

# Zona System

Blog personal de Adrián Lois. Comentando un poco de Seguridad Informática y Sistemas

[Inicio](#) [Contacto](#)

19 ABRIL, 2020

## Gestión de políticas de contraseñas en Linux: login.defs y pam\_pwquality (pam\_cracklib)

Antes de entrar en materia y comentar donde se definen las directivas de contraseñas en entornos Linux, es necesario conocer los campos que estructuran al fichero donde almacenan cifradas las contraseñas de usuarios locales **/etc/shadow**.

Este fichero nos muestra el estado actual de como se están aplicando estas directivas a los usuarios y que nos resulta útil en procesos de auditoria interna para conocer esta información.

### Estructura del fichero /etc/shadow



Figura 1: Estructura del fichero /etc/shadow.

- 1. Nombre de usuario.
- 2. Contraseña cifrada. Se establece con la estructura `$id$salt$hashed`. El tipo de algoritmo utilizado se define en el inicio en números entre los símbolos \$.
- \$1\$ - MD5.
- \$2a\$ - Blowfish.
- \$2y\$ - Blowfish.
- \$5\$ - SHA-256.
- \$6\$ - SHA-512.
- 3. Último cambio de contraseña desde el 1/Enero/1970 (**epoch**).
- 4. Cantidad de días restantes para que el usuario cambie su contraseña. -1 significa que nunca expira.
- 5. Número máximo de días que la contraseña es válida después del cambio de contraseña por parte del usuario.
- 6. Días antes de que caduque la contraseña, advierte al usuario para que la cambie.
- 7. Días después de que caduque la contraseña, esa cuenta estará deshabilitada.
- 8. Días desde el 1/Enero/1970. Fecha absoluta que especifica cuando ya no se pueda usar el inicio de sesión para esa cuenta.

### Comando chage (caducidad en las contraseñas)

Con el comando **chage** (change age) podemos establecer la caducidad de contraseñas y cuentas de usuario. Esto no afecta a los nuevos usuarios, se establece de forma nominal a usuarios existentes.

Para aplicar estas opciones a nuevos usuarios y así generar estas directivas por defecto habría que hacer uso de **login.defs** que se comenta más adelante en este artículo.

Opciones del comando chage:

- **-d, --lastday**: Establece el día del último cambio de la contraseña.
- **-E, --expiredate**: Establece la fecha de caducidad.
- **-I, --inactive**: Deshabilita la cuenta después inactividad de días de la fecha de caducidad.
- **-l, --list**: Muestra la información de la edad de la cuenta.
- **-m, --mindays**: Establece el número mínimo de días antes de cambiar la contraseña.
- **-M, --maxdays**: Establece el número máximo de días antes de cambiar la contraseña.

BUSCAR EN EL BLOG

OTROS PROYECTOS

- [Mis libros en Amazon Kindle](#)
- [Docker Swarm AWS ASG ELB](#)
- [github.com/adrianlois](https://github.com/adrianlois)
- [youtube.com/adrianlois](https://youtube.com/adrianlois)
- [500px.com/adrianlois](https://500px.com/adrianlois)
- [vimeo.com/maliciousmaik](https://vimeo.com/maliciousmaik)

SUSCRIPCIONES Y CONTACTO

Buzón Público de Adrián Lois



Suscripción por email

Dirección Email...

### Tweets por @adrianlois\_

**Adrian Lois**  
@adrianlois\_

Los puntos de referencia del CIS son reconocidos internacionalmente como estándares de seguridad para la defensa de sistemas.

Wynis realiza los benchmarks del CIS Best Practices. [zonasystem.com/2021/02/wynis-...](https://zonasystem.com/2021/02/wynis-...)  
#CIS #ActiveDirectory #hardening #windows #cybersecurity #ciberseguridad

14h

[Insertar](#) [Ver en Twitter](#)

ARCHIVO DEL BLOG

- [2021](#) (2)
- ▼ [2020](#) (23)
  - [noviembre](#) (1)

- **-R, --root:** Directorio en el que hacer chroot.
- **-W, --warndays:** Establece los días de aviso de expiración.

Con el parámetro **-l** podemos ver información sobre las cuentas.

```
# chage -l pepe
Last password change           : Apr 18, 2020
Password expires                : never
Password inactive               : never
Account expires                 : never
Minimum number of days between password change : 0
Maximum number of days between password change : 90
Number of days of warning before password expires : 5
```

Más información sobre el comando chage: <https://linux.die.net/man/1/chage>

## Cumplimiento de directivas de seguridad en la calidad de contraseñas seguras de usuarios

Al igual que en sistemas Windows es necesario implementar un mínimo cumplimiento de seguridad en el hardening de contraseñas estableciendo directivas a nivel de dominio para definir la complejidad de contraseña, longitud mínima, días de expiración, contraseñas no repetidas respecto a las anteriores, etc.

En sistemas Linux, aplicaciones de terceros o servicios exógenos que hacen uso de usuarios Linux, también es necesario establecer políticas de seguridad estrictas al tratarse de cuentas que pueden tener el mismo riesgo de ser vulneradas y poder pivotar a otros sistemas que sean de un mayor interés para los atacantes.

Lógicamente también dependerá de la infraestructura y su ecosistema. En entornos de producción no sería conveniente provocar la desactivación o bloqueo de una cuenta de usuario y dejar inoperativo un servicio crítico que dependa de dicha cuenta. Independientemente de estos casos es recomendable y diría que necesario establecer este tipo de criterios para garantizar y elevar un poco más las medidas de seguridad de los sistemas de la empresa.

Se pueden establecer políticas por defecto en el momento de creación de nuevas cuentas en el sistema, sin la necesidad de usar chage nominalmente a cada usuario como vimos anteriormente.

## login.defs

Se trata de un fichero de texto situado en **/etc/login.defs** de forma nativa se encuentra en cualquier entorno Linux. Se definen las políticas de gestión de contraseñas para la creación de nuevos usuarios. También podemos establecer otras directivas como el valor umask por defecto, si el usuario tendrá un home para el si no se le especifica en su creación (useradd -m), cambiar la secuencia de ID (por defecto suelen ser 1000 y algo para usuarios normales), mensaje de bienvenida del inicio de sesión del usuario (motd\_file), etc.

El motivo de este artículo será enfocado únicamente a las que tengan que ver con las políticas de contraseñas.

Algunas de las directivas más comunes que podemos establecer:

- **PASS\_MAX\_DAYS:** Número máximo de días que se puede usar una contraseña.
- **PASS\_MIN\_DAYS:** Número mínimo de días permitido entre cambios de contraseña
- **PASS\_WARN\_AGE:** Número de días de advertencia antes de que caduque una contraseña.
- **PASS\_MIN\_LEN** y **PASS\_MAX\_LEN:** Número mínimo y máximo de caracteres que debe tener la contraseña.
- **PASS\_ALWAYS\_WARN:** Advierte sobre contraseñas débiles.
- **PASS\_CHANGE\_TRIES:** Número máximo de intentos de cambiar la contraseña si se rechaza por que es demasiado "fácil".
- **ENCRYPT\_METHOD:** Tipo de cifrado que tendrá la contraseña (SHA256 \$5\$ o SHA512 \$6\$).

► [septiembre](#) (2)

► [julio](#) (2)

► [junio](#) (6)

► [mayo](#) (3)

▼ [abril](#) (3)

[Bypass UAC Fileless usando AppPaths sdclt.exe](#)

[Gestión de políticas de contraseñas en Linux: logi...](#)

[sshuttle: Múltiples túneles SSH y "VPN" \(Proxy tra...](#)

► [marzo](#) (2)

► [febrero](#) (2)

► [enero](#) (2)

► [2019](#) (21)

► [2018](#) (15)

► [2017](#) (11)

► [2016](#) (10)

► [2015](#) (17)

► [2014](#) (13)

► [2013](#) (9)

► [2012](#) (16)

► [2011](#) (8)

► [2010](#) (14)

ETIQUETAS

[Windows](#) [Seguridad](#) [Redes](#) [Seguridad](#) [Informática](#) [Hacking](#) [Linux](#) [Passwords](#) [Pentesting](#) [Windows](#) [Server](#) [Análisis](#) [Forense](#) [Hacking](#) [Ético](#) [Administración](#) [Remota](#) [Hardening](#) [PsExec](#) [PsTools](#) [RDP](#) [Regedit](#) [SMB](#) [ARP](#) [CIFS](#) [Firewalls](#) [Permisos](#) [CLI](#) [Cracking](#) [GPO](#) [Virtualización](#) [Auditoría](#) [Backups](#) [Github](#) [Metasploit](#) [SSH](#) [Active](#) [Directory](#) [Certificados](#) [digitales](#) [Port](#) [Forwarding](#) [PowerShell](#) [ACLs](#) [Netsh](#) [Network](#) [Scan](#) [Scripting](#) [Virtualbox](#) [ADDS](#) [Brute-force](#) [CA](#) [Cron](#) [DNS](#) [Data](#) [Recovery](#) [FTP](#) [Hardware](#) [Hashes](#) [Impresoras](#) [MITM](#) [Malware](#) [Metadatos](#) [Pivoting](#) [Recuperar](#) [Datos](#) [Tunneling](#) [WMI](#) [Wireshark](#) [BIOS](#) [Batch](#) [processing](#) [Bootimg](#) [Bypass](#) [UAC](#) [Elevación](#) [de](#) [privilegios](#) [Escalada](#) [de](#) [privilegios](#) [FileSystem](#) [PKI](#) [Políticas](#) [de](#) [seguridad](#) [Post-Explotación](#) [RDS](#) [WiFi](#) [Wireless](#) [X.509](#) [DDNS](#) [Esteganografía](#) [Eventvwr](#) [Explotación](#) [local](#) [FSMO](#) [Fingerprinting](#) [HTTP](#) [HTTPS](#) [Hijacking](#) [IDPS](#) [Internet](#) [Explorer](#) [John](#) [The](#) [Ripper](#) [NTLM](#) [OpenSSL](#) [Perfiles](#) [Phishing](#) [Proxy](#) [Terminal](#) [Services](#) [UAC](#) [Virus](#) [Web](#) [Browser](#) [Windows](#) [Update](#) [AWS](#) [Android](#) [Antimalware](#) [Antivirus](#) [Apache](#) [CAL](#) [Data](#) [Wiping](#) [Docker](#) [Ettercap](#) [Exploit](#) [FRS-DFSR](#) [Fileless](#) [Fingerprint](#) [Google](#) [Hashcat](#) [Integridad](#) [Kali](#) [Login](#) [Logs](#) [Meterpreter](#) [Mimikatz](#) [Mozilla](#) [Firefox](#) [NTFS](#) [Nmap](#) [Pass](#) [the](#) [hash](#) [Portproxy](#) [Privacidad](#) [Pth](#) [Python](#) [Replicación](#) [Samba](#) [Servicios](#)

- **LOGIN\_RETRIES:** Número máximo de reintentos de inicio de sesión en el caso de que la contraseña sea incorrecta.
- **LOGIN\_TIMEOUT:** Tiempo máximo en segundos para iniciar sesión

Más información sobre login.defs: <https://linux.die.net/man/5/login.defs>

## pam\_cracklib y pam\_pwquality

Si queremos tener una mayor nivel de detalle en la personalización de las directivas en la complejidad de contraseñas para las cuentas de usuarios sería más interesante hacer uso de los módulos **PAM** (*Pluggable Authentication Modules*) **pam\_cracklib** y **pam\_pwquality**.

**pwquality** es una versión más actual y mejorada de cracklib. pwquality llama a una rutina cracklib para verificar si la contraseña es parte de un diccionario si este se especifica en la directiva *dictpath*. En este caso haré referencia siempre a pam\_pwquality.

En entornos RHEL/Centos se incorpora de forma nativa, es compatible con derivados Debian pero será necesario instalarlo.

```
sudo apt install -y libpam-cracklib libpam-pwquality libpwquality-tools
```

Una vez instalado el módulo de libpwquality podemos editar sus opciones en el fichero ***/etc/security/pwquality.conf***.

Opciones para definir las políticas:

- **difok:** Número de caracteres en una nueva contraseña que no deben estar presentes en la contraseña anterior.
- **minlen:** Tamaño mínimo aceptable para la nueva contraseña.
- **dcredit:** Crédito máximo por tener dígitos en la nueva contraseña.
- **ucredit:** Crédito máximo por tener letras mayúsculas en la nueva contraseña.
- **lcredit:** Crédito máximo por tener letras minúsculas en la nueva contraseña.
- **ocredit:** Crédito máximo por tener otros caracteres en la nueva contraseña.
- **minclass:** Número mínimo de clases de caracteres requeridas para la nueva contraseña
- **maxrepeat:** Número máximo de caracteres repetidos.
- **maxclassrepeat:** Número máximo de caracteres consecutivos en la misma clase.
- **gecoscheck:** Verifica si las palabras individuales de más de 3 caracteres del campo passwd GECOS (campo de comentarios) del usuario están contenidas en la nueva contraseña.
- **dictpath:** Ruta a los diccionarios de cracklib.
- **badwords:** Lista de palabras separadas por espacios que no deben incluirse en la contraseña.

## Sistema de créditos, valores negativos y clases en pwquality

### Créditos

Para definir la calidad y complejidad de las contraseñas se utiliza un sistema de créditos. Esto es muy interesante, básicamente se obtienen créditos por la complejidad. Una contraseña más corta podría ser aceptable si es más compleja en otras formas.

Por ejemplo una contraseña "ahwouwtbye" podría pasar una prueba minlen = 10. Si dcredit se establece en 2, una contraseña "ahwouw12" pasaría la prueba por que obtendríamos 2 créditos por cada dígito, entonces 8 caracteres más 2 créditos se valoran como 10 caracteres. Esto dependerá de como se establezcan los parámetros ucredit, lcredit, dcredit y ocredit.

### Valores negativos

Establecer valores de créditos negativos significa que debe tener al menos ese tipo de carácter. Por ejemplo, establecer dcredit a -1 significaría que debe incluir al menos un dígito para que se acepte una contraseña. Es decir, no se trata de una suma de créditos sino de un requerimiento obligatorio.

### Clases (minclass)

Taskschd Troyano VMware msfvenom vCenter  
vSphere AP Anonimato Apps Auto Scaling BitLocker  
Buscadores CIS Checksum Clonación Cloud  
Computing DHCP DLL Hijacking DLL Injection DLLs  
Data Exfiltration Drivers ENS ESXi Eliminación de datos  
Enumeración Footprinting Google Chrome Google  
Dorks Google Hacking Hacking Buscadores ICACLS  
IPv6 ISO Keylogger Movimiento lateral Movimiento  
vertical NIDS Nginx OSINT POSIX RaspberryPI S3 SAM  
SCCM SSL/TLS Scanning Seguridad Web Social  
Network Spyware System Center UEFI USB VCSA VPN  
Vulnerabilidades WCE WHOIS Winscp rsync

Otra configuración interesante es minclass. Determina cuántas clases diferentes de caracteres se deben usar para que una contraseña sea aceptable. Hay 4 tipos de clases: minúsculas, mayúsculas, dígitos, caracteres especiales (símbolos o signos).

Por ejemplo, un minclass = 2 exige que una contraseña contenga la combinación de dos tipos de clases. Ya sean mayúsculas o minúsculas, mayúsculas y caracteres especiales, minúsculas y dígitos, etc. Lo mismo pasaría si se establece a minclass = 4, la contraseña debería contener

También se puede establecer un límite al número máximo de caracteres de cualquier tipo de clase. Por ejemplo, con el parámetro *maxclassrepeat* = 4 indicamos que las contraseñas no pueden contener más de 4 minúsculas, mayúsculas, dígitos y otros caracteres especiales.

Más información sobre el uso de directivas del fichero de pwquality: [https://linux.die.net/man/8/pam\\_pwquality](https://linux.die.net/man/8/pam_pwquality)

Probando el cumplimiento de requisitos con varias contraseñas.

```
# passwd pepe
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña tiene menos de 8 caracteres.

# passwd pepe
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña no supera la verificación de diccionario - No
contiene suficientes caracteres DIFERENTES.

# passwd pepe
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña no supera la verificación de diccionario - Es
demasiado simple/sistemática.

# passwd pepe
Nueva contraseña:
CONTRASEÑA INCORRECTA: La contraseña contiene más de 3 caracteres de la misma clase
en forma consecutiva
```

## pwscore

pwquality (libpwquality-tools) dispone de una herramienta llamada **pwscore** que podemos usar para comprobar la complejidad de una contraseña en base a los criterios establecidos en pwquality. Algunos ejemplos de uso.

```
# echo 123 | pwscore
Falló la comprobación de calidad de la contraseña:
La contraseña tiene menos de 8 caracteres

# echo abc123.. | pwscore
Falló la comprobación de calidad de la contraseña:

La contraseña no supera la verificación de diccionario - Es demasiado
simple/sistemática.

# echo L3h5as/2a%-ls=72 | pwscore
100
```

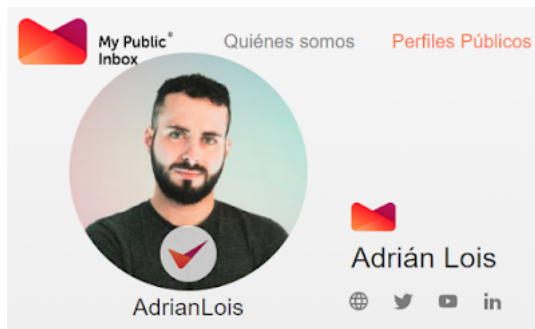
## Conclusión

Estos módulos PAM también pueden comprobar si las contraseña son un palíndromo o si solamente se cambió un solo carácter de mayúscula a minúscula o viceversa, si las contraseñas anteriores son similares o ya han sido usadas anteriormente, si contiene el nombre del usuario, o si hay alguna coincidencia con el campo passwd GECOS (campo de comentarios) de /etc/passwd. Dependiendo el nivel de complejidad que se establezca sería difícil establecerse una mala contraseña.

En definitiva, ofrecen una serie de comprobaciones que ayudan a garantizar un nivel de robustez y calidad de las contraseñas. Así como en la mayoría de compañías se aplican para entornos Windows, intentar aplicar estas políticas de seguridad en sistemas Linux y que no habiten en el olvido.

Saludos!

Autor: [Adrián Lois](#)




Publicado por [Adrián Lois](#)

Etiquetas: [Auditoría](#), [ENS](#), [Hardening](#), [Linux](#), [Passwords](#), [Políticas de seguridad](#), [Seguridad](#)

No hay comentarios:

Publicar un comentario

 Comentar como: vicentalderon ▼

☐ **Avisarme**

[<< Entradas antiguas](#)

[Inicio](#)

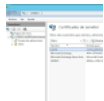
[Entradas recientes >>](#)

Suscribirse a: [Enviar comentarios \(Atom\)](#)

#### ENTRADAS RELACIONADAS



[Exportar certificados digitales no exportables a PFX \(PKCS #12\) y restablecer la password de la clave privada con Jailbreak y Mimikatz](#)



[OpenSSL - Convertir certificados digitales: .PFX .CSR .PEM .CRT o .CER y extraer la clave privada de certificados](#)



[Habilitar virtualización anidada VT-x/AMD-V en Virtualbox cuando no se puede activar \(nested virtualization\)](#)



[Ejecutar una CMD en un equipo remoto con psexec](#)