

Corso RedHat per sistemisti INFN

RHEL/SL/CentOS 7

Autenticazione & Autorizzazione

- Autenticazione
 - Processo attraverso il quale posso confermare o meno l'identità della «parte»
- Autorizzazione
 - Processo attraverso il quale si definisce cosa può o non può fare o a cosa può o non può accedere la «parte» che ho riconosciuto attraverso il processo di Autenticazione

Autenticazione

- Prevede che vengano presentate un qualche tipo di credenziali. In funzione del tipo di credenziali l'autenticazione potrà essere quindi basata su:
 - Password
 - Certificati
 - Ticket Kerberos
 - Smart-card

Password

- E' la modalità più semplice da implementare e di fatto non ci sono software che non permettano l'autenticazione basata su coppia di username/password
- E' anche la meno gestibile in quanto un utente avrà in sistemi differenti differenti coppie di username/password
 - E' possibile «centralizzare» il gestore di credenziali, ma bisogna stare attenti alle credenziali che viaggiano sulla rete

- E' la modalità con cui avviene l'autenticazione tra «pari» (o tra client e serve) in SSL
- Crittografia asimmetrica (chiavi pubblica e privata)
 - Il certificato è di fatto la chiave pubblica firmata da una CA che ne definisce anche la validità temporale
 - L'identificazione avviene decifrando –con la chiave pubblica– qualcosa che è stata cifrata con la chiave privata

Kerberos

- Il sistema kerberos si basa sull'emissione di «credenziali di breve durata» i *Ticket-Granting-Ticket* (TGT) ed i *Ticket-Granting-Service* (TGS)
- L'utente ottiene il TGT da un *Key Distribution Center* (KDC) impiegando «username» e «password»
- Il KDC viene contattato via rete, ma il sistema è fatto in modo da non far mai viaggiare la password sulla rete
- L'utente usa i TGT per ottenere dal KDC i TGS con i quali accede ai vari servizi
- Tutto questo permette il *Single Sign-On* (SSO) in quanto l'utente deve acquisire il TGT solo una volta (all'interno della validità temporale del TGT)

Smart-card

- E' una evoluzione del sistema di autenticazione basato su certificati
- La *smart-card* contiene sia il certificato sia la chiave privata alla quale l'utente può accedere solo usando una ulteriore chiave (un PIN)
- L'autenticazione via *smart-card* permette anche il SSO
 - Il sistema collega l'utente alla *smart-card* attraverso il certificato presente sia nella *smart-card* che nel proprio DB
 - L'autenticazione avviene attraverso l'accesso alla chiave privata
 - Il sistema può mantenere in memoria, fino a quando la *smart-card* è inserita il PIN per accedere alla chiave privata
 - E' possibile usare smart-card in un ambiente Kerberos ed ottenere un TGT partendo dal certificato (PKINIT)

- Single Sign-On è una delle parole-chiave dell'IT negli ultimi anni
 - Motivazioni semplicemente legate alla gestione
 - Più importanti motivazioni di sicurezza
- Due paradigmi
 - Principal centralizzati
 - Kerberos
 - Principal distribuiti
 - Certificati

- Come altri vendors (leggi MS) RedHat ha messo in piedi un eco-sistema completo per la gestione e l'utilizzo di Identità Digitali
 - RedHat IdM (Identity Management) o IPA
- Si basa su
 - Kerberos
 - Certificati/Smart-card
 - LDAP
- Interoperabile con MS Active Directory
- Richiede IPv6

Ecosistema RH VS Standards

- A differenza di MS AD, essendo comunque RH un sistema operativo «aperto», conserva interoperabilità con pezzi di infrastruttura «standard»
 - Kerberos: MIT o heimdal o AD
 - LDAP: 389DS o OpenLDAP o qualunque altra implementazione
 - X.509: OpenSSL

- Infrastruttura server-client
- Server o KDC
 - yum install krb5-server
 - libevent
 - libkadm5
 - libverto-tevent
 - words
- Client o workstation
 - yum install krb5-workstation
 - libkadm5

- `/etc/krb5.conf`

Configuration snippets may be placed in this directory as well
`includedir /etc/krb5.conf.d/`

`[logging]`

`default = FILE:/var/log/krb5libs.log`

`kdc = FILE:/var/log/krb5kdc.log`

`admin_server = FILE:/var/log/kadmind.log`

`[libdefaults]`

`dns_lookup_realm = false`

`ticket_lifetime = 24h`

`renew_lifetime = 7d`

`forwardable = true`

`rdns = false`

`default_realm = EXAMPLE.COM`

`default_ccache_name = KEYRING:persistent:%{uid}`

Configurazione del KDC II

- `/etc/krb5.conf`

```
[realms]
EXAMPLE.COM = {
  kdc = vbox102.example.com
  admin_server =vbox102.example.com
}
```

```
[domain_realm]
.example.com = EXAMPLE.COM
example.com = EXAMPLE.COM
```

Configurazione del KDC III

- `/var/kerberos/krb5kdc/kdc.conf`

```
[kdcdefaults]
```

```
kdc_ports = 88
```

```
kdc_tcp_ports = 88
```

```
[realms]
```

```
EXAMPLE.COM = {
```

```
#master_key_type = aes256-cts
```

```
acl_file = /var/kerberos/krb5kdc/kadm5.acl
```

```
dict_file = /usr/share/dict/words
```

```
admin_keytab = /var/kerberos/krb5kdc/kadm5.keytab
```

```
supported_encetypes = aes256-cts:normal aes128-cts:normal \
```

```
des3-hmac-sha1:normal arcfour-hmac:normal camellia256-cts:normal \
```

```
camellia128-cts:normal des-hmac-sha1:normal des-cbc-md5:normal \
```

```
des-cbc-crc:normal
```

```
}
```

Inizializzazione del KDC

- `kdb5_util create -s`
 - richiede la master password con cui cifra il DB
 - produce il DB in `/var/kerberos/krb5kdc/`
 - `principal`
 - `principal.kadm5`
 - `principal.kadm5.lock`
 - `principal.ok`
- `systemctl start krb5kdc`
- `systemctl enable krb5kdc`
 - `multi-user.target.wants/krb5kdc.service → krb5kdc.service`
- `systemctl start kadmind`
- `systemctl enable kadmind`
 - `multi-user.target.wants/kadmind.service → kadmind.service`

Gestione dei principal Kerberos

- Un principal Kerberos è un'entry nel DB del KDC, alla quale viene associato un gruppo di chiavi (hash della password).
- Utente
 - password vera, mantenuta dall'utente attraverso gli usuali comandi
- Altro (host, servizio)
 - password «random» normalmente caricata in un file
 - /etc/krb5.keytab per l'host
 - dipendente dalla configurazione del servizio

- kadmin
 - Programma di amministrazione del KDC, presente nel client Kerberos (krb5-workstation)
 - Parla via rete con il daemon kadmind che ascolta sulla porta 749
 - Richiede autenticazione Kerberos
 - Azioni soggette ad autorizzazioni definite dalle ACL presenti nel file /var/kerberos/krb5kdc/kadm5.acl
- kadmin.local
 - Programma presente nel KDC (krb5-server)
 - Utilizzabile da root senza autenticazione Kerberos

Creazione di principal utente

```
[root@vbox102 ~]# kadmin.local  
Authenticating as principal root/admin@EXAMPLE.COM with password.
```

```
kadmin.local: addprinc enrico  
WARNING: no policy specified for enrico@EXAMPLE.COM; defaulting to no policy  
Enter password for principal "enrico@EXAMPLE.COM":  
Re-enter password for principal "enrico@EXAMPLE.COM":  
Principal "enrico@EXAMPLE.COM" created.
```

```
kadmin.local: addprinc enrico/admin  
WARNING: no policy specified for enrico/admin@EXAMPLE.COM; defaulting to no policy  
Enter password for principal "enrico/admin@EXAMPLE.COM":  
Re-enter password for principal "enrico/admin@EXAMPLE.COM":  
Principal "enrico/admin@EXAMPLE.COM" created.
```


Il principal utente

kadmin.local: getprinc enrico
Principal: enrico@EXAMPLE.COM
Expiration date: [never]
Last password change: Thu Apr 19 08:22:55 CEST 2018
Password expiration date: [never]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 0 days 00:00:00
Last modified: Thu Apr 19 08:22:55 CEST 2018 (root/admin@EXAMPLE.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0
Number of keys: 8
Key: vno 1, aes256-cts-hmac-sha1-96
Key: vno 1, aes128-cts-hmac-sha1-96
Key: vno 1, des3-cbc-sha1
Key: vno 1, arcfour-hmac
Key: vno 1, camellia256-cts-cmac
Key: vno 1, camellia128-cts-cmac
Key: vno 1, des-hmac-sha1
Key: vno 1, des-cbc-md5
MKey: vno 1
Attributes:
Policy: [none]

Il principal utente

kadmin.local: getprinc enrico
Principal: enrico@EXAMPLE.COM
Expiration date: [never]
Last password change: Thu Apr 19 08:22:55 CEST 2018
Password expiration date: [never]
Maximum ticket life: 1 day 00:00:00
Maximum renewable life: 0 days 00:00:00
Last modified: Thu Apr 19 08:22:55 CEST 2018 (root/admin@EXAMPLE.COM)
Last successful authentication: [never]
Last failed authentication: [never]
Failed password attempts: 0

Number of keys: 8
Key: vno 1, aes256-cts-hmac-sha1-96
Key: vno 1, aes128-cts-hmac-sha1-96
Key: vno 1, des3-cbc-sha1
Key: vno 1, arcfour-hmac
Key: vno 1, camellia256-cts-cmac
Key: vno 1, camellia128-cts-cmac
Key: vno 1, des-hmac-sha1
Key: vno 1, des-cbc-md5
MKey: vno 1

Attributes:

Policy: [none]

Il principal di host o di servizio

```
kadmin.local: addprinc -randkey host/vbox102.example.com  
WARNING: no policy specified for host/vbox102.example.com@EXAMPLE.COM; defaulting to no policy  
Principal "host/vbox102.example.com@EXAMPLE.COM" created.
```

```
kadmin.local: getprinc host/vbox102.example.com  
Principal: host/vbox102.example.com@EXAMPLE.COM
```

...omissis...

```
Number of keys: 8  
Key: vno 1, aes256-cts-hmac-sha1-96  
Key: vno 1, aes128-cts-hmac-sha1-96  
Key: vno 1, des3-cbc-sha1  
Key: vno 1, arcfour-hmac  
Key: vno 1, camellia256-cts-cmac  
Key: vno 1, camellia128-cts-cmac  
Key: vno 1, des-hmac-sha1  
Key: vno 1, des-cbc-md5  
MKey: vno 1  
Attributes:  
Policy: [none]
```

kadmin.local: ktadd host/vbox102.example.com

Entry for principal host/vbox102.example.com with kvno 2,
encryption type aes256-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/vbox102.example.com with kvno 2,
encryption type aes128-cts-hmac-sha1-96 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/vbox102.example.com with kvno 2,
encryption type des3-cbc-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/vbox102.example.com with kvno 2,
encryption type arcfour-hmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/vbox102.example.com with kvno 2,
encryption type camellia256-cts-cmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/vbox102.example.com with kvno 2,
encryption type camellia128-cts-cmac added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/vbox102.example.com with kvno 2,
encryption type des-hmac-sha1 added to keytab FILE:/etc/krb5.keytab.
Entry for principal host/vbox102.example.com with kvno 2,
encryption type des-cbc-md5 added to keytab FILE:/etc/krb5.keytab.

Acquire il TGT

```
[root@vbox102 ~]# tail -f /var/log/krb5kdc.log &
```

```
[root@vbox102 ~]# kinit enrico
```

```
Apr 19 09:15:12 vbox102.example.com krb5kdc[10203](info):
```

```
AS_REQ (8 etypes {18 17 20 19 16 23 25 26}) 192.168.56.102:
```

```
ISSUE: authtime 1524122112, etypes {rep=18 tkt=18 ses=18},
```

```
enrico@EXAMPLE.COM for krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

```
Password for enrico@EXAMPLE.COM:
```

```
[root@vbox102 ~]# klist -fe
```

```
Ticket cache: KEYRING:persistent:0:0
```

```
Default principal: enrico@EXAMPLE.COM
```

```
Valid starting      Expires          Service principal
```

```
04/19/2018 09:15:12 04/20/2018 09:15:12 krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

```
Flags: FI, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96
```


Utilizzo del TGT per il SSO

```
[root@vbox102 ~]# ssh enrico@vbox102
The authenticity of host 'vbox102 (192.168.56.102)' can't be established.
ECDSA key fingerprint is SHA256:uSdPaRGdwZZygaKBgBmNqGUWNI5f2sDmqo8K8r3X8xc.
ECDSA key fingerprint is MD5:a3:29:65:ca:27:ce:44:2f:b2:11:3a:a3:1a:16:b8:77.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'vbox102' (ECDSA) to the list of known hosts.
Apr 19 09:24:07 vbox102.example.com krb5kdc[10203](info):
  TGS_REQ (8 etypes {18 17 20 19 16 23 25 26}) 192.168.56.102:
  ISSUE: authtime 1524122112, etypes {rep=18 tkt=18 ses=18},
  enrico@EXAMPLE.COM for host/vbox102.example.com@EXAMPLE.COM
Last login: Thu Apr 19 06:37:57 2018 from vbox102.example.com
[enrico@vbox102 ~]$

[enrico@vbox102 ~]$ klist -fe
Ticket cache: KEYRING:persistent:1000:krb_ccache_LdWUMqM
Default principal: enrico@EXAMPLE.COM

Valid starting      Expires            Service principal
04/19/2018 06:23:53 04/20/2018 06:23:53  krbtgt/EXAMPLE.COM@EXAMPLE.COM
    Flags: FI, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96
```

```
[enrico@vbox102 ~]$ exit
logout
Connection to vbox102 closed.
[root@vbox102 ~]# klist -fe
Ticket cache: KEYRING:persistent:0:0
Default principal: enrico@EXAMPLE.COM
```

Valid starting	Expires	Service principal
04/19/2018 09:24:07	04/20/2018 09:15:12	host/vbox102.example.com@EXAMPLE.COM
Flags: FT, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96		
04/19/2018 09:15:12	04/20/2018 09:15:12	krbtgt/EXAMPLE.COM@EXAMPLE.COM
Flags: FI, Etype (skey, tkt): aes256-cts-hmac-sha1-96, aes256-cts-hmac-sha1-96		



Autenticazione

Esercitazione

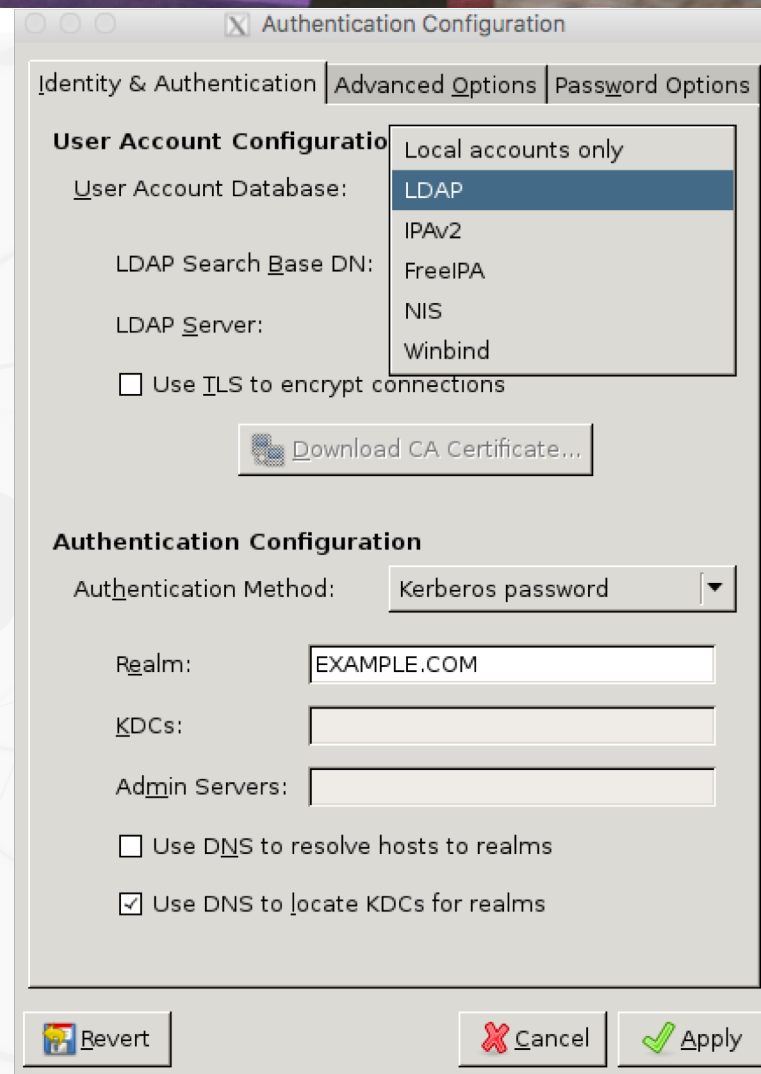
- <https://wiki.infn.it/cn/ccr/formazione/centos7/2018-11/autnx-1>
- **Esercizio 1**
 - Installazione e configurazione di un Kerberos KDC nel REALM EXAMPLE.COM
- **Esercizio 2**
 - Definire un principal di utente e configurare il sistema in modo che tale utente possa effettuare il login con autenticazione Kerberos
- **Esercizio 3**
 - Permettere l'accesso via ssh in modalità Single Sign-On (usando GSSAPI)

- In RH7 la configurazione dell'autorizzazione all'accesso al sistema (ossia la definizione degli account) si effettua usando uno strumento apposito, disponibile in tre varianti
- `authconfig`
 - interfaccia command-line
- `authconfig-gtk`
 - Interfaccia grafica
- `authconfig-tui`
 - interfaccia «testo-grafica», deprecata

- Nelle sue tre varianti, authconfig, permette di configurare sia l'autorizzazione che l'autenticazione
- authconfig command-line è la variante più completa
 - permette di configurare *tutte* le possibili opzioni dei vari sistemi di di AuthN ed AuthZ
 - permette di creare un backup della configurazione corrente (e ripristinarlo)
 - fornisce supporto per la creazione di file kickstart.
- Una grande quantità di opzioni
 - Il riferimento è la man-page
- **WARNING!!!**
 - Non verifica che la configurazione che si ottiene con la definizione delle opzioni inserite a linea di comando sia completa ed accurata
 - opzione --test

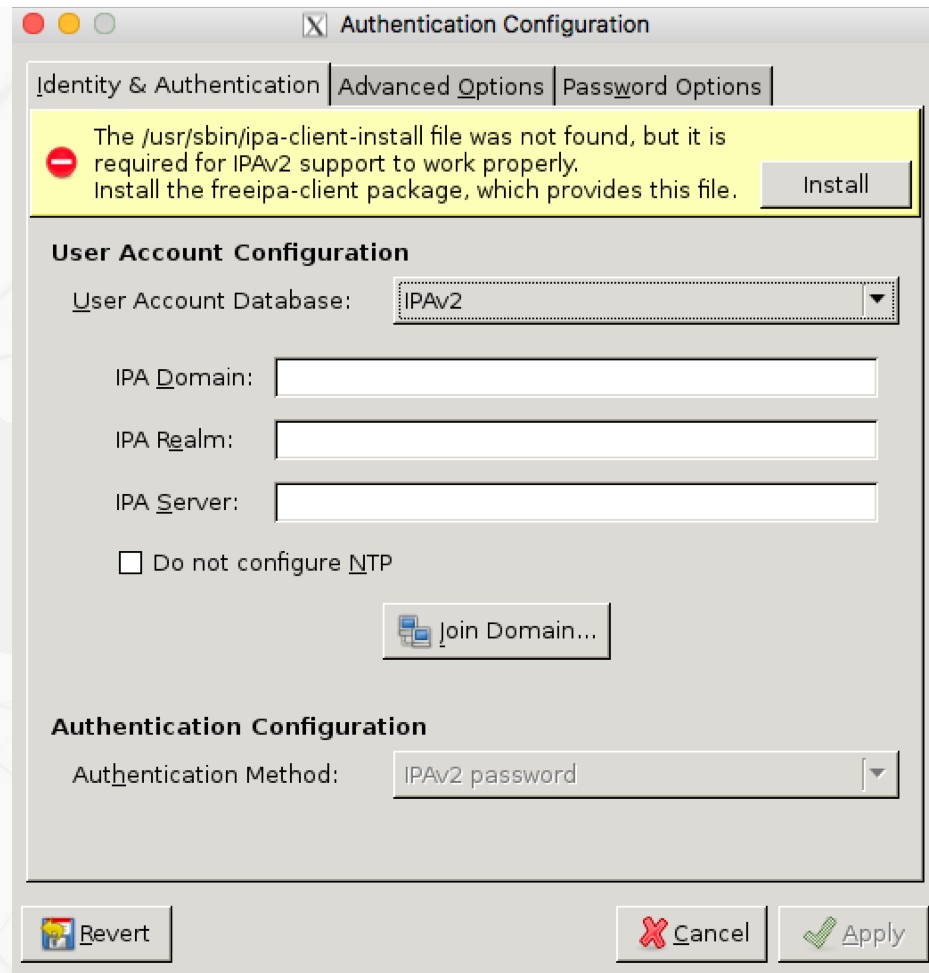
- Il pacchetto authconfig-gtk non è installato per default
- yum install authconfig-gtk
 - Dipende da una cinquantina di pacchetti
 - Per poterlo usare da sessione ssh bisogna installare anche xauth ed i font
 - yum install xauth
 - yum install dejavu-lgc-sans-fonts
 - Se si vuole configurare l'autorizzazione con LDAP o freeIPA è necessario anche il relativo modulo PAM
 - yum install nss-pam-ldapd

- Identity & Authentication
 - Configurazione del DB delle identità e metodo di autenticazione
- Advanced Options
 - smart-card o impronte digitali
- Password Options
 - vincoli sulle password



The screenshot shows the 'Authentication Configuration' window with three tabs: 'Identity & Authentication', 'Advanced Options', and 'Password Options'. The 'Identity & Authentication' tab is active. It contains a 'User Account Configuration' section with a dropdown menu for 'User Account Database' showing options: 'Local accounts only', 'LDAP' (selected), 'IPAv2', 'FreeIPA', 'NIS', and 'Winbind'. Below this are fields for 'LDAP Search Base DN:' and 'LDAP Server:', and a checkbox for 'Use ILS to encrypt connections'. A 'Download CA Certificate...' button is also present. The 'Authentication Configuration' section below has a dropdown for 'Authentication Method' set to 'Kerberos password'. It includes fields for 'Realm:' (EXAMPLE.COM), 'KDCs:', and 'Admin Servers:'. At the bottom, there are checkboxes for 'Use DNS to resolve hosts to realms' (unchecked) and 'Use DNS to locate KDCs for realms' (checked). The window has 'Revert', 'Cancel', and 'Apply' buttons at the bottom.

- Ecosistema IdM Red Hat che ricorda in modo fin troppo esplicito AD di Microsoft



Authentication Configuration

Identity & Authentication | Advanced Options | Password Options

The /usr/sbin/ipa-client-install file was not found, but it is required for IPAv2 support to work properly. Install the freeipa-client package, which provides this file. **Install**

User Account Configuration

User Account Database: IPAv2

IPA Domain:

IPA Realm:

IPA Server:

☐ Do not configure NTP

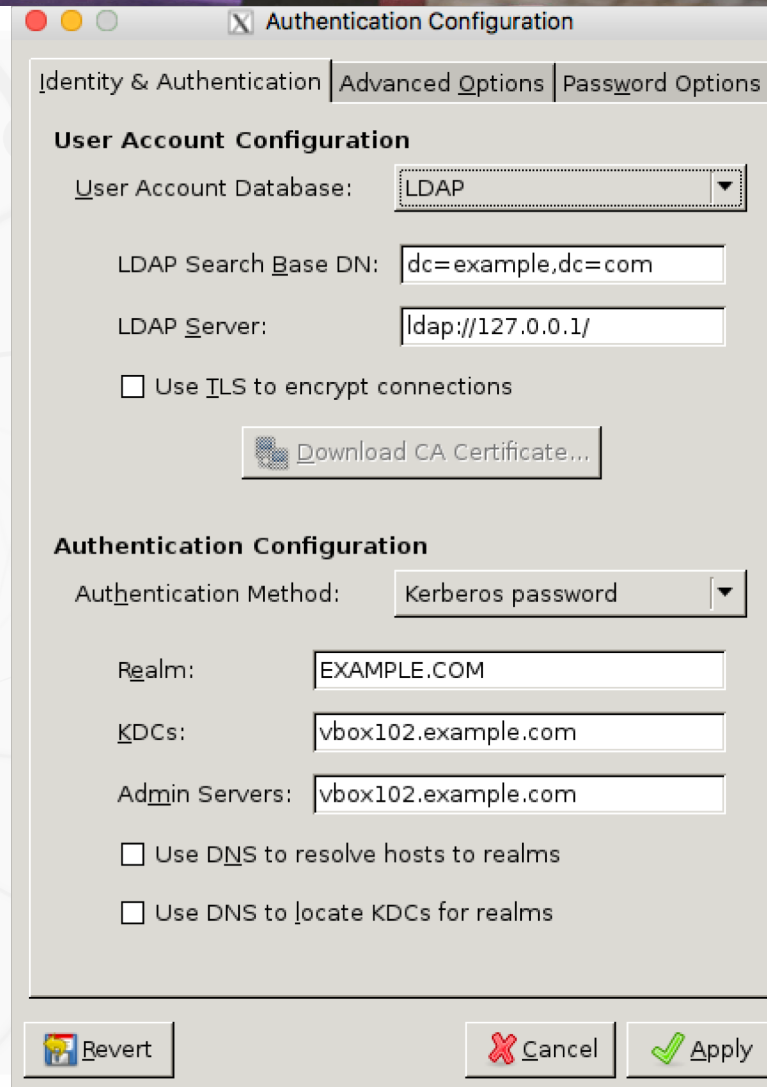
Join Domain...

Authentication Configuration

Authentication Method: IPAv2 password

Revert **Cancel** **Apply**

- Database utenti su LDAP
- Autenticazione
 - LDAP
 - Kerberos
- La configurazione via command-line permette di definire opzioni non accessibili da UI
 - RFC2307bis



Authentication Configuration

Identity & Authentication | Advanced Options | Password Options

User Account Configuration

User Account Database: LDAP

LDAP Search Base DN: dc=example,dc=com

LDAP Server: ldap://127.0.0.1/

☐ Use TLS to encrypt connections

Download CA Certificate...

Authentication Configuration

Authentication Method: Kerberos password

Realm: EXAMPLE.COM

KDCs: vbox102.example.com

Admin Servers: vbox102.example.com

☐ Use DNS to resolve hosts to realms

☐ Use DNS to locate KDCs for realms

Revert Cancel Apply

Accesso alle informazioni

- Quando un sistema unix fa riferimento ad un server di informazioni esterno, come un IdM o una directory LDAP, esso effettua una richiesta ogni volta che è necessario avere una informazione.
- Se le informazioni posix di un utente sono in LDAP, ogni volta che il sistema deve risolvere un nome, fa una query LDAP
- Ad esempio, le informazioni sul filesystem locale sono uidNumber e gidNumber. Se voglio sapere a chi appartiene un file, devo effettuare due query LDAP (una per uidNumber ed una per gidNumber)
- Vi lascio immaginare cosa succede ad un comando del tipo
 - `ls -lR`

- La soluzione a problemi di questo tipo è implementare una cache locale
- Name Service Cache Daemon (NSCD) è una vecchia implementazione di una cache
- Sostituita dal nuovo SSSD

- **System Security Services Daemon**
 - Servizio che accede a directory e meccanismi di autenticazione remoti, collegando il sistema locale (client SSD) ad un sistema esterno di back-end (SSSD provider).
 - Il sistema esterno può essere di vari tipi
 - una directory LDAP, un Identity Management (IdM), un dominio Active Directory (AD), un realm Kerberos
- **Per fare questo SSSD**
 - Collega il client ad un Identity Store esterno recuperando da esso le informazioni relative all'autenticazione
 - Usa tali informazioni per popolare una cache locale degli utenti e delle loro credenziali
- Gli utenti del sistema si autenticano usando le credenziali «remote»
- SSSD non crea utenti locali, ma usa le informazioni del sistema esterno
- Permette ad altri servizi del sistema (come NSS o PAM) di utilizzare dati nella sua cache

Benefici di SSD

- Riduce il carico sui server di autenticazione e delle identità
 - Quando è necessario accedere alle informazioni, il client SSSD effettua la richiesta ad SSSD che prima di tutto verifica nella cache, richiedendo al server solo le informazioni che non sono in cache
- Permette l'autenticazione su sistemi fuori rete
 - SSSD può essere configurato per mantenere una cache anche delle credenziali dell'utente che può quindi autenticarsi anche se il server di autenticazione non è raggiungibile
- Razionalizzazione degli account
 - Non è necessario avere un account locale per l'accesso quando il sistema non è in rete
 - Grazie al caching ed all'autenticazione off-line gli utenti possono accedere a risorse remote usando le credenziali che SSSD ha messo nella cache del sistema locale

Configurazione di SSSD

- SSSD legge la configurazione
 - dal file `/etc/sss/sss.conf`
 - da qualunque file `*.conf` presente in `/etc/sss/conf.d/` in ordine alfabetico
 - se uno stesso parametro è presente in più file di configurazione, il valore usato da SSSD è l'ultimo letto

- Come per l'accesso al sistema, anche per SSSD è possibile autenticare via Kerberos ed autorizzare via LDAP

```
[domain/example.com]  
cache_credentials = true  
id_provider = ldap
```

```
auth_provider = krb5  
chpass_provider = krb5
```

```
ldap_uri = ldap://vbox102.example.com  
ldap_search_base = dc=example,dc=com
```

```
krb5_server = vbox102.example.com  
krb5_backup_server = vbox102.example.com  
krb5_passwd = vbox102.example.com  
krb5_realm = EXAMPLE.COM
```

- E' possibile restringere l'accesso ad un host in funzione di valori di attributi

```
[domain/le.infn.it]
id_provider = ldap
access_provider = ldap
ldap_access_filter = (isMemberOf=g:infn:le:scr::ag:m)
ldap_uri = ldaps://dsa.le.infn.it,ldaps://dsb.le.infn.it,ldaps://ds.infn.it
ldap_search_base = dc=le,dc=infn,dc=it
ldap_tls_cacert = /etc/ssl/certs/ca-certificates.crt

auth_provider = krb5
chpass_provider = krb5
krb5_realm = LE.INFN.IT
dns_discovery_domain = LE.INFN.IT
krb5_server = kdc.le.infn.it
```

AuthN & AuthZ

Esercitazione

- <https://wiki.infn.it/cn/ccr/formazione/centos7/2018-11/autnz-1>
- **Esercizio 1**
 - Configurare un client in modo che permetta la login con Autenticazione Kerberos ed Autorizzazione LDAP
- **Esercizio 2**
 - Configurare SSSD in modo da permettere l'accesso con autenticazione Kerberos ed Autorizzazione LDAP