa una grave vulnerabilidad explotable, sobre todo en equipos que tienen servicios de acceso remoto habilitados y accesibles desde Internet.

En la figura 2.9 se muestra cómo el módulo de PAM cracklib no permite que se escoja una contraseña basada en el nombre de usuario, ni contraseñas que se encuentren en alguno de los diccionarios. En primer lugar se intentó escoger la contraseña "prueba123", en segundo lugar la contraseña "administrator". La primera no fue aceptada por estar basada en el nombre de usuario, la segunda tampoco por estar en el diccionario.

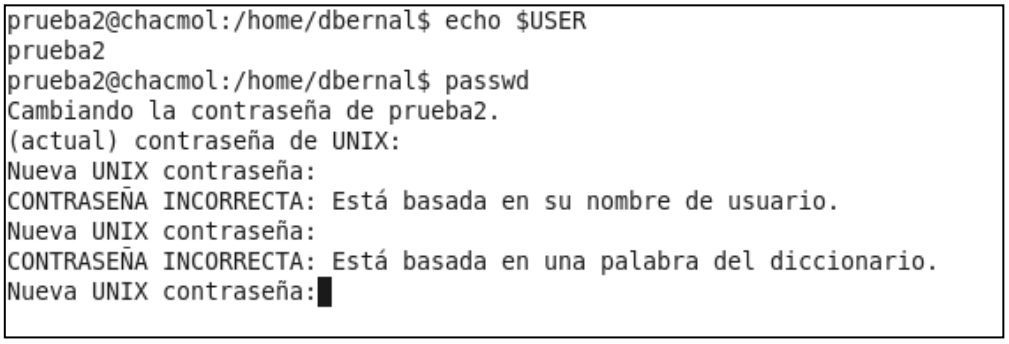

Fig. 2.9 Prueba de concepto de verificaciones de cracklib

Aunque una contraseña no sea de diccionario ni se base en el nombre de usuario, tampoco será aceptada si no cumple con la condición definida por el parámetro *minlen*, el cual es configurable mediante el módulo de cracklib y define el número de créditos que deben cumplirse para que la contraseña sea aceptada.

2.2.9. Conclusión del manual

Es importante recordar que cracklib sólo obliga a los usuarios no privilegiados a seleccionar contraseñas robustas, no así para el usuario root a quien sólo le mostrará una advertencia, es por eso que la cesión de privilegios de administradores a otros usuarios debe supervisarse.

La instalación de cracklib en sistemas operativos LINUX, ayuda a incrementar la seguridad de los equipos ante ataques de fuerza bruta y de diccionario. La funcionalidad de diccionarios

personalizados permite mitigar hasta cierto punto los ataques de ingeniería social que un intruso podría hacer al buscar información específica de la persona que desea atacar. Es responsabilidad de cada dependencia definir qué procedimientos seguir para crear estos diccionarios personalizados.

La instalación de cracklib en las distribuciones LINUX basadas en Debian (como Ubuntu) mediante repositorios tiene una mínima posibilidad de afectar la estabilidad del sistema, ya que no requiere la ejecución de pasos adicionales, que podrían llegar a afectar el funcionamiento del sistema si no son ejecutadas correctamente. Además, permite instalarlos rápidamente, facilitando su instalación en entornos con muchos equipos.

2.2.10. Referencias del manual:

http://cracklib.sourceforge.net, Página oficial del proyecto Cracklib, 5 de mayo de 2011.

http://packages.debian.org/sid/wordlist, Descripción de la lista de paquetes de listas de palabras, 6 de mayo de 2011.

http://www.deer-run.com/~hal/sysadmin/pam_cracklib.html, Documentación del cracklib por Hal Pomeranz, 6 de mayo de 2011

http://www.linuxsecurity.com/resource_files/host_security/securing-debian-howto/ch4.en.html, Buenas prácticas de seguridad (hardening) en sistemas GNU/LINUX, 1 de mayo de 2011.

http://manpages.ubuntu.com/manpages/natty/man8/cracklib-format.8.html, Manual de cracklib en la distribución Ubuntu, 1 de mayo de 2011

http://www.43things.com/entries/view/1775650, Configuración de diccionarios de cracklib en Ubuntu, 1 de mayo de 2011.

http://www.ubuntu-es.org/node/533, Descripción de la herramienta apt-get para manejo de paquetes por repositorios, 1 de mayo de 2011

http://www.rsa.com/rsalabs/node.asp?id=2176, Descripción de algoritmo de digestión (hash) por los Laboratorios RSA, 1 de diciembre de 2011.

2.3. Programa para supervisar defacements en RED-UNAM

En este apartado se explica el funcionamiento, propósito y configuración de dos programas para supervisar defacements en RED-UNAM escritos en lenguaje Bash Shell. También se incluye el código fuente de cada programa.

2.3.1. Introducción

Los defacements son modificaciones no autorizadas que se realizan en las páginas web, habitualmente, por grupos de intrusos informáticos por motivos diversos. Normalmente, este tipo de ataques tienen éxito cuando el servidor web afectado posee alguna vulnerabilidad en aplicaciones, servicios o por el uso de contraseñas débiles. Un defacement tiene un efecto adverso en la reputación y la imagen de la organización afectada. En el caso concreto de la UNAM, no sólo se afecta la imagen de la dependencia involucrada, sino en general la de la UNAM.

Con el propósito de detectar este tipo de incidentes en RED-UNAM, desarrollé un programa que facilita su detección.

2.3.2. Objetivo del proyecto

Supervisar automáticamente los defacements publicados del dominio unam.mx en el dominio Zone-h para que personal de respuesta a incidentes de UNAM-CERT contacte al responsable del servidor afectado y se solucione el incidente.

2.3.3. Desarrollo

Zone–h es una organización dedicada desde el 2002 al análisis y supervisión de sitios web que hayan sido víctimas de defacements. A través de su portal web, permiten realizar consultas de defacements por domino y fecha. Esto permite a miembros del equipo de respuesta a incidentes buscar reportes de defacements en dominios unam.mx, sin embargo, no permite realizar búsquedas de manera automática mediante un programa. Por este motivo, desarrollé un programa en lenguaje Bash Shell para consultar diariamente la página y enviar un correo a la cuenta incidentes@seguridad.unam.mx, así los miembros del equipo de respuesta a incidentes podrían contactar a los administradores responsables del equipo afectado lo más pronto posible conteniendo y erradicando el problema.

El primer programa que desarrollé envía al sitio web de Zone-h el dominio y la fecha que se desea consultar. Zone-h trunca la dirección web de los dominios que despliega en la página desplegada, por lo que no se puede obtener la URL de la página afectada directamente del código HTML, sin embargo incluye una dirección URL por cada registro de defacement. Por este motivo fue necesario acceder a cada una de estas páginas para revisar si la URL reportada pertenecía al dominio unam.mx.

Por ejemplo, el código HTML que regresa el sitio de Zone-h después de realizar la consulta es el siguiente:

```
</tr>
onMouseOver="this.className='highlight'"
onMouseOut="this.className='normal'">
<td>2011/05/06</td>
<td><a
href="/archive/notifier=PHG">PHG</a></td>
   <td>H</td>
   <td><a
href="/archive/ip=<Dirección IP>">M</a></td>
   <td></td>
   <td></td>
   <td>www.sitioweb.com/con/URL/larga/truncada
   </td>
   <td>Linux</td>
   <td><a
href="/mirror/id/<No. de mirror>">mirror</a></td>
</tr>
```

El primer programa descarga la página principal y luego obtiene las URLs de las páginas de espejo (mirror) para obtener la URL completa de cada sitio en donde se observó un defacement. Luego realiza una nueva petición a cada URL para enviar un correo a incidentes@seguridad.unam.mx con la URL, la fecha y hora de detección. A continuación se incluye el código fuente del programa antes descrito:

```
#!/bin/bash
htmlIndice="htmlGeneral.html"
archivoMirrors="archivoMirrors.html"
ruta="/home/dbernal/Defacements/"
dia=`date +%d`
mes=`date +%m`
anio=`date +%Y`
dominio=unam.mx

perl "$ruta"socket http://www.zone-
h.org/archive?filter_date_select=exact&domain="$dominio"&filter_date_d="$dia"&fil
ter_date_m="$mes"&filter_date_y="$anio"" > "$ruta""$htmlIndice"
grep ".*href=.*id.*" "$ruta""$htmlIndice" | awk 'BEGIN{FS="\""};{print $2}' |
grep -i mirror > "$ruta""$archivoMirrors"
totalMirrors=`wc -l "$ruta""$archivoMirrors" | cut -d ' ' -f1`
linea=1
while(( $linea <= $totalMirrors ));do

  mirror=`sed -n "$linea,$linea"p "$ruta""$archivoMirrors"`
  pag=`perl "$ruta"socket "http://www.zone-h.org"$mirror"" | grep Domain: |
head -n 1 | sed "s/^.*\">//g" | sed "s/<\/a>.*$//g"`
  echo "Se ha detectado un defacement en la página: "$pag" el $dia del $mes de
$anio a las `date +%k:%M` " > "$ruta"tempZone.txt

  cat "$ruta"tempZone.txt >> "$ruta"Salida.txt
```

```
     mail incidentes@seguridad.unam.mx -s "Defacement-Detectado UNAM" < "$ruta"tempZone.txt
     let linea++

   done
   # El archivo tempZone.txt no se crea si no se encuentran defacements, así que se manda a /dev/null el mensaje de error si se intenta borrar y no existe
   if [ ! -f "$ruta"tempZone.txt ];then

     echo "El $dia del $mes de $anio no se detectaron defacements en el dominio $dominio a las `date +%k:%M`" | mail incidentes@seguridad.unam.mx -s "No se detectaron defacements en el dominio $dominio" 2>/dev/null
     echo "El $dia del $mes de $anio no se detectaron defacements en el dominio $dominio a las `date +%k:%M`" >> "$ruta"Salida.txt

   fi
```

Desafortunadamente, a finales del año pasado la organización Zone-h colocó un captcha en su sitio web con el fin de evitar el uso de programas automáticos para búsqueda de información, lo que ya no permite que el programa anterior funcione adecuadamente. Por ese motivo, desarrollé otro programa que realiza consultas al RSS feed [2], única sección de la página en donde actualmente se puede obtener información sin tener que resolver el captcha. En el RSS, Zone-h publica cada cinco minutos los últimos 20 defacements detectados. A continuación, se incluye el código fuente de este programa:

```
#!/bin/bash

user=dbernal
ruta=/home/$user/Defacements/zonehCapcha/defaceUnam
touch $ruta/defaceOld

while [ 1 ]; do
wget -O $ruta/defacements http://www.zone-h.org/rss/defacements
grep title $ruta/defacements | egrep -v "Zone-H.org Defacements" | sed 's/^.*\[//g' | sed 's/].*$//g' | egrep ".unam.mx($|/)" | sort > $ruta/defaceUnam

comm -13 $ruta/defaceOld $ruta/defaceUnam > $ruta/defaceNew
for i in $(cat $ruta/defaceNew); do

      echo $i;

   echo "Se ha detectado un defacement en la pagina $i. Fecha del servidor: $(date)" | mail -s Defacement-UNAM incidentes@seguridad.unam.mx
   echo "Se ha detectado un defacement en la pagina $i. Fecha del servidor: $(date)" >> /var/log/defacementUNAM

done
sleep 60 #
mv $ruta/defaceUnam $ruta/defaceOld
done

rm "$ruta""$htmlIndice" "$ruta""$archivoMirrors" "$ruta"tempZone.txt 2>/dev/null
```

2.3.4. Documentación del proyecto

Se realizó un archivo README con el fin de que personal de UNAM-CERT pudiera entender y modificar el script en caso de que fuera necesario, así como especificar a grandes rasgos su funcionamiento.

Para realizar el seguimiento de la aplicación, se creó el log `/var/log/defacementUNAM`, de manera que pudiera verificarse el buen funcionamiento de la aplicación.

Para asegurar que el script se ejecutara después de la carga del sistema operativo, agregué la siguiente línea al archivo `/etc/rc.local`:

```
/rutaDelScript/defacementUNAM.sh &
.exit
```

2.3.5. Conclusiones del manual

El programa para supervisar los defacements por medio del feed RSS permite obtener rápidamente las direcciones de las páginas web afectadas que pertenecen al dominio unam.mx y, por lo tanto, dar una atención rápida al incidente. Si se desea supervisar un dominio diferente al de unam.mx, es posible hacerlo con una sencilla modificación del programa.

2.3.6. Referencias del manual

http://www.zone-h.org , Página oficial del sitio Zone H, 2 de mayo de 2011

2.4. Manual de implementación de passwdqc en sistemas LINUX

2.4.1. Introducción

Durante el tiempo que me desempeñé en el área de respuesta a incidentes en UNAM-CERT, observé que la causa más común de los accesos no autorizados a servidores LINUX de RED-UNAM fue el uso de contraseñas débiles. A pesar de haber realizado el manual de cracklib para mitigar este problema, su falta de documentación y complejo manejo de créditos dificultaron su implementación en RED-UNAM, por ello busqué un módulo parecido que ofreciera funcionalidades no presentes en cracklib.

Passwdqc es un conjunto de herramientas que permite implementar una política de verificación y uso de contraseñas robustas, el cual incluye un módulo de PAM pam_passwdqc, los programas por línea de comandos pwqcheck, pwqgen y la biblioteca libpasswdqc.

Pwqcheck y pwqgen son aplicaciones independientes para comprobar la robustez de las contraseñas o crear frases de contraseñas (passphrase) respectivamente y pueden ser llamadas desde scripts. Passwdqc tiene las siguientes características:

- Cuenta con el módulo Pam_passwdqc, éste puede ser agregado al archivo de autenticación de los sistemas con PAM para realizar las verificaciones correspondientes.
- Ofrece funcionalidades no presentes en cracklib, el cual es una de las alternativas que existen para mejorar la robustez de las contraseñas en sistemas basados en UNIX. En la tabla 2.3 se realiza una comparación entre estos dos módulos.
- *Fácil instalación* –Reduce el tiempo necesario para instalar la herramienta en un entorno amplio con varios servidores.
- *Gratuito* – Favorece su uso, dada la dificultad de muchas dependencias de la UNAM para obtener recursos económicos.
- *Bien documentado* – Soportado por la organización de seguridad, en la página oficial del proyecto se ofrecen documentos y completas referencias que indican cómo instalar el módulo en plataformas específicas. Además se proveen páginas de manuales en la instalación que permiten configurar el módulo.
- *Flexible* – Permite implementar una política específica de contraseñas.
- *Soportado* – La herramienta cuenta con una comunidad de desarrollo que constantemente lo mejora y actualiza.

En la tabla 2.3 se realiza una comparación entre los módulos cracklib y passwdqc.

Passwdqc	*Cracklib*
Permite flexiblemente asegurar una política de contraseñas.	Permite integración con la opción `remember`=x del módulo `pam_unix`, lo cual evita que un usuario repita una contraseña usada previamente.
Permite el uso de frases de paso, además de contraseñas.	Permite el uso de diccionarios personalizados.
Permite definir una longitud máxima para las contraseñas.	
Cada que un usuario es obligado a cambiar su contraseña, se le muestran las reglas que la nueva contraseña o clave de paso debe cubrir.	
Está bien documentado tanto en la página oficial del proyecto Openwall, en páginas man disponibles en el sistema y en páginas de distribuciones específicas.	
Es posible utilizarlo en una gran variedad de sistemas basados en UNIX, como LINUX, Solaris, BSD y HP-UX	

Tabla. 2.3 Comparación entre cracklib y passwdqc

2.4.2. Objetivo del manual

Proporcionar una guía para instalar, configurar y probar la biblioteca `pam_passwdqc` la cual permite establecer una política de contraseñas robustas en los sistemas LINUX y en otros sistemas basados en UNIX que cuenten con PAM.

Brindar instrucciones específicas para instalar y configurar passwdqc en los sistemas operativos Debian versiones 5 y 6, Ubuntu versión 10-10, Fedora versión 14 y OpenSolaris versión 11.4. Se escogieron estas versiones porque en el momento de elaboración de este manual son las más recientes de esos sistemas operativos.

2.4.3. Advertencia

La instalación, configuración y correcto funcionamiento de las herramientas indicadas en este manual fueron verificados en máquinas virtuales con configuración predeterminada. Cualquier cambio realizado en un sistema, tiene la probabilidad de causar un mal funcionamiento. A fin de reducir la probabilidad de que esto ocurra, se sugiere establecer un ambiente de pruebas configurado según las necesidades de la dependencia antes de la instalación en servidores de producción.

2.4.4. Parámetros de configuración

Las opciones más importantes se describen a continuación, para una referencia más completa se puede consultar el archivo README incluido en el código fuente de la biblioteca passwdqc o bien los manuales de passwdqc, disponibles una vez que el kit passwdqc ya se encuentra instalado, mediante los comandos: `man pam_passwdqc` y `man passwdqc.conf`

config = FILE

Define un archivo de configuración (disponible en la versión más reciente 1.2.0-1).

min=N0,N1,N2,N3,N4

Uno de los criterios que definen la robustez de las contraseñas es que los caracteres que la conformen pertenezcan a diferentes grupos. Passwdqc considera cuatro grupos o clases de caracteres: mayúsculas, minúsculas, números y otros símbolos. Los valores de *N0,N1;N2,N3,N4* de la opción min permiten definir las longitudes mínimas para las contraseñas formadas por caracteres de un grupo, dos grupos, frases de paso, tres grupos y cuatro grupos respectivamente.

Alternativamente, se podrá deshabilitar una clase en particular con la palabra reservada *disabled si no se quiere aceptar contraseñas de ese tipo*. Los números posteriores no pueden tener una longitud mayor que el número anterior. P. ej. `min=disabled,disabled,8,8,9` no se permite.

Ejemplo

- Permitir contraseñas con caracteres de todas las clases con longitud mínima de 10 caracteres o contraseñas cono caracteres de tres clases con una longitud mínima de 12.
- Permitir contraseñas con una longitud máxima de 14 caracteres.
- Asegurar que tanto los usuarios como el usuario root cumplan las reglas.
- Las contraseñas similares a las anteriores no deben ser autorizadas.

En la figura 2.10 se muestra la configuración de la política en el archivo de PAM.

```
#password   required   pam_unix.so nullok obscure md5
password required pam_passwdqc.so min=disabled,disabled,disabled,12,10 enforce=everyone max=14 similar=deny

# Alternate strength checking for password. Note that this
# requires the libpam-cracklib package to be installed.
# You will need to comment out the password line above and
# uncomment the next two in order to use this.
# (Replaces the `OBSCURE_CHECKS_ENAB', `CRACKLIB_DICTPATH')
#
# password required       pam_cracklib.so retry=3 minlen=6 difok=3
# password required       pam_unix.so use_authtok nullok md5
```

Fig. 2.10 Implementación de política de contraseñas

Las clases se definen como sigue:
N0 – Usada para contraseñas de una sola clase.

N1 – Usada para contraseñas de dos clases que no reúnen los requisitos para ser consideradas una frase de paso.

N2 – Usada para frases de paso. Además de este requisito, las frases de paso también deben tener un número suficiente de palabras.

N3 y N4 – Usada para contraseñas de tres y cuatro grupos respectivamente.
Cuando se calcula el número de contraseña, la mayúscula al inicio de la palabra y un dígito al final de la contraseña no son tomadas en cuenta, al ser algo muy común, proporciona poca robustez a la contraseña.

Se debe observar que la siguiente regla se cumpla, de lo contrario fallará la llamada a la biblioteca pam_passwdqc: N0<=N1<=N3<=N4

Para no permitir el uso de alguno de los casos anteriores, se puede especificar la opción disabled.

max – La longitud máxima de la contraseña. Si el valor es max=8, la contraseña no se rechazará sino que se truncará a 8 caracteres.

passphrase - Número de palabras aceptadas en las frases de paso, 0 para deshabilitar que los usuarios escojan las frases (serían sugeridas por el sistema).

match – Establece los criterios para buscar palabras comunes en la contraseña.
Numero

enforce – Define si se obliga a los usuarios o también al usuario root a escoger una contraseña robusta. Si no se obliga, sólo advierte, pero permite el uso de contraseñas débiles.

similar - Permite o rechaza contraseñas parecidas a la anterior. Quita la contraseña común y revisa la robustez de la cadena restante.

retry – Cantidad de veces que se le permite introducir otra contraseña al usuario después de que escoge una que no cumple con las reglas.

En caso de que alguno de los parámetros no se especifiquen, tomarán los valores predeterminados que pueden consultarse en el archivo README.

2.4.5. Instalación

La instalación con paquetes pre compilados o repositorios es generalmente más estable, ya que resuelve automáticamente las dependencias y es menos susceptible a errores. La instalación por código fuente, por lo general, es más tardada y complicada, sin embargo permite en algunas ocasiones obtener versiones más recientes que los binarios precompilados y una mayor flexibilidad, pues es posible modificar el código fuente por el que sea requerido por nuestra organización.

2.4.6. Instalación específica por sistema operativo

A continuación se incluyen instrucciones específicas de instalación y configuración de passwdqc en Debian5, Debian 6, Ubuntu 10.10, Fedora 14 y OpenSUSE 11.4.

2.4.6.1. Sistema Operativo GNU/LINUX Debian versión 5

Instalación por repositorios

Actualizar los repositorios

```
# apt-get update
# apt-get install libpam-passwdqc
```

Instalación por código fuente

Descargar la versión estable más reciente de passwdqc, en la fecha de elaboración es la 1.2.2

```
# wget http://www.openwall.com/passwdqc/passwdqc-1.2.2.tar.gz
```

Descomprimir el archivo

```
# tar –xzf passwdqc-1.2.2.tar.gz
```

Ubicarse en el directorio del kit passwdqc

```
# cd passwdqc-1.2.2
```

Opcional:

Traducir al español los mensajes de texto de los archivos pam_ passwdqc.c y passwd_check.c para que las instrucciones sean mostradas en español cada que sea utilizada la biblioteca pam_passwdqc.so, por ejemplo, cuando se cambie una contraseña.

A continuación se incluyen los mensajes traducidos al español de los archivos pam_passwdqc.c y passwd_check.c, los cuales pueden usarse para reemplazar los mensajes en inglés de estos archivos antes de compilar la biblioteca.

pam_passwdqc.c

```
#define PROMPT_OLDPASS \
        "Introduce la contraseña actual: "
#define PROMPT_NEWPASS1 \
        "Introduce la nueva contraseña: "
#define PROMPT_NEWPASS2 \
        "Confirma la nueva contraseña: "
#define MESSAGE_MISCONFIGURED \
        "Error de configuración del sistema. Por favor, contacta al administrador."
#define MESSAGE_INVALID_OPTION \
        "pam_passwdqc: %s."
#define MESSAGE_INTRO_PASSWORD \
        "\n Ahora puedes escoger la nueva contraseña.\n"
#define MESSAGE_INTRO_BOTH \
        "\n Ahora puedes escoger la nueva contraseña o la clave de paso.\n"
#define MESSAGE_EXPLAIN_PASSWORD_1CLASS \
        "Una buena contraseña debe contener minúsculas, mayúsculas,\n" \
        "números y símbolos especiales. Puedes usar una contraseña de %s %d caracteres de
longitud.\n"
#define MESSAGE_EXPLAIN_PASSWORD_CLASSES \
        "Una buena contraseña debe contener minúsculas, mayúsculas,\n" \
        "números y símbolos especiales. Puedes usar una contraseña de %s %d \n" \
        "caracteres de longitud con caracteres de al menos %d de estas clases.\n" \
        "Una mayúscula inicial en la contraseña y un número al final no son\n" \
```

```
        "tomados en cuenta en el número de clases utilizadas.\n"
#define MESSAGE_EXPLAIN_PASSWORD_ALL_CLASSES \
        "Una contraseña válida debe ser una mezcla de minúsculas, mayúsculas,\n" \
        "números y otros caracteres. Puedes usar una contraseña de %s %d caracteres de longitud\n" \
        "de todas estas clases.\n" \
        "Una mayúscula inicial en la contraseña y un número al final no son\n"\
        "tomados en cuenta en el número de clases utilizadas.\n"
#define MESSAGE_EXPLAIN_PASSWORD_ALT \
        "Una contraseña válida debe estar compuesta por las siguientes clases: mayúsculas, minúsculas,\n" \
        "números y símbolos especiales. Puedes usar una clave de %s %d caracteres de \n" \
        "longitud formada por caracteres de al menos 3 de las 4 clases o\n" \
        "una clave de %s %d caracteres de longitud formada por caracteres las 4 clases\n" \
        "Una mayúscula inicial en la contraseña y un número al final no son\n"\
        "tomados en cuenta en el número de clases utilizadas.\n"
#define MESSAGE_EXPLAIN_PASSPHRASE \
        "Una frase de paso debe estar formada por al menos %d palabras\n" \
        "debe tener una longitud de %d a %d caracteres y contener suficientes\n" \
        "caracteres diferentes\n"
#define MESSAGE_RANDOM \
        "Alternativamente, si nadie puede ver tu terminal ahora, puedes \n" \
        "escoger la siguiente palabra como tu contraseña: \"%s\".\n"
#define MESSAGE_RANDOMONLY \
        "Este sistema está configurado para permitir las contraseñas creadas aleatoriamente\n" \
        "Si nadie más puede ver tu terminal ahora, puedes escoger la siguiente palabra como tu contraseña\n" \
        ": \%s\". Si no, puedes realizar este procedimiento después.\n"
#define MESSAGE_RANDOMFAILED \
        "Este sistema está configurado para generar contraseñas aleatoriamente\n" \
        "pero el intento de generar una contraseña ha fallado. Esto \n" \
        " puede deberse a varias razones, por ejemplo, que se haya \n" \
        "solicitado una longitud de contraseña no viable o que el acceso al \n" \
        "generador de números aleatorios del kernel no esté disponible. \n"
#define MESSAGE_TOOLONG \
        "Esta contraseña es demasiado larga para algunos servicios, escoge otra."
#define MESSAGE_TRUNCATED \
        "Advertencia, tu contraseña será truncada a 8 caracteres."
#define MESSAGE_WEAKPASS \
        "Contraseña débil: %s."
```

```
#define MESSAGE_NOTRANDOM \
        "Error, usted no ha escrito correctamente la contraseña que se le ha generado."
#define MESSAGE_MISTYPED \
        "Error, las contraseñas no coinciden."
#define MESSAGE_RETRY \
        "Favor de intentar nuevamente."
```

passwdqc_check.c

```
#define REASON_ERROR \
        "Falló la verificación"

#define REASON_SAME \
        "Es la misma que la anterior"

#define REASON_SIMILAR \
        "Se basa en la anterior"

#define REASON_SHORT \
        "Es demasiado corta"

#define REASON_LONG \
        "Es demasiado larga"

#define REASON_SIMPLESHORT \
        "No se proporcionaron suficientes clases o caracteres diferentes tomando en cuenta
la longitud"

#define REASON_SIMPLE \
        "No se proporcionaron suficientes clases o caracteres diferentes"

#define REASON_PERSONAL \
        "Se basa en información personal"

#define REASON_WORD \
        "Se basa en una palabra de diccionario"

#define REASON_SEQ \
        "Se basa en una secuencia común y no en una frase de paso"
```

Instalar los archivos de desarrollo para PAM

```
apt-get install libpam0g-dev
```

Instalación del kit passwdqc

```
make && make install
```

Actualización de bibliotecas compartidas.

```
ldconfig
```

2.4.7. Configuración

En el archivo `/etc/pam.d/common-password` comentar la línea siguiente:

```
Password required pam_unix.so nullok obscure md5
```

Y agregar las siguientes líneas:

```
password required pam_passwdqc.so
password [success=1 default=ignore] pam_unix.so obscure use_authtok try_first_pass md5
password required pam_permit.so
```

Con esta configuración se indica que el primer módulo en recibir la contraseña y realizar la verificación de la misma es passwdqc, en caso de autorizarla, la pasa al módulo pam_unix, responsable de actualizar el registro correspondiente en el archivo `/etc/shadow`. Por último, se utiliza el módulo `pam_permit` para salir con éxito de la llamada a PAM.

La configuración anterior utiliza valores predeterminados del módulo passwdqc. Para definir valores diferentes, se puede incluir a la derecha del nombre del módulo el nombre del parámetro seguido del signo de igual y el valor del mismo. Los parámetros deben separarse por espacio o tabulador.

En la figura 2.11 se observa la implementación con passwdqc en el archivo `/etc/pam.d/passwdqc` de la siguiente política en el archivo de PAM:

- Permitir contraseñas con caracteres de todas las clases con longitud mínima de 10 caracteres o contraseñas de tres clases con una longitud mínima de 12.
- Permitir contraseñas con una longitud máxima de 14 caracteres.
- Asegurar que tanto los usuarios como el usuario root cumplan las reglas y que las contraseñas similares a las anteriores no sean autorizadas.

```
#password   required   pam_unix.so nullok obscure md5
password required pam_passwdqc.so min=disabled,disabled,disabled,12,10 enforce=everyone max=14 similar=deny

# Alternate strength checking for password. Note that this
# requires the libpam-cracklib package to be installed.
# You will need to comment out the password line above and
# uncomment the next two in order to use this.
# (Replaces the `OBSCURE_CHECKS_ENAB', `CRACKLIB_DICTPATH')
#
# password required       pam_cracklib.so retry=3 minlen=6 difok=3
# password required       pam_unix.so use_authtok nullok md5
```

Fig. 2.11 Implementación de política de contraseñas

Se puede probar la configuración y funcionamiento de passwdqc con un programa como passwd. En este caso, pam_passwdqc muestra las reglas que deben seguirse para tener una contraseña robusta. En la figura 2.12 se muestra la ejecución del comando passwd con la configuración de PAM mostrada en la figura 2.11. En este caso se modificaron los mensajes del inglés al español durante la instalación por código fuente.

```
debian:/home/prueba/Desktop/passwdqc-1.2.2# passwd prueba2

Ahora puedes escoger la nueva contraseña.

Una contraseña válida debe estar compuesta por las siguientes clases: mayúsculas, minúsculas,
números y símbolos especiales. Puedes usar una clave de  12 caracteres de
longitud formada por caracteres de al menos 3 de las 4 clases o
una clave de  10 caracteres de longitud que tenga caracteres de las 4 clases
Una mayúscula inicial en la contraseña y un número al final no son
tomados en cuenta en el número de clases utilizadas.

Introduce la nueva contraseña:
```

Fig. 2.12 Ejecución del comando passwd

En caso introducir una contraseña no aceptada, se podrían mostrar algunos de los mensajes mostrados en la figura 2.13, dependiendo del caso:

```
contraseña Debil: demasiado corta.
Favor de intentar nuevamente.

contraseña Debil: no se proporcionaron suficientes clases o caracteres diferentes.
Favor de intentar nuevamente.

Esta contraseña es demasiado larga para algunos servicios, escoge otra.
```

Fig. 2.13 Mensajes de error de la biblioteca passwdqc

2.4.7.1. Sistemas Operativos GNU/LINUX Ubuntu versión 10.10 y Debian versión 6

```
apt-get install libpam-passwdqc
```

En este caso, al instalar el paquete por repositorios, automáticamente se modifica la configuración del archivo`/etc/pam.d/common-password`, por lo que sólo basta instalarlo y agregar los parámetros deseados.

La ejecución de la instalación de la biblioteca por repositorios en Ubuntu 10.10 se muestra en la figura 2.14:

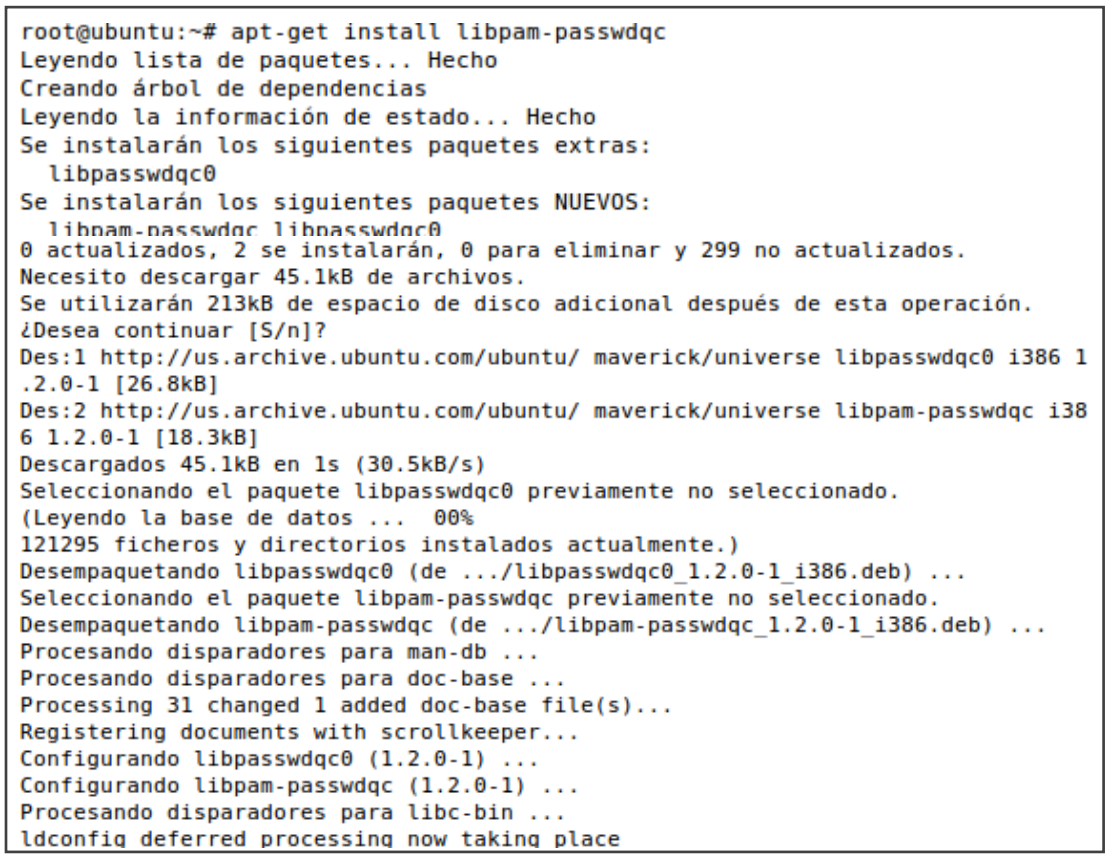

Fig. 2.14 Instalación de libpam-passwdqc por repositorios en Ubuntu 10.10

Después de instalar passwdqc por repositorios, automáticamente se modifica el archivo `/etc/pam.d/common-password`, como se observa en la figura 2.15, por lo que sólo falta definir los parámetros deseados directamente en ese archivo o mediante un archivo de configuración si se desean usar parámetros diferentes a los predeterminados.

```
# See the pam_unix manpage for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password        requisite                       pam_passwdqc.so
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok
try_first_pass sha512
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
password        optional        pam_gnome_keyring.so
# end of pam-auth-update config
```

Fig. 2.15 Archivo de configuración de PAM en Ubuntu 10.10 después de instalar libpam-passwdqc.

Instalación mediante código fuente.

Las instrucciones de instalación son iguales que para el sistema operativo GNU/LINUX Debian versión 5.

Configuración

Se deberá modificar el archivo `/etc/pam.d/common-password` como se muestra en la figura 2.16.

```
#
# See the pam_unix manpage for other options.

# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
password        requisite                       pam_passwdqc.so
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok
try_first_pass sha512
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
password        optional        pam_gnome_keyring.so
# end of pam-auth-update config
root@ubuntu:~#
```

Fig. 2.16 Configuración del modulo passwdqc con opciones predeterminadas

Finalmente, si se desean utilizar parámetros diferentes a los preestablecidos, se puede definir la configuración mediante un archivo (opción sólo disponible para la instalación por código fuente) o directamente en el archivo `/etc/pam.d/common-password`. En este manual se hará mediante un archivo de configuración ubicado en `/etc/passwdqc.conf`, el cual puede tener los parámetros mostrados en la figura 2.17 (aquellos que no se indiquen tomarán los valores predeterminados):

```
min=disabled,disabled,disabled,10,8
max=12
similar=deny
match=4
enforce=everyone
```

Fig. 2.17 Archivo de parámetros de passwdqc

Ahora sólo falta indicar en el archivo de de autenticación de PAM la ubicación del archivo de configuración de parámetros de passwdqc, como se aprecia en la figura 2.18.

```
# here are the per-package modules (the "Primary" block)
password        requisite pam_passwdqc.so config=/etc/passwdqc.conf
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok try
```

Fig. 2.18 Configuración de parámetros de passwdqc mediante archivo.

Se verifica el correcto funcionamiento al cambiar la contraseña de un usuario como se muestra en la figura 2.19.

```
root@ubuntu:~/Escritorio/passwdqc-1.2.2# passwd prueba2

Ahora puedes escoger la nueva contraseña.

Una contraseña válida debe estar compuesta por las siguientes clases: mayúsculas, minúsculas,
números y símbolos especiales. Puedes usar una clave de  10 caracteres de
longitud formada por caracteres de al menos 3 de las 4 clases o
una clave de n 8 caracteres de longitud que tenga caracteres de las 4 clases
Una mayúscula inicial en la contraseña y un número al final no son
tomados en cuenta en el número de clases utilizadas.

Introduce la nueva contraseña:
Confirma la nueva contraseña:
passwd: contraseña actualizada correctamente
root@ubuntu:~/Escritorio/passwdqc-1.2.2#
```

Fig. 2.19 Ejecución del comando passwd con la configuración mostrada en las figuras 2.15 y 2.16.

2.4.7.2. Sistema operativo GNU/LINUX Fedora versión 14

Cracklib se encuentra instalado en la instalación predeterminada. También se encuentra compilado y disponible el módulo passwdqc. Para habilitarlo habrá que deshabilitar cracklib y habilitar y configurar passwdqc en el archivo `/etc/pam.d/system-auth`, el cual es una liga simbólica al archivo `/etc/pam.d/system-auth-ac`, el cual contiene, entre otras directivas, las de autenticación del sistema (password).

A diferencia de los sistemas basados en Debian, como Ubuntu, Fedora cuenta con una utilería llamada authconfig utilizada para modificar opciones de configuración en el sistema, incluyendo el archivo `/etc/pam.d/system-auth-ac`. Si éste es modificado manualmente, los cambios serán destruidos la siguiente vez que se ejecute authconfig, a menos que se indique en el propio archivo de configuración de authconfig (`/etc/sysconfig/authconfig`) que se ha realizado este cambio.

```
# vi /etc/sysconfig/authconfig
```

De manera predeterminada se tiene habilitado cracklib y deshabilitado passwdqc, como se muestra en la figura 2.20.

```
USEMKHOMEDIR=no
USEPAMACCESS=no
CACHECREDENTIALS=yes
USESSSDAUTH=no
USESHADOW=yes
USEWINBIND=no
USEDB=no
FORCELEGACY=no
USEFPRINTD=yes
FORCESMARTCARD=no
PASSWDALGORITHM=sha512
USELDAPAUTH=no
USEPASSWDQC=no
USELOCAUTHORIZE=yes
USECRACKLIB=yes
USEWINBINDAUTH=no
USESMARTCARD=no
USELDAP=no
USENIS=no
USEKERBEROS=no
USESYSNETAUTH=no
USESSSD=no
USEHESIOD=no
```

Fig. 2.20 Archivo de configuración de auth-config en Fedora.

Simplemente hay que modificar las siguientes líneas:

```
USEPASSWDQC=yes
USECRACKLIB=no
```

Ahora, se requiere modificar la configuración del archivo `/etc/pam.d/system-auth-ac`, el cual utiliza de manera predeterminada el módulo cracklib, como se muestra en la figura 2.21

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_fprintd.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3 type=
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session     optional      pam_systemd.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
```

Fig. 2.21 Archivo de configuración de PAM en Fedora.

En este archivo se comenta la línea de cracklib y se agrega la línea de passwdqc en donde se puede definir la configuración deseada, como se muestra en la figura 2.22.

```
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_fprintd.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account     required      pam_permit.so

#password requisite pam_cracklib.so try_first_pass retry=3 type=
password    requisite     pam_passwdqc.so min=disabled,disabled,disabled,10,8 max=12 enforce=everyone
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
-session     optional      pam_systemd.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet use_uid
session     required      pam_unix.so
~
```

Fig. 2.22 Archivo de configuración de PAM en Fedora.

El la figura 2.23 se observa la ejecución de passwdqc.

```
[root@localfedora pam.d]# passwd prueba2
Changing password for user prueba2.

You can now choose the new password.

A valid password should be a mix of upper and lower case letters,
digits, and other characters.  You can use a 10 character long
password with characters from at least 3 of these 4 classes, or
an 8 character long password containing characters from all the
classes.  An upper case letter that begins the password and a
digit that ends it do not count towards the number of character
classes used.

Enter new password:
Re-type new password:
passwd: all authentication tokens updated successfully.
```

Fig. 2.23 Archivo de configuración de PAM en Fedora.

Como paso opcional y sólo para asegurar que authconfig no modifique la configuración de passwdqc cuando se utilice, se ejecuta el siguiente comando, el cual no debe indicar inconsistencias:

```
# authconfig –updateall
```

2.4.7.3. Sistema Operativo Open Suse versión 11.4

El archivo de configuración de PAM en este sistema se encuentra en */etc/pam.d/common.password* y se puede editar directamente, pero la manera recomendada de editarlo es mediante una utilidad llamada pam-config usada para agregar o quitar módulos de PAM en el sistema. De manera predeterminada el sistema usa un módulo llamado pam_pwcheck.so, el cual usa a su vez a la biblioteca cracklib para realizar verificaciones adicionales en las contraseñas. Al igual que cracklib, este módulo no permite implementar una política precisa de contraseñas en el sistema, por lo que lo reemplazaremos con passwdqc. En la figura 2.24 se aprecia parte del archivo de autenticación de PAM en este sistema operativo.

```
requisite       pam_pwcheck.so  nullok cracklib
optional        pam_gnome_keyring.so    use_authtok
required        pam_unix2.so    use_authtok nullok
```

Fig. 2.24 Módulo de autenticación pwcheck en Open Suse versión 11.4.

De manera predeterminada el módulo passwdqc no está compilado y disponible, por lo que hay que instalarlo, lo cual se puede realizar mediante código fuente.

```
# wget http://www.openwall.com/passwdqc/passwdqc-1.2.2.tar.gz
# tar –xzf passwdqc-1.2.2.tar.gz
# cd passwdqc-1.2.2
```

Opcional
Traducir los mensajes en inglés de los archivos pam_passwdqc.c y passwdqc_check.c
(Este procedimiento es igual al descrito para Debian 5)

Próximamente será posible descargar el código fuente en español desde la sección de contribuciones de la página de passwdqc de Openwall, con lo que no será necesario modificar manualmente el código fuente.

```
# make && make install
# ldconfig
```

Configuración

Siguiendo las indicaciones propias del sistema, se modificará el archivo `/etc/pam.d/common-password` mediante la utilidad pam-config. Para ello, primero se agrega el módulo passwdqc:

```
# pam-config -a --passwdqc
```

Luego se quita el módulo pam_pscheck.so, ya que éste realiza una función análoga a passwdqc y por lo tanto no es necesario.

```
# pam-config -d --pwcheck
```

Mediante la siguiente sintaxis se pueden definir las opciones deseadas de passwdqc:

```
# pam-config –a --passwdqc-<opción>=<valor>
```

En caso de omitir alguna opción, tomará los valores predeterminados.

Ejemplo:

- Permitir contraseñas de tres clases con una longitud mínima de 10 caracteres o una contraseña de cuatro clases con una longitud mínima de 8.
- Obligar a todos los usuarios, incluido root a seguir las reglas.
- Denegar las contraseñas similares a las anteriores.
- Sólo permitir dos intentos para generar la contraseña antes de regresar error.

Se establece la política anterior mediante los siguientes comandos:

```
# pam-config –a –passwdqc-min=disabled,disabled,disabled,10,8
# pam-config –a –passwdqc-enforce=everyone
# pam-config –a –passwdqc-similar=deny
# pam-config –a –passwdqc-retry=2
```

Finalmente, sólo queda abrir el archivo `/etc/pam.d/common-password` para verificar que se aplicaron las opciones correctamente.

2.4.8. Establecer una política de caducidad de contraseñas

De manera predeterminada las contraseñas de usuarios en la mayoría de los sistemas LINUX jamás caducan, lo cual es una mala práctica de seguridad.

Para auditar el periodo de expiración de las contraseñas en el sistema, se puede utilizar el comando chage o bien consultar el parámetro PASS_MAX_DAYS, ubicado en el archivo `/etc/login.defs`

Para forzar al cambio periódico de la contraseña para las nuevas cuentas, se debe definir en el archivo `/etc/login.defs` los siguientes parámetros:

PASS_MAX_DAYS n – Representa el número de días que la contraseña será válida.
PASS_MIN_DAYS n – Número de días que deberán pasar para que el usuario pueda cambiar su contraseña.
PASS_WARN_AGE n – Número de días antes de que expire la contraseña del usuario, tras el cual una advertencia indicándole que su contraseña está por expirar.

Por ejemplo, para forzar que la contraseña sea cambiada cada 90 días y se le advierta al usuario 14 días antes del cambio de su contraseña, se deben establecer los siguientes parámetros:
PASS_MAX_DAYS 90
PASS_WARN_AGE 14
El parámetro PASS_MIN_DAYS se podría dejar en cero si no se quiere hacer esperar a un usuario un tiempo determinado para que pueda cambiar su contraseña.

2.4.9. Eliminar contraseñas débiles del sistema

La biblioteca passwdqc evitará que las contraseñas de las nuevas cuentas de usuario o las contraseñas cambiadas a partir de la fecha de instalación de la biblioteca sean débiles, sin embargo, es posible que las contraseñas de las cuentas creadas antes de la instalación lo sean, por lo que representan un riesgo de seguridad.

Para obligar el cambio inmediato de las contraseñas de las cuentas creadas antes de la configuración de passwdqc se usará el comando chage. Para ello, se definirá el mismo valor de PASS_MAX_DAYS con la opción –M y se especificará con la opción –d una fecha de último cambio de contraseña anterior al valor PASS_MAX_DAYS, para obligar a que la contraseña se cambie la siguiente vez que el usuario inicie sesión. El comando chage y el archivo `/etc/login.defs` es estándar para todos los LINUX manejados en este manual y muchos otros, por brevedad, a continuación sólo se incluyen capturas de pantalla de Fedora 14.

En la figura 2.25 se muestra la configuración inicial de los parámetros de la contraseña del usuario. A continuación, se establece el valor de PASS_MAX_DAYS en 90 días y se establece la fecha de último cambio de contraseña en 1970, con lo que los valores desplegados por el comando chage indican “password must be changed”, es decir, que la contraseña debe cambiarse.

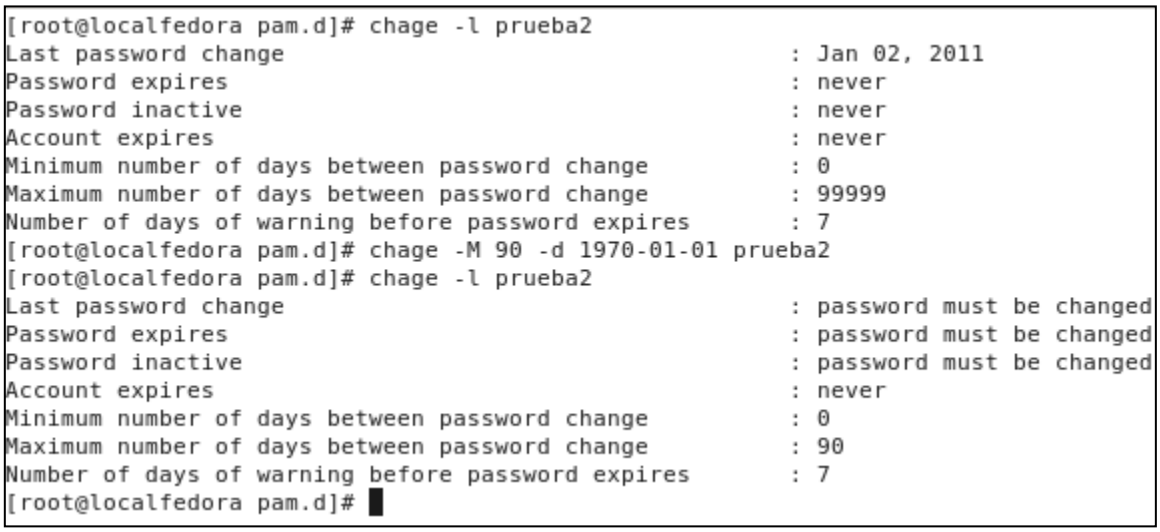

Fig. 2.25 Expiración de contraseñas mediante el comando chage.

Al iniciar sesión, se obliga al usuario a cambiar su contraseña inmediatamente, como se muestra en la figura 2.26.

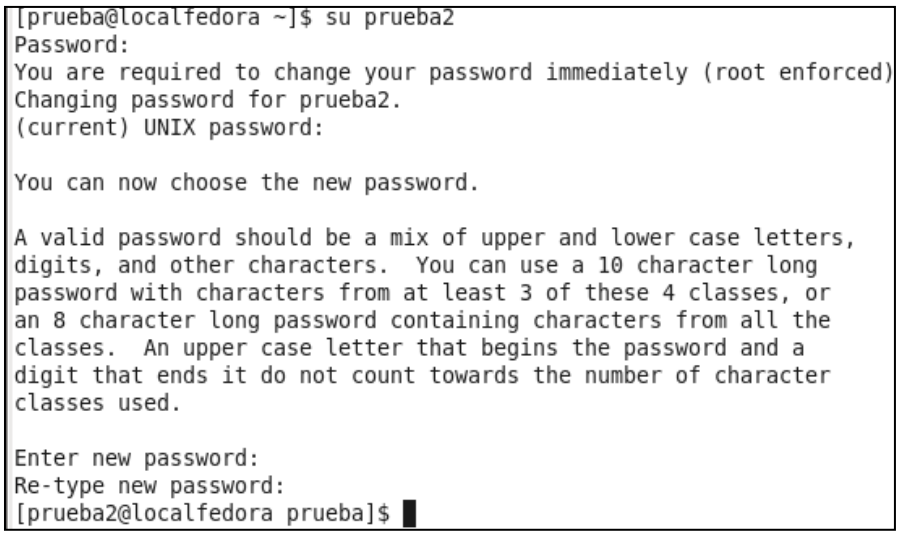

Fig. 2.26 Cambio forzado de contraseña de usuario al iniciar sesión.

A partir de este momento, el usuario deberá cambiar su contraseña cada que pasen la cantidad de días especificados por el parámetro PASS_MAX_DAYS a partir de la fecha del último cambio de contraseña, como se muestra en la figura 2.27.

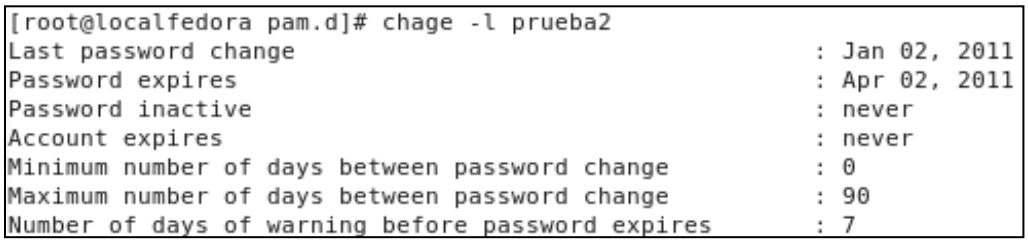

Fig. 2.27 Cambio forzado de contraseña de usuario al iniciar sesión.

El comando `chage` también permite establecer una fecha de expiración acorde al tiempo de uso definido para una cuenta de usuario, con lo que evita el riesgo de inicio de sesión después de este tiempo.

Por ejemplo, si de antemano se sabe que un usuario deberá realizar tareas de mantenimiento durante dos meses, puede definirse mediante `chage` una fecha de expiración acorde con el tiempo que dure esa actividad. En la figura 2.28 se muestra como definir una fecha de expiración para la cuenta de usuario prueba2.

```
[root@localfedora pam.d]# chage -E 2011-03-20 prueba2
[root@localfedora pam.d]# chage -l prueba2
Last password change                                    : Jan 20, 2011
Password expires                                        : Apr 20, 2011
Password inactive                                       : Apr 27, 2011
Account expires                                         : Mar 20, 2011
Minimum number of days between password change          : 0
Maximum number of days between password change          : 90
Number of days of warning before password expires       : 7
```

Fig. 2.28 Uso del comando chage para establecer la fecha de expiración de la cuenta.

2.4.10. Conclusiones del manual

La biblioteca de PAM passwdqc permite mejorar la seguridad en el uso de contraseñas en sistemas operativos basados en LINUX. A diferencia del módulo similar cracklib, permite implementar una política específica de contraseñas, uso de frases de contraseñas, indicar al usuario los requisitos que éstas deben cumplir y obligar al usuario root a cumplir con las verificaciones de contraseñas robustas.

2.4.11. Sugerencias de mejora

Estas configuraciones fueron probadas en ambientes virtuales. Considero que el UNAM-CERT debe encontrar administradores de redes de la UNAM que quieran probar las herramientas para determinar el impacto que tienen en ambientes de producción controlados antes de ser liberados a toda la comunidad.

El CERT planea realizar a futuro manuales de autenticación con LDAP, ya que ese esquema de autenticación centralizada puede brindar ventajas significativas para reducir el esfuerzo de configuración y facilitar la supervisión.

Los manuales presentes en este trabajo, aunados a otros desarrollados por otros miembros de UNAM-CERT y administradores experimentados de RED-UNAM, serán provistos a la comunidad de administradores de la UNAM en el evento y el sitio web de ADMIN-UNAM.

2.4.12. Referencias del manual

http://www.openwall.com/passwdqc, Página oficial del proyecto Passwdqc, 1 de mayo de 2011.
http://linux.die.net/man/1/chage, Referencia del comando chage, 1 de mayo de 2011.
http://linux.die.net/man/5/pam.d, Referencia de los módulos de autenticación de contraseñas (PAM), 2 de mayo de 2011.

REFERENCIAS DEL CAPÍTULO

[1] http://buscon.rae.es, Diccionario de la Real Academia de la Lengua, 7 de mayo de 2011.

[2] http://www.whatisrss.com/, Descripción del Formato RSS, 7 de mayo de 2011.