Pràctica Kerberos

Curs 2019-2020

Authenticació Kerberos

Pràctica1

Imatges Docker

edtasixm11/k19:kserver servidor kerberos detach. Crea els principals pere(kpere) pau(kpau, rol: admin), jordi(kjordi), anna (kanna), marta (kmarta), marta/admin (kmarta rol:admin), julia (kjulia) i admin (kadmin rol:admin). Crear també els principals kuser01...kuser06 amb passwd (kuser01...kuser06). Assignar-li el nom de host: *kserver.edt.org*.

edtasixm11/k19:khost host client de kerberos. Simplement amb eines kinit, klist i kdestroy (no pam). El servidor al que contacta s'ha de dir *kserver.edt.org*. Cal verificar el funcionament de kadmin.

Authenticació

edtasixm11/k19:khostp host amb PAM de kerberos. El servidor al que contacta s'ha de dir *kserver.edt.org*. Aquest host configura el *system-auth* de pam per usar el mòdul *pam_krb5.so*. Crear els usuris local01..local06 (idem nom de passwd) i kuser01..kuser06 (sense passwd). Aquest host utlitza /etc/passwd de IP Information Provider i valida usuaris locals local01... amb pam_unix.so (on /etc/passwd fa de IP i AP) i usuaris locals+principals kuser01... (on /etc/passwd fa de IP i kerberos de AP Authentication Provider).

Verificació:

En una sessió interactiva en el container khostp iniciar amb “su -” sessió com a local01, convertr-se en altre cop amb “su -” en local02 i finalment convertir-se en kuser01. Validar que kuser01 obté un ticket i que pot accedir amb kadmin a l’administració del servidor kerberos (amb independència dels permisos que tingui).

Pràctica2

Instal·lació

Eliminar del vostre host físic les particions sda2, sda3 i sda4. Crear una partició sda2 de 8GB. Instal·lar-hi Fedora-27 amb una instal·lació **MINIMAL**.

Refer el GRUB deixant per defecte la partició matí, les etiquetes MATI, TARDA i HISX2-LAB. Cal que el grub que mani (i el fitxer grub.conf) sigui el del matí.

Engegar la màquina a la partició matí (sda5)

Fer:

- # cp /boot/grub2/grub.cfg /boot/grub2/grub.hisx2
- # grub2-mkconfig > /boot/grub2/grub.cfg
- # vim /boot/grub2/grub.cfg   (veure què cal modificar)
- # grub2-install /dev/sda

Cal modificar:

- set timeout=-1
- set default=0 (o el número corresponent a l'entrada del matí, comencen per zero)
- MATI (posem aquesta etiqueta a la partició matí sda5)
- TARDA (posem aquesta etiqueta a la partició tarda sda6)
- HISX2-LAB (posem aquesta etiqueta a la partició de treball hisx2 sda2)

Host aula + Kerberos + AWS EC2

Usarem un host real de l'aula, la partició on hem instal·lat un Fedora 27 MINIMAL. Cal configurar la autenticació dels usuaris utilitzant Unix i Kerberos. El servidor kserver.edt.org estarà desplegat a AWS EC2.

Caldrà configurar una AMI a AWS EC2 amb docker i executar el kserver fent un mapping dels ports de kerberos al host de Amazon AWS EC2. També caldrà configurar el firewall. Per fer-ho crearem un *Security groups* propi anomenat *kerberos* que obri els ports del firewall per poder accedir des de l'exterior al kerberos i al ssh. Identifica els ports i de quin tipus són.

Penseu en tot el què cal configurar en el host de l'aula, podeu consultar la configuració que fem en iniciar el curs i planxar els ordinadors a fedora@inf.

En especial cal:

- Selinux
- Authconfig
- <trick>

Problema amb el caché KCM de kerberos:

Problema: pam_krb5[10992]: error updating ccache "KCM:"
Solució:

- Comentar la línia que defineix que utilitzi KCM de caché.
- ull que està en un altre fitxer en els sistemes reals, en els containers no hi és, per això van.
- /etc/krb5.conf.d/kcm_default_ccache

 # cat /etc/krb5.conf.d/kcm_default_ccache

```
# This file should normally be installed by your distribution into a
# directory that is included from the Kerberos configuration file (/etc/krb5.conf)
# On Fedora/RHEL/CentOS, this is /etc/krb5.conf.d/
#
# To enable the KCM credential cache enable the KCM socket and the service:
#   systemctl enable sssd-secrets.socket sssd-kcm.socket
#   systemctl start sssd-kcm.socket
#
# To disable the KCM credential cache, comment out the following lines.

[libdefaults]
            #default_ccache_name = KCM:
```

Pràctica3

Kerberos + LDAP (PAM)

Farem un nou container host client de kerberos i de ldap per verificar que sabem fer un muntatge equivalent al de l'escola. En aquest esquema usem dos containers servidors, un de kerberos i un de ldap (ja els tenim fets). Cal crear el container host client que es descriu a continuació.

edtasixm11/k19:khostpl (khost-pam-ldap) host amb PAM amb autenticació AP de kerberos i IP de ldap. El servidor kerberos al que contacta s'ha de dir *kserver.edt.org*. El servidor ldap s'anomena *ldap.edt.org*. Aquest host es configura amb *authconfig* (us ajudarà saber que és una configuració mimètica a la que fem en realitzar la instal·lació de les aules)..

Verificar en el host client l'autenticació d'usuaris locals i usuaris globals (ldap+kerberos). En el host client hi ha usuaris locals (local01…) usuaris locals amb passwd al kerberos (kuser01, etc que en realitat podem eliminar o ignorar) i usuaris de ldap (pere…, user1…). Aquests usuaris cal que tinguin password al kerberos (tipus kpere, kuser01, etc).

Host Aula + Kerberos + LDAP + AWS EC2

Configurar el host de l'aula amb Fedora-27-Minimal per tal de permetre l'autenticació d'usuaris locals amb pam_unix.so i usuaris globals kerberos+ldap. Cal utilitzar *authconfig*. Verificar l'accés d'usuaris locals local01,etc i d'usuaris globals pere, user01, etc.

nota: no confongueu els usuaris de ldap user01 amb els de 'mentida' que vam crear localment al client anomenats kuser01.

Caldrà configurar una AMI a AWS EC2 amb docker i executar el *kserver.edt.org* i el *ldap.edt.org* fent un mapping dels ports de kerberos i ldap al host de Amazon AWS EC2. També caldrà configurar el firewall. Per fer-ho crearem un *Security groups* propi anomenat *kerberos-ldap* que obri els ports del firewall per poder accedir des de l'exterior al kerberos i al ldap. Identifica els ports i de quin tipus són.

Authconfig

Practiqueu la utilització de les opcions --savebackup i --restorebackup de l'ordre authconfig. Recordeu que vam treballar aquesta ordre al fer PAM (HowTo-ASIX_PAM.pdf)a ASIX-M06. Permet desar i restaurar configuracions a /var/lib/authconfig/<nom>.
Creeu tres configuracions amb authconfig:

- ❏ Estàndard unix (la que venia per defecte).
- ❏ Unix amb Kerberos (corresponent a la pràctica 2).
- ❏ Unix, Kerberos i Ldap (corresponent a la pràctica 3).

Seveis Kerberitzats

Pràctica 4

Servei SSH Kerberitzat Bàsic

<salteu al següent exercici si heu fet completament la Pràctica 3 i ja disposeu d'un container amb autenticació Kerberos+ldap.>

edtasixm11/k19:sshd Servidor SSHD *kerberitzat*. Servidor ssh que permet l'accés d'usuaris locals i usuaris locals amb autenticació kerberos. El servidor s'ha de dir *sshd.edt.org*.

Primera versió simple (podem usar de base k19:khost) d'un host amb usuaris locals (local01…) i usuaris locals amb passwd al kerberos (kuser01…). A aquest host li afegim el servei ssh per convertir-se en un servidor SSH Kerberitzat. Ha de permetre l'accés tant a usuaris locals (local01) com a usuaris kerberos (kuser01).

El model de funcionament és disposar de un host client de kerberos, per exemple k19:khost i aquest servidor sshd kerberitzat. En el client un usuari 'qualsevol' es pot connectar i iniciar sessió al servidor SSH com a usuari destí local (local01).

En el client un usuari que disposi de ticket kerberos (per exemple kuser01) pot iniciar sessió remota al servidor ssh com a usuari kuser01 automàticament, ja que disposa de les credencials kerberos (similar a iniciar sessió desatesa amb claus pública/privada).

Servei SSH Kerberitzat (Kerberos + LDAP)

Si ja heu fet la Pràctica 3 i heu construït un host amb autenticació kerberos i Ldap que únicament disposa dels usuaris locals local01… i la resta els autentica via Kerberos (AP) i Ldap (IP), podeu usar de base aquesta imatge que s'anomenava k19:khostpl.

L'objectiu és crear un servidor sshd que simplement disposa dels usuaris locals (local01…) i dels usuaris de xarxa (kerberos+ldap). Aquest servidor permet que es connectin remotament tant usuaris locals com usuaris de xarxa.

Als usuaris que disposen d'un ticket de kerberos han de poder fer login automàticament (sense que se'ls demani el password). Per fer-ho caldrà configurar SSH per actuar com un servidor kerberitzat. Podeu consultar els apunts (How-to-ASIX_kerberos.pdf) con configurar un servidor kerberitzat. Bona sort amb l'aprenentatge del Keytab!.

edtasixm11/k19:sshdpl (sshd-pam-kerberos-ldap) Servidor SSH amb PAM amb autenticació AP de kerberos i IP de ldap. El servidor kerberos al que contacta s'ha de dir *kserver.edt.org*. El servidor ldap s'anomena *ldap.edt.org*. Aquest host es configura amb

authconfig . S'ha generat partint del host edtasixm11/k19:khostpl i se li ha afegit la part del servidor sshd. Conté els fitxers per poder activar el mount del home samba, però no s'ha configurat.

Desplegament SSH a AWS EC2

Desplegueu tots els servidors en una màquina AWS EC2. Cal engegar-hi *kserver.edt.org*, *ldap.edt.org* i *sshd.edt.org*. Poseu atenció a la redirecció de ports necessària per accedir al servei sshd, no podem usar el port 22 perquè és el que ens permet accedir a la AMI. Useu el port 1022 del host AMI per poder accedir al servei sshd (port 22) del container. Genereu un nou Security Groups anomenat kerberos-ldap-sshd.

Recordeu que en el host client també cal configurar el client SSH per indicar-li que utilitzi Kerberos/GSSAPI. Cal que quan usem l'ordre SSH client aquesta transmeti automàticament les credencials de kerberos (si n'hi han).

Recordeu també de configurar apropiadament el fitxers client /etc/hosts indicant els FQDN dels servidors, **començant** per el *sshd.edt.org*.

Verifiqueu

- ❏ Des d'un client container host que podeu fer login i podeu fer sessions remotes al sshd un cop disposeu de tiquets de kerberos.
- ❏ Ídem des del host real de l'aula.

Pràctica 5

Afegim Samba

edtasixm11/k19:sshdpls (sshd-pam-kerberos-ldap-home-samba) Servidor SSH amb PAM (kerberos+ldap) que munta els homes dels usuaris (dins del home) via samba.

Samba
edtasixm11/k19:khostpls (khost-pam-ldap-samba) Conté els fitxers per activar el mount del home samba, que munta els homes dels usuaris (dins del home) via samba. Caldrà crear un volum amb els homes dels usuaris. Primer el farem manualment hardcoded i després amb un script de creació.

Volumes / Entrypoint / Scripts

Pràctica 6

Volumes: krb5-data i ldap-data

Volumes krb5-data
Desar la base de dades en un volum anomenat *krb5-data* de manera que les dades de kerberos siguin perdurables. Practiqueu amb kadmin des del client i amb un compte d'administració crear, modificar, esborrar i llistar principals (manteniu els per defecte).

Practiqueu a assignar permisos diferents als usuaris, en especial el de poder llistar els principals.

Consulteu els apartats:

- ❏ Kerberos ACLS
- ❏ Docker exec amb ladmin-local
- ❏ Kerberos i volumes
- ❏ Httpd i volums

Volumes ldap-data
La base de dades ldap es desa en un volum anomenat *ldap-data*.
La configuració ldap es desa en un volum anomenat *ldap-config*.

Entrypoint: kserver i ldap

Consulteu l'apartat:

- ❏ Entrypoint versus CMD

Entrypoint kserver
Modificar l'script startup.sh del servidor Kerberos per actuar com a entrypoint amb els següents arguments possibles:

- *res*: engegar el servei kerberos usant la base de dades existent actualment (el volum).
- *Initdb*: inicialitza la base de dades.¡ i engega el servei.
- *initdbedt*: inicialitza la base de dades de kerberos amb els principals per defecte i engega el servei.
- *kadmin*: executa kadmin-local passant-li la resta de paràmetres que es rebin en l'execució del container.

Entrypoint ldap

Modificar la imatge ldapserver:latest (*ldapserver:entrypoint*) de manera que tingui un script startup.sh de entrypoint que permeti:

- *initdb*: inicialitzar la base de dades ldap sense dades i engegar el servei.
- *initdbedt*: initiclitzar la base de dades ldap amb les dades per defecte usuals i engegar el servei.
- *listdn*: llistar els dn de la base de dades ldap usant una comanda de baix nivell "slapcat | grep dn".
- *start*: engegar el servei ldap (la base de dades, amb dades o sense, ha d'existir prèviament). Aquesta és l'opció per defecte.
- *: qualsevol altre conjunt de paràmetres que es passin com a CMDi s'executarà usant eval.

Cal usar un volume anomenat *ldap-data* per a les dades i un volume anomenat *ldap-config* per a la configuració ldap. L'script startup.sh ha de mostrar un debug de tot el que va fent si la *variable d'entorn DEBUG* és superior a zero. Cal passar aquesta variable amb l'ordre docker run amb un valor de 1 per verificar el funcionament de l'script.

Entrypoint kserver useradd/userdel

Ampliar l'script d'administració startup.sh del keserver de manera que contingui les opcions:

- *useradd*: rep les dades necessàries per crear un principal i una entrada d'usuari ldap.
- *userdel*: rep les dades necessàries per eliminar un usuari (principal i entrada ldap).
- *list*: llista els principals.

Kerberos orchestration: docker-compose / swarm

Nota Consultar el document Practica_Docker_Swarm_kerberos

Pràctica 7

docker -compose

Desplegar la app d'autenticació (kerberos + ldap + sshd) en un host AMI de AWS EC2 usant un fitxer docker-compose.yml. Usar el Security Groups anomenat kerberos-ldap-sshd creat prèviament.

Consulta l'apartat:

- ❏ docker-compose
- ❏ Docker-compose repliques / scale / deploy

docker-swarm local

Crear un swarm de 2 nodes usant els dos hosts de l'aula que teniu assignats. Desplegar-hi la app d'autenticació (kerberos+ldap+sshd).

Modificar en calent el desplegament fet de l'stack de la app i i afegir-hi un ***visualizer*** (port 8080) per monitorar el desplegament dels nodes i containers.

Modificar en calent el desplegament fet de l'stack de la app i i afegir-hi un ***portainer*** (port 9000) per monitorar el desplegament dels nodes i containers.

Observeu el visualizer i els serveis desplegats en l'stack. Practiqueu modificar l'estat dels nodes (*active*|*paused*|*drain*) i establir *constraints* de col·locació dels serveis (usant *rols* i *labels*).

Consulta els apartats:

- ❏ Docker Swarm

docker-swarm AWS EC2

Desplegar en dues (o tres!) màquines AMI de AWS EC2 la app d'autenticació (kerberos+ldap+sshd) més el visualizer (això dependrà de si les AMI 'aguanten' la càrrega, en general posar-hi el portainer és mala idea...).

Verifiqueu el funcionament des del host client local del funcionament de la app i observeu el desplegament fet amb el visualizer.

Feu modificacions al desplegament modificant els serveis, l'estat dels nodes i establint *constraints* per *rol* i *label*.

docker-swarm local + AWS

Passos a fer:

Desplegament replicated

1. Genereu un swarm amb dues màquines AMI de AWS i dos hosts de l'aula, quatre en total. El manager ha de ser un dels nodes AWS, perquè?.
   Utilitzeu apropiadament l'opció de docker swarm --advertise-ip.
2. Genereu un security groups anomenat *swarm-hello-visualizer* que obri els ports necessaris per a la comunicació dels nodes que formen el swarm i dels serveis visualizer i hello.
3. Modifiqueu el servei hello de manera que la pàgina web que mostra index.html contingui el nom del host, per exemple amb un echo $(hostname) en l'script install.sh.
4. Desplegueu al swarm el servei visualizer amb una constraint que el fixi al node manager i cinc rèpliques del servei hello. Observeu com es distribueixen amb el visualizer.
5. Modifiqueu el desplegament variant el número de rèpliques tant manualment amb l'ordre docker service scale com fent noves versions del deploy.

Global

6. Modificar el servei hello passant-lo de replicated a global. Observa com es desplega a tots els nodes.

Nodes

7. Torna a fer el desplegament del servei hello amb replicated amb 6 rèpliques. Pausa un dels nodes. Fes drain a un dels nodes. Torna a fer actiu un dels nodes.
   Es redistribueixen les rèpliques? Com ho fem?
8. Elimina un dels nodes (un de local) del swarm. Observa el desplegament. Torna'l a afegir al swarm.

Constraints

9. Posa etiquetes *lloc* als nodes AWS amb el valor *aws* i als nodes locals amb el valor *local*.
10. Desplega el servei hello únicament a nodes amb etiquetes *local*.
11. Posa el label *tipus* amb el valor *alfa* al node manager i a un dels hosts locals. Posa l'etiqueta *beta* als altres dos nodes.
12. Desplega el servei hello als nodes beta. Observa el desplegament amb visualizer.
13. Autra el desplegament.

Teoria

Autenticaction Provider AP

Kerberos proporciona el servei de proveïdor d'autenticació. No emmagatzema informació dels comptes d'usuari com el uid, git, shell, etc. Simplement emmagatzema i gestiona els passwords dels usuaris, en entrades anomenades *principals* en la seva base de dades.

Coneixem els següents AP:

- */etc/passwd* que conté els password (AP) i també la informació dels comptes d'usuari (IP).
- *ldap* el servei de directori ldap conté informació dels comptes d'usuari (IP) i també els seus passwords (AP).
- *kerberos* que únicament actua de AP i no de IP.

Information Provider IP

Els serveis que emmagatzemen la informació dels comptes d'usuari s'anomenen Information providers. Aquests serveis proporcionen el uid, gid, shell, gecos, etc. Els clàssics són /etc/passwd i ldap.

Model de pràctiques

El model que mantindrem a tot el mòdul ASIX M11-SAD és el següent:

- **ldap** al servidor ldap tenim els usuaris habituals pere, marta, anna, julia, pau, jordi. El seu password és el seu propi nom.

- **/etc/passwd** en els containers hi ha els usuaris locals local01, local02 i local03 que tenen assignat com a password el seu mateix nom.

- **kerberos + IP** els usuaris kuser01, kuser02 i kuser03 són principals de kerberos amb passwords tipus kuser01, kuser02 i kuser03. La informació del seu compte d'usuari és local al */etc/passwd* on **no** tenen password assignat.

- **kerberos + ldap** Al servidor kerberos hi ha també principals per als usuaris usuals ldap pere, marta, anna, julia, jordi, pau i els user01 .. user10. Els seus passwords són del tipus kpere, kmarta, kanna, kjulia, kjordi, kpau i kuser01..10.

Es resum, podem verificar l'accés/autenticació d'usuaris locals usant el prototipus *local01*, podem fer test de la connectivitat kerberos amb comptes locals amb usuaris tipus *kuser01*. I

finalment podem verificar l'autenticació d'usuaris kerberos amb ldap (fent de IP) amb els clàssics pere (kpere).

Apèndix

Krb5_Cache

Problema: pam_krb5[10992]: error updating ccache "KCM:"
Solució:

- Comentar la línia que defineix que utilitzi KCM de caché.
- ull que està en un altre fitxer en els sistemes reals, en els containers no hi és, per això van.
- /etc/krb5.conf.d/kcm_default_ccache

```
# cat /etc/krb5.conf.d/kcm_default_ccache
# This file should normally be installed by your distribution into a
# directory that is included from the Kerberos configuration file (/etc/krb5.conf)
# On Fedora/RHEL/CentOS, this is /etc/krb5.conf.d/
#
# To enable the KCM credential cache enable the KCM socket and the service:
#   systemctl enable sssd-secrets.socket sssd-kcm.socket
#   systemctl start sssd-kcm.socket
#
# To disable the KCM credential cache, comment out the following lines.

[libdefaults]
        #default_ccache_name = KCM:
```

/etc/krb5.conf (afegim la secció)

```
[appdefaults]
 debug=true
 debug_sensitive=true
 ccache_dir=/tmp
 cred_session=false
```

journalcrl -f

```
feb 25 16:58:19 asus unix_chkpwd[10344]: password check failed for user (pere)
feb 25 16:58:19 asus su[10342]: pam_unix(su:auth): authentication failure; logname=root uid=1007 euid=0 tty=pts/1 ruser=local01 rhost=  user=pere
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: debug
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: debug_sensitive
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: don't always_allow_localname
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: no ignore_afs
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: no null_afs
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: no cred_session
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: no ignore_k5login
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: user_check
```

```
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: will try previously set password first
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: will ask for a password if that fails
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: will let libkrb5 ask questions
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: no use_shmem
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: no external
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: multiple_ccaches
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: validate
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: flag: warn
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: banner: Kerberos 5
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: ccache dir: /tmp
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: ccname template: KCM:
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: keytab: FILE:/etc/krb5.keytab
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: token strategy: 2b
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: called to authenticate 'pere', configured realm 'EDT.ORG'
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: authenticating 'pere@EDT.ORG'
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: trying previously-entered password for 'pere', allowing libkrb5 to prompt for more
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: authenticating 'pere@EDT.ORG' to 'krbtgt/EDT.ORG@EDT.ORG'
feb 25 16:58:19 asus su[10342]: pam_krb5[10342]: attempting with password="kpere"
feb 25 16:58:29 asus su[10342]: pam_krb5[10342]: krb5_get_init_creds_password(krbtgt/EDT.ORG@EDT.ORG) returned 0 (Success)
feb 25 16:58:29 asus su[10342]: pam_krb5[10342]: validating credentials
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: error reading keytab 'FILE:/etc/krb5.keytab'
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: TGT verified
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: got result 0 (Success)
feb 25 16:58:34 asus su[10347]: pam_krb5[10347]: no need to create "/tmp"
feb 25 16:58:34 asus su[10347]: pam_krb5[10347]: created ccache "FILE:/tmp/krb5cc_1010_i7hnfq"
feb 25 16:58:34 asus su[10347]: pam_krb5[10347]: created ccache 'FILE:/tmp/krb5cc_1010_i7hnfq' for 'pere'
feb 25 16:58:34 asus su[10347]: pam_krb5[10347]: krb5_kuserok() says "true" for ("pere@EDT.ORG","pere")
feb 25 16:58:34 asus su[10347]: pam_krb5[10347]: destroyed ccache "FILE:/tmp/krb5cc_1010_i7hnfq"
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: 'pere@EDT.ORG' passes .k5login check for 'pere'
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: authentication succeeds for 'pere' (pere@EDT.ORG)
feb 25 16:58:34 asus audit[10342]: USER_AUTH pid=10342 uid=1007 auid=0 ses=4 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
msg='op=PAM:authentication grantors=pam_krb5 acct="pere" exe="/usr/bin/su" hostname=asus addr=? terminal=pts/1 res=success'
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: pam_authenticate returning 0 (Success)
feb 25 16:58:34 asus audit[10342]: USER_ACCT pid=10342 uid=1007 auid=0 ses=4 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
msg='op=PAM:accounting grantors=pam_unix acct="pere" exe="/usr/bin/su" hostname=asus addr=? terminal=pts/1 res=success'
feb 25 16:58:34 asus su[10342]: (to pere) root on pts/1
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: debug
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: debug_sensitive
feb 25 16:58:34 asus audit[10342]: CRED_ACQ pid=10342 uid=1007 auid=0 ses=4 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
msg='op=PAM:setcred grantors=pam_krb5 acct="pere" exe="/usr/bin/su" hostname=asus addr=? terminal=pts/1 res=success'
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: don't always_allow_localname
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no ignore_afs
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no null_afs
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no cred_session
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no ignore_k5login
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: user_check
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: will try previously set password first
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: will ask for a password if that fails
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: will let libkrb5 ask questions
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no use_shmem
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no external
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: multiple_ccaches
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: validate
feb 25 16:58:34 asus audit[10342]: USER_START pid=10342 uid=1007 auid=0 ses=4 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
msg='op=PAM:session_open grantors=pam_unix acct="pere" exe="/usr/bin/su" hostname=asus addr=? terminal=pts/1 res=success'
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: warn
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: banner: Kerberos 5
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: ccache dir: /tmp
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: ccname template: KCM:
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: keytab: FILE:/etc/krb5.keytab
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: token strategy: 2b
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: debug
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: debug_sensitive
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: don't always_allow_localname
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no ignore_afs
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no null_afs
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no cred_session
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no ignore_k5login
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: user_check
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: will try previously set password first
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: will ask for a password if that fails
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: will let libkrb5 ask questions
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no use_shmem
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: no external
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: multiple_ccaches
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: validate
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: flag: warn
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: banner: Kerberos 5
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: ccache dir: /tmp
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: ccname template: KCM:
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: keytab: FILE:/etc/krb5.keytab
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: token strategy: 2b
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: creating ccache for 'pere', uid=1010, gid=1010
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: error creating ccache using pattern "KCM:"
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: error creating ccache for user "pere"
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: failed to create ccache for 'pere'
feb 25 16:58:34 asus su[10342]: pam_krb5[10342]: pam_sm_open_session returning 14 (Cannot make/remove an entry for the specified session)
feb 25 16:58:34 asus su[10342]: pam_unix(su:session): session opened for user pere by root(uid=1007)
```