Pràctica2: Usuaris-Password-Classes

Curs 2019 - 2020

ASIX M01-ISO Implantació de sistemes operatius
UF2 Administració d'usuaris

Pràctica a fer

Usarem el recurs creat en la pràctica anterior al vostre **gitlab** anomenat **asix-m01** d'accés públic. Deseu els exercicis a fer d'aquesta pràctica en aquest recurs. Envieu un email al professor on consti clarament el vostre nom i cognoms i un enllaç al recurs de gitlab on hi ha les vostres solucions *informant que ja heu acabat aquesta segona pràctica*. Teniu fins a finals de setmana santa.

Repositori del professor

Recordeu que a classe hem estat treballant amb el repositori de gitlab on hi ha penjat el material que hem estat fent a classe gitlab de edtasixm01 recurs Scripts-2019-2020.

Recordeu també que el material de l'assignatura està a la web de Google Sites ASIX-M01 i que ara estem treballant la part d'administració d'usuaris.

Una mica de descripció i repàs del què farem

En aquest bloc hem estat treballant l'administració d'usuaris i ja hem vist les ordres generals que s'utilitzen, però encara ens falta la part de gestió de la política d'expiració de passwords.

També ens falta practicar scripts automatitzats amb la eliminació (o no) de les dades i recursos dels usuaris i també la creació de conjunts d'usuaris, com per exemple donar d'alta tota la classe hisix1.

Finalment ens falta la part de personalització de alies, variables i funcions globals al sistema o per a cada usuari.

Pràctiques a realitzar:

1. Polítiques de password.
2. Eliminar un usuari i els 'seus elements'.
3. Crear un conjunt d'usuari de cop: una classe
4. Fitxers bash_profile i bash_rc

Política de passwords

Exercicis Camps /etc/shadow

Feu els exercicis que treballen amb el fitxer /etc/shadow i que us serviran per identificar cada un dels seus camps (consulteu man 5 shadow). Són els exercicis clàssics de retallar i mostrar camps.

Al github de edtasixm11 dins de Scripts-2019 al directori 04-usuaris hi ha el fitxer Exercicis_scripts_usuaris.pdf, **feu els exercicis 26, 27 i 28**.

Ordres i fitxers de la política de gestió de password:

Haureu identificat que els camps del /etc/shadow són:

- Login
- Password
- Date of last password change
- Minimum password age
- Maximun passord age
- Password warning period
- Password inactivity period
- Account expiration date

El què cal és que entengueu què significa i per a què serveix cada camp. La manera d'aprendre-ho serà practicant.

Les ordres amb les que treballarem aquest apartat i de les que heu de practicar-ne les opcions són:

- ❏ passwd
- ❏ chage
- ❏ cat /etc/shadow
- ❏ cat /etc/login.defs (observeu-ne els valors relacionats)

Les principals accions a realitzar són: (heu de fer-les totes!)

- Identificar el contingut del camp password quan encara no se n'ha assignat cap.
- Identificar-ne el contingut quan té un password assignat
- Identificar-ne el contingut quan el password està bloquejat.
- Identificar-ne el contingut quan és un usuari passwordless

- Bloquejar un compte d'usuari
- Desbloquejar un compte d'usuari
- Generar un compte d'usuari passwordless

- Amb l'ordre passwd veure les característiques del password d'un compte d'usuari.

- Amb l'ordre chage veure les característiques del password d'un compte d'usuari.

- Establir a un password una determinada política (max, min, warning, etc).
- Modificar a un password valors de la seva política d'expiració.

En establir requisits a la política de password hem de practicar iniciar sessió amb l'usuari user1 i observar cada una de les següents situacions:

- No permet canviar el password perquè fa poc que ja el vam canviar.
- Toca canviar el password perquè ja ha passat el període màxim de validesa.
- Mostra un waring informant que cal canviar el password.
- Obliga a canviar el password en iniciar sessió
- Ja no permet iniciar sessió
- El compte (no el password) h expirat.

Gestió de la política d'expiració d'un password

Primerament cal identificar cada un dels camps del /etc/shadow, saber què signifiquen i per a què serveixen i després saber modificar aquests valors amb les ordres passwd i chage (preferentment).

Observem les dades de la política del compte de l'usuari user1:

```
# passwd -S user1
user1 PS 2020-03-06 0 99999 7 -1 (Password set, SHA512 crypt.)

# chage -l user1
Last password change                                    : Mar 06, 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          : 99999
Number of days of warning before password expires       : 7
```

En aquest document NO us explicaré què es cada camp, aquesta és la feina que heu de fer vosaltres, però sí que us explicaré un truc per a poder practicar.

Observeu que si ara mateix li canvieu el password a user1 el camp 3 que indica quan se li ha establert el password indica un número corresponent a la data d'avui, però expressat en el número de dies des del 1 de gener de 1970 (la data unix de creació de l'univers segons la religió linux).

```
# passwd user1
Changing password for user user1.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
```

```
passwd: all authentication tokens updated successfully.

# grep "^user1:" /etc/shadow
user1:$6$AEbqeWPtURV95ypg$mje7XI5.wxdYRIl8kz5MpafPqfKIui7dCzECTIoDbtFVdO8
Ap3/4RmZDludcCsa1T0eKNW1Lcy9XGxKGpHA6m0:18351:0:99999:7:::

# passwd -S user1
user1 PS 2020-03-30 0 99999 7 -1 (Password set, SHA512 crypt.)
```

En l'exemple han passat 18351 dies des del 1 de gener de 1970, que correspon a la data en que he fet l'ordre, el 30.03-2020 (un exercici xuli per fer amb la anna a programació oi?).

Per practicar modificar el max, min, warning, invalidity, expires, etc podeu fer les següents opcions:

A. Posar un password que per exemple expira d'aqui tres dies. Preparar una cerveseta (preferiblement voll dam) unes olivetes i pacientment esperar tres dies i verificar si un cop transcorreguts el password ha expirat.

B. Posar un password que per exemple expiri d'aquí tres dies i tot seguit canviar la data del sistema i posar-la avançada tres dies per verificar si el password s'ha bloquejat o no. L'inconvenient de provar-ho així és que usualment acabem deixant malament la data del sistema. Per exemple a l'aula, a classe, no podem fer això perquè llavors es queda penjat el kerberos i el nfs en no coincicir les dates del vostre ordinador i les del servidor gandhi.

C. Fer trampa (això mola) i modificar manualment el 3r camp on consta la data en que hem establert el password. Aquest és el mecanisme més perillós però és el que usarem!.

El que farem és editar manualment amb vi el fitxer /etc/shadow i a l'usuari user1 anar-li modificant el valor del tercer camp que indica quant li vam canviar el password. Hem de ser molt curosos perquè si espatlleu el /etc/shadow el sistema NO ARRANCARÀ!.

Feu una còpia del /etc/shadow abans de fer els exercicis!

Anoteu en un paper el valor que hi ha al 3r camp de user1 (18351 en aquest exemple corresponent al 30/03/2020).

Observeu per exemple si li posem el valor 0 al 3r camp: 1 de gener de 1970

```
# passwd -S user1
user1 PS 1970-01-01 0 99999 7 -1 (Password set, SHA512 crypt.)

# grep "^user1:" /etc/shadow
user1:$6$AEbqeWPtURV95ypg$mje7XI5.wxdYRIl8kz5MpafPqfKIui7dCzECTIoDbtFVdO8
Ap3/4RmZDludcCsa1T0eKNW1Lcy9XGxKGpHA6m0:0:0:99999:7:::
```

Si li posem un 1 veureu que ara és al 2 de gener del 1970

```
# passwd -S user1
user1 PS 1970-01-02 0 99999 7 -1 (Password set, SHA512 crypt.)

user1:$6$AEbqeWPtURV95ypg$mje7XI5.wxdYRII8kz5MpafPqfKIui7dCzECTIoDbtFVdO8
Ap3/4RmZDludcCsa1T0eKNW1Lcy9XGxKGpHA6m0:1:0:99999:7:::
```

Tornem a posar la data d'avui (18351)

```
# passwd -S user1
user1 PS 2020-03-30 0 99999 7 -1 (Password set, SHA512 crypt.)

user1:$6$AEbqeWPtURV95ypg$mje7XI5.wxdYRII8kz5MpafPqfKIui7dCzECTIoDbtFVdO8
Ap3/4RmZDludcCsa1T0eKNW1Lcy9XGxKGpHA6m0:18351:0:99999:7:::
```

El truc consisteix en anar tirant endarrera la data del 3r camp, si la tirem deu dies endarrera llavors avui és el desè i podem mirar si el password ja ha caducat o no. Si anem fent això anirem passant per tots els períodes max, min, warning, etc i podem anar veient quin efecte fan. A cada prova intentem iniciar sessió amb l'usuari user1 en una altra consola i observem si ens deixa o no, si mostra el warning, si obliga a canviar el password o si ja no deixa entrar.

Us recomano que dibuixeu una recta amb els dies anotats on el dia de referència és avui (el dia on practiqueu), el dia on estem validant què passa i aneu tirant endarrera dia a dia (3r camp).

Per exemple

16	17	18	19	20	21	22	23	24	25	26	27	28	30	31
X	min	min			war	war	ina	ina					avui	

Hem simulat que hem posat la data el dia 16 de març, al password de l'usuari user1 se li ha establet dos dies de min, 6 de max, 2 de warning i dos de inactivity. Si ara, avui dia 30 de març, intenta entrar ja no podrà.

24	25	26	27	28	29	30	31	1	2	23	4	5	6	7
X	min	min			war	avui war	ina	ina						

Si simulem que hem canviat el password el dia 24 llavors avui dia 30 estem just en el punt de warning i últim dia en que el passwd és vàlid.

Es tracta d'anar fent aquestes proves fins entendre que fan tots els camps del /etc/shadow.

Eliminar un usuari

Fins ara sabem que podem eliminar un compte d'usuari amb userdel, també sabem eliminar el compte de l'usuari i el seu home amb userdel -r usuari.

Però segur que això és tot el que té l'usuari?

L'usuari pot tenir:

- Processos
- Treballs d'impressió
- Tasques periòdiques (cron, at, batch)
- Correu
- Fitxers en el sistema de fitxers (fora del seu home)
- El seu home.

Què en fem de tot això? El típic exemple és aquell procés en background en el sistema que va verificant que l'usuari en forma part, el dia que no en forma part (l'han acomiadat) el procés decideix esborrar tot el disc dur…

L'altre aspecte a considerar és: segur que volem esborrar el seu home i els fitxers de l'usuari que no estan al seu home sinó repartits per altres llocs? Si són informes, treballs, plànols, càlculs científics de l'organització/empresa no els volem llençar, els volem mantenir. Molt sovint el que interessa és recopilar-ho tot i fer-ne un tar.gz.

Tasca 1:

Identificar com obtenir la informació de:

- Processos que un usuari té al sistema i com matar-los.
- Treballs d'impressió que un usuari té al sistema i com matar-los.
- Tasques periòdiques que un usuari té al sistema i com matar-les.
- Fitxers (fora del home) que un usuari té al sistema i com desar-los en un tar.gz.
- Moure tot el home d'un usuari a un tar.gz.

Tasca 2:

Feu un programa anomenat **del_usuari.sh** que rep un login i elimina TOT el que pertany a l'usuari, però deixant un tar.gz de tots els fitxers que li pertanyen. Va generant una traça per stderr de tot allò que va eliminant.

Crear un conjunt d'usuaris

Fins ara hem vist com crear un usuari amb l'ordre useradd establint-hi les opcions particulars que facin falta, però com podem automatitzar la creació per exemple d'una classe com hisx1?

En crear classes d'alumnes (per exempel hisix1) cal tenir en compte:

- Es crea un grup del sistema amb el nom del grup, si ja existeix no és un error. Aquest és el grup principal dels alumnes, però com a grup secundari també pertanyen a users.

- Es crea un directori home base propi per la classe tipus /home/inf/hisx1/<alumne> dins del qual hi van els directoris dels alumnes.

Tasca 3:

Crea un programa anomenat **crea-classe-hardcoded.sh** que crea 30 alumnes (proveu amb 3 per començar…) del curs que rep com a argument.

Podeu generar els noms dels alumnes automatitzadament amb brace expansion utilitzant hisx1-{01-30}.

Als alumnes els assignem el passwd 'alum' per defecte.

Tasca 4:

Crea un programa anomenat **crea-classe-file.sh** que crea alumnes del curs que rep com a primer argument. Rep un segon argument corresponent a un fitxer amb noms:password d'alumnes a donar d'alta.

Es tracta de crear la classe amb el nom indicat per el primer argument i afegir-hi els alumnes que conté el fitxer rebut com a segon argument. Aquest fitxer conté per a cada línia el nom de l'usuari i el seu password amb el format *nom:passwd*.

Tasca 5:

Crea un programa anomenat **crea-classe-random.sh** que crea alumnes del curs que rep com primer argument. Rep un segon o més arguments corresponents a noms d'usuaris a donar d'alta en aquesta classe. Cada argument és un login (excepte el primer).

A cada usuari se li assigna un password random de 8 dígits/caracters.

Evidentment si es creen els usuaris i se'ls assigna un passwd random si no es desa aquesta informació enlloc els usuaris no podran entrar… El programa genera un fitxer **passwd.log** amb l'associació de usuari i passwd generat amb el format l*ogin:passwd*.

Bash_profile i Bashrc

Al llarg del curs hem estudiat variables d'entorn, àlies, funcions i ara estem treballant l'administració d'usuaris. Hem vist que en crear un usuari se li crea un directori home amb un contingut predeterminat per l'SKEL a usar. Aquest contingut usualment conté entre altres els fitxers:

- bash_profile
- bashrc

De fet hi ha quatre fitxers relacionats dos serveixen per a definir valors globals per a tots els usuaris del sistema, és el que anomenem **System Wide**. Dos serveixen per personalitzar valors per a cada **usuari individual**.

Perquè n'hi ha dos de cada (un profile i un rc) és un tema prehistòric que queda fora del temari. La qüestió és que algunes coses es desen en el profile i d'altres en el rc. Quines?

Anem a pams:

- L'administrador del sistema pot definir variables d'entorn, funcions, àlies i startup programs per a tots els usuaris del sistema. Ell els defineix i els hereten tots els usuaris.
- Un usuari individual es pot definir i redefinir (les de l'administrador) variables d'entorn, àlies, funcions i startup programs al seu gust.

Tasca 6:

- Identifiqueu quin són els fitxers (ruta absoluta) de definició system wide.
- Identifiqueu quins són els fitxers de personalització individual d'usuari.
- Identifiqueu a quins fitxers es defineixen les variables d'entorn i els startup programs.
- Identifiqueu a quins fitxers es defineixen les funcions i els àlies.

Tasca 7:

- Com a administrador defineix 3 variables d'entorn i 3 àlies.
- Com a usuari defineix una variable d'entorn i un àlies propi.
- Com a usuari redefineix una variable d'entorn i un àlies dels que havia definit l'administrador.