AUTENTICAZIONE

Autenticazione

- Si intende una serie di problemi e operazioni diverse che vanno dall'identificazione e **autenticazione** (**chi sei?**) alle **autorizzazioni** vere e proprie (**cosa vuoi e cosa puoi fare**).

- **ATTENZIONE**: e sempre bene sapere chi accede al sistema e cosa può fare!!! Ecco perchè **NON DOVETE CEDERE LOGIN E PASSWORD A NESSUNO.** Vi ricordo, in quanto utenti, che la sicurezza è anche un vostro preciso dovere.

Autenticazione base

- Storicamente, i sistemi UNIX usavano il contenuto del file `/etc/passwd` per entrambi gli scopi (chi sei e cosa puoi fare). Il file contiene tante righe quanti sono gli utenti del sistema, ed ognuna di queste aveva la forma:

  ```
  username:encpassword:UID:GID:gecos:homedirectory:
  shell
  ```

  ```
  gsottana:xZELC6ObSje96:1959:1000:Giorgia
  Sottana,,,:/home/gsottana:/bin/bash
  ```

- Poiché alcune delle informazioni contenute nel file (ad esempio la corrispondenza **username** → **UID**) devono essere disponibili a tutti gli utenti di sistema (ed in particolare alle loro applicazioni), il file era (ed è) pubblicamente accessibile in lettura. La password era protetta dalla (supposta) robustezza dell'algoritmo di hashing usato, originariamente basato sull'algoritmo crittografico DES(3). **XD**

Autenticazione Base

- I file `/etc/group` contiene la lista dei gruppi del sistema e dei relativi **GID**, nonché, per ognuno di essi, l'elenco dei suoi membri, ad esclusione di quelli che hanno il suo **GID** come loro gruppo primario. La sintassi di ogni riga del file

  `groupname:`*`encpassword`*`:GID:listamembri`

- Gli UNIX moderni non usano più direttamente il file `/etc/passwd` per la funzionalità che gli da il nome, cioè la memorizzazione della password. Questo ruolo è stato assunto dal file `/etc/shadow`, che, servendo solo per la fase di autenticazione, non è pubblicamente leggibile.
  Il formato di `/etc/shadow` è il seguente:

  ```
  username:encpassword:ultimocambio:etàminima:et
  àmassima:periodoavviso:periodoinattività:datas
  cadenza:altro
  ```

Autenticazione base - svantaggi

- Il problema principale di questa soluzione è la scarsa l'elasticità!
- Originariamente, tutti i file precedentemente indicati erano gestiti direttamente da programmi che ne richiedevano la consultazione o la manipolazione (esempio `vipw`).
- Ad ogni modifica dello standard dei file era necessario modificare anche i programmi che li gestivano.
- Allo stesso modo, sistemi che storicamente permettevano di centralizzare la gestione degli utenti su più sistemi, come **NIS**, richiedevano una nuova versione delle stesse utility.

Nuove soluzioni: NSS e PAM

- Per ovviare a questa situazione, nel tempo sono nati sistemi che permettono, tramite l'uso di **componenti modulari**, di aggiungere meccanismi diversi di autenticazione e di gestione delle informazioni sugli utenti, senza per questo dover modificare i programmi che ne fanno uso. In particolare:
  - **nss (Name Service Switch)**: per la gestione delle informazioni sugli utenti e per “altre” mappe che associano nomi ad altre informazioni.
  - **PAM (Pluggable Authentication Modules)**: per i meccanismi di autenticazione.

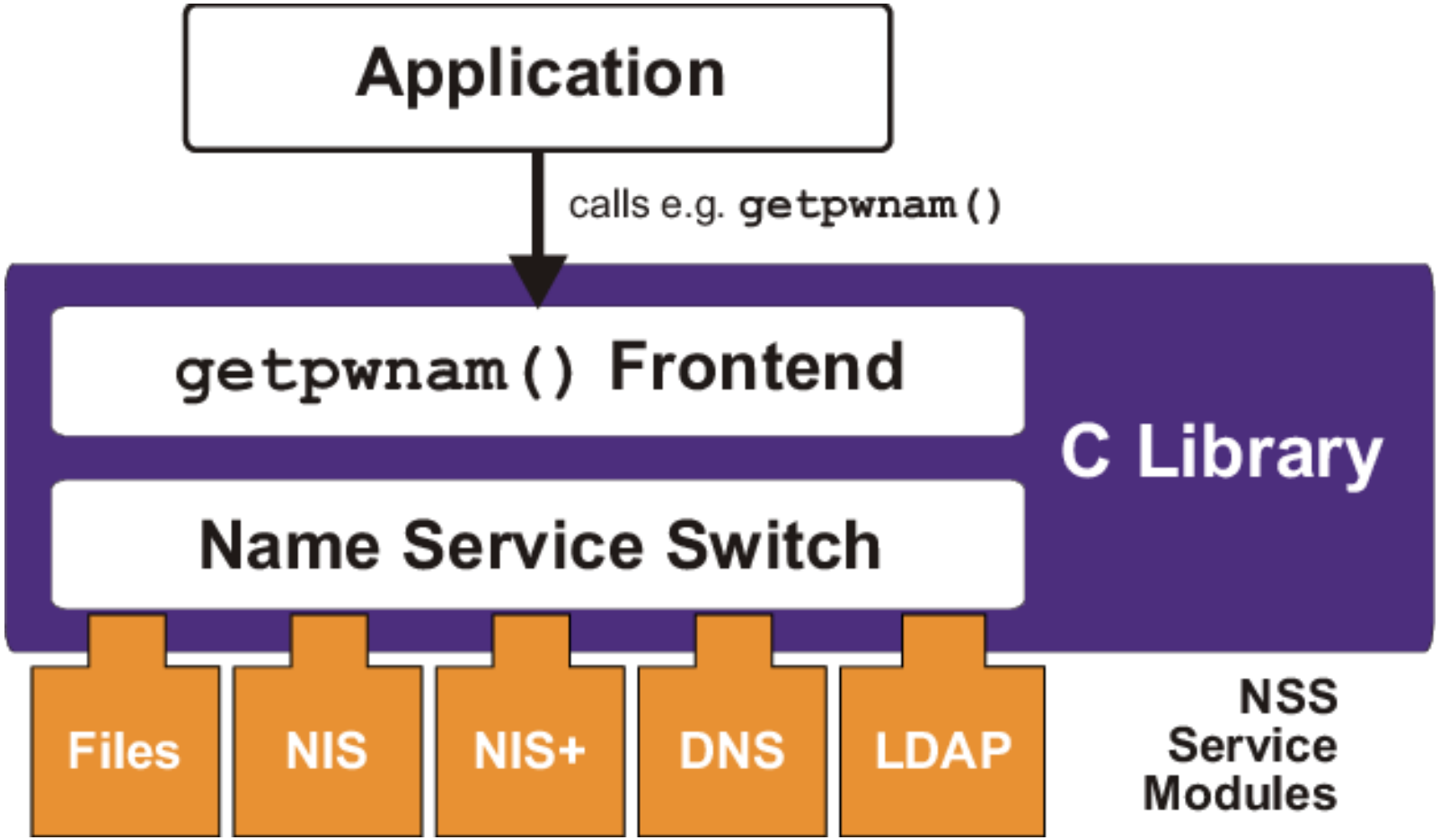
Name Service Switch - NSS

- Il sistema **nss**, in Linux, è parte dell'implementazione delle librerie C del progetto GNU.
- **Fornisce i dati dei system database** (tra cui `passwd` e `group`) **alle applicazion**i. Le sorgenti dei dati, oltre ai file omonimi, possono essere le più disparate (ad esempio **mysql**, **ldap**, ecc).
- Dei **moduli**, anche di terze parti, si occupano di tradurre richieste e risposte a/da questi database affinché possano essere effettuate sulla fonte prescelta, come ad esempio una directory LDAP o un database SQL.
- La configurazione di **nss** avviene principalmente tramite il file `/etc/nsswitch.conf`, mentre moduli aggiuntivi si trovano, solitamente, nei pacchetti Debian/Ubuntu dal nome `libnss-*`.
- Per l'installazione in Ubuntu/Debian (nel caso di autenticazione ldap):

```
# apt install libnss-ldap
```

Name Service Switch - NSS

- **NSS** fornisce delle funzioni che fanno da interfaccia tra le applicazioni e i sistemi di autenticazione per la gestione delle informazioni utente.




```
The getpwnam() function returns a pointer to a structure
containing the broken-out fields of the record in the
password database (e.g., the local password file /etc/psswd,
NIS, and LDAP) that matches the username name.
```

Name Service Switch - Esempio

```
# /etc/nsswitch.conf
#
passwd:         compat ldap
group:          compat ldap
shadow:         compat ldap
gshadow:        files

automount       ldap

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

`compat`: indica che le informazioni sono prese da file presenti in /etc.

`ldap`: indica che le informazioni sono prese da un database ldap.

`files`: indica che le informazioni sono prese dal file presenti in `/etc/hosts`.

`dns`: indica che le informazioni sono prese da un server dns.

Esempio: libnss-ldap.conf

```
nss_base_passwd cn=Users,dc=win,dc=dsi,dc=unive,dc=it?sub

nss_base_shadow cn=Users,dc=win,dc=dsi,dc=unive,dc=it?sub

nss_base_group  cn=Builtin,dc=win,dc=dsi,dc=unive,dc=it?sub


# Direttive per mappare oggetti e attributi tra ldap e active directory.

nss_map_objectclass posixAccount User

nss_map_objectclass shadowAccount User

nss_map_objectclass posixGroup group

nss_map_attribute uid sAMAccountName

nss_map_attribute uniqueMember member

nss_map_attribute uidNumber uidNumber

nss_map_attribute gidNumber gidNumber

nss_map_attribute homeDirectory unixHomeDirectory

nss_map_attribute loginShell loginShell

nss_map_attribute gecos displayName

nss_map_attribute cn sAMAccountName

nss_initgroups_ignoreusers avahi,avahi-
autoipd,backup,bin,colord,daemon,dnsmasq,games,gdm,gnats,hplip,irc,kdm,kernoops,libuuid,lightdm,list,lp,mail,man,messag
ebus,news,ntp,nvidia-persistenced,proxy,pulse,root,rtkit,saned,speech-
dispatcher,sshd,statd,sync,sys,syslog,usbmux,uucp,whoopsie,www-data
```

Pluggable Authentication Modules - PAM

- Il sistema **PAM** fornisce i servizi d'autenticazione ad una o più applicazioni/servizi (**services**), che possono avere configurazioni separate. Le attività (**task**) di autenticazione sono suddivise in 4 gruppi detti **management group:**
  - **account**: si occupa di verificare eventuali requisiti che l'account deve rispettare, ad es. se è scaduto o se ha il permesso o divieto di accedere.
  - **auth**(-entication): si occupa di autenticare l'utente e di acquisire le credenziali necessarie.
  - **password**: si occupa di aggiornare/modificare le informazioni di autenticazione. Un classico esempio è proprio l'operazione di cambio password.
  - **session**: si occupa di gestire le attività che devono essere effettuate in apertura ed in chiusura di una sessione, ad esempio
    - all'accesso aggiorna i file `utmp` e `wmtp`, fa partire una sessione `ssh-agent` …
    - all'uscita aggiorna i file `utmp` e `wmtp` termina la sessione `ssh-agent` …

- `Per approfondire:`
  `https://www.freebsd.org/doc/en_US.ISO8859-1/articles/pam/pam-essentials.html`

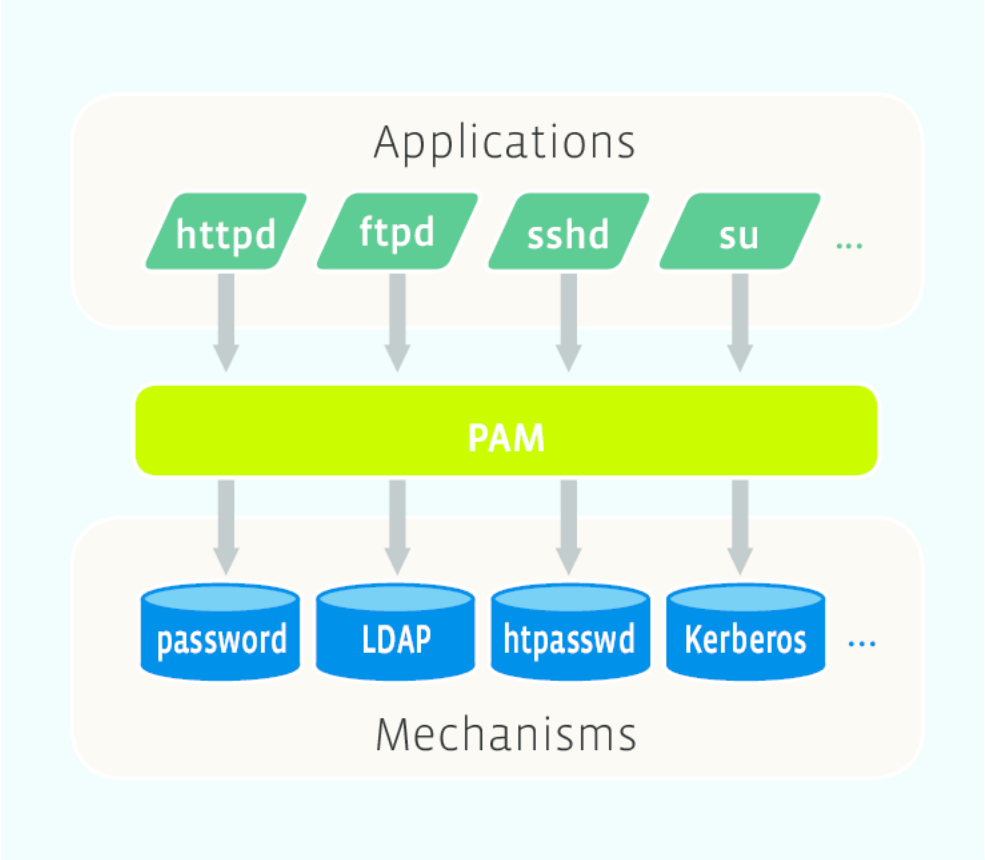
Pluggable Authentication Modules - PAM

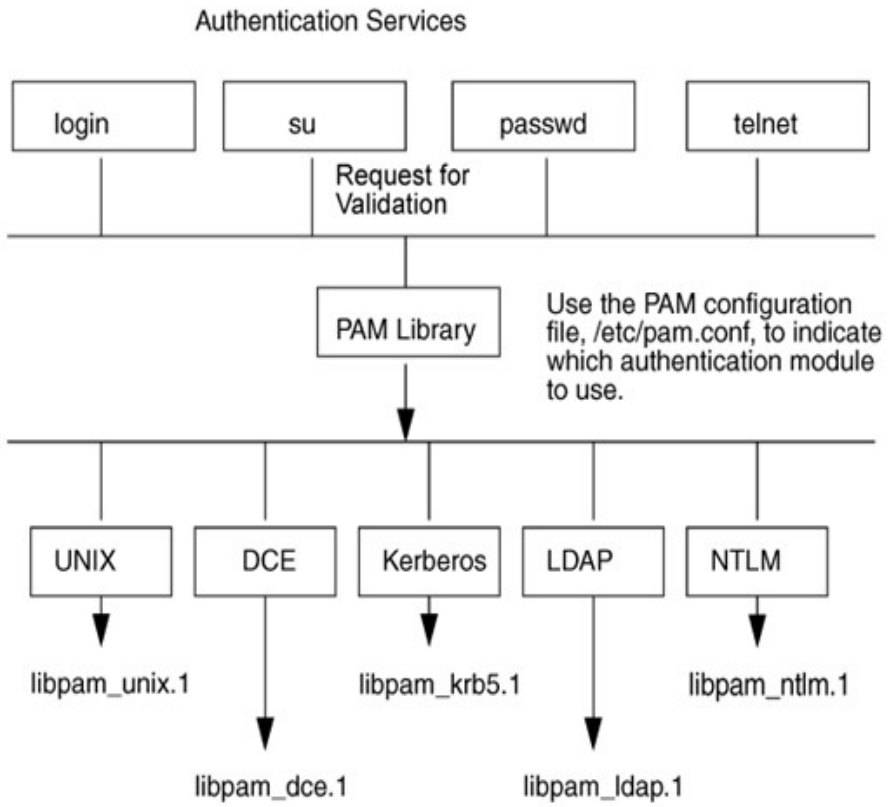
- *Per ogni servizio ed ognuna di queste attività si possono specificare una sequenza di operazioni da eseguire tramite l'uso di moduli (o plugin) intercambiabili*.
- Vi sono molti plugin di terze parti, che in Debian/Ubuntu sono forniti in pacchetti dal nome `libpam-*` e si installano così (nel caso di autenticazione con ldap):

```
# apt install libpam-ldap
```

- La configurazione si trova nei file nella directory `/etc/pam.d/`, oppure, anche se sconsigliato, solo nel file `/etc/pam.conf`.
- Nota: I file in `/etc/pam.d/common-*` indicano configurazioni di default, che devono essere importate nei file che specificano eventuali modifiche del comportamento standard.

Pluggable Authentication Modules - PAM





NSS e PAM – Esempio: file common-account e common-password

```
# /etc/pam.d/common-account - authorization settings common to all services
#
account  [success=1 new_authtok_reqd=done default=ignore]       pam_unix.so
account requisite                       pam_deny.so
account required                        pam_permit.so
```

```
# /etc/pam.d/common-password - password-related modules common to all services
password requisite                      pam_cracklib.so retry=3 minlen=8 difok=3
password [success=3 default=ignore]     pam_krb5.so minimum_uid=1000 try_first_pass \
                                                                        use_authtok
password [success=2 default=ignore]     pam_unix.so obscure use_authtok try_first_pass \
                                                                        sha512
password [success=1 user_unknown=ignore default=die]    pam_ldap.so use_authtok
try_first_pass
password requisite          pam_deny.so
password required           pam_permit.so
```

NSS e PAM – Esempio: file common-auth

```
# /etc/pam.d/common-auth - authentication settings common to all services

# here are the per-package modules (the "Primary" block)

auth [success=3 default=ignore] pam_krb5.so \ minimum_uid=1000
use_first_pass keytab=/etc/krb5.pam

auth    [success=2 default=ignore]      pam_unix.so \
                nullok_secure try_first_pass

auth    [success=1 default=ignore]      pam_ldap.so \
                                        use_first_pass

# here's the fallback if no module succeeds

auth requisitepam_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

auth required pam_permit.so

# and here are more per-package modules (the "Additional" block)

# end of pam-auth-update config
```

...in windows.....

- SAM File
- https://learn.microsoft.com/it-it/windows-server/security/windows-authentication/credentials-processes-in-windows-authentication

Autenticazione centralizzata

- Fin dalla nascita delle reti, la gestione dell'accesso degli utenti ai vari host del sistema è sempre stato un **problema centrale** per l'amministratore di sistema.
- Poter gestire accessi e permessi a tutto il sistema attraverso un singolo servizio semplifica di molto la vita del Sys Admin e aiuta nella gestione sicurezza dei sistemi.
- Ad oggi, quasi tutti i sistemi che necessitano di **autenticazione centralizzata** si basano su **LDAP** per la gestione degli utenti e dei loro permessi. Spesso **LDAP** è usato in coppia con **Kerberos** per la gestione degli accessi e delle password.
- Una volta però….

NIS e NIS+

- Eoni fa….
  - **NIS**, **Network Information Services**, fu sviluppato da Sun Microsystems per centralizzare l'amministrazione di sistemi UNIX® (in origine SunOS™). Ad oggi, anche se abbandonato tutti i sistemi UNIX® like (Solaris™, HP-UX, AIX®, Linux, NetBSD, OpenBSD, FreeBSD, etc) lo supportano.
  - **NIS** in precedenza era noto come **Yellow Pages**, ma per una questione di marchi, Sun ne ha cambiato il nome. Il vecchio termine (**yp**) viene spesso ancora usato.
  - *E' un sistema client/server basato su **RPC** che permette ad un gruppo di macchine in un dominio, detto dominio NIS, di condividere un insieme comune di file di configurazione.* Questo permette ad un amministratore di sistema di installare sistemi **client NIS** con il minimo di dati di configurazione e di aggiungere, rimuovere o modificare dati di configurazione da una singola macchina.
  - Sostanzialmente permetteva di condividere i file `/etc/passwd`, `/etc/group` e `/etc/shadow` tra tutte le macchine "iscritte" al **dominio NIS:** le modifiche si propagavano da un master alle macchine del dominio..
  - **Insicuro!!!!! viaggiava tutto in chiaro…. Diventando root di un client (usando ad esempio un exploit del sistema operativo) si riusciva facilmente ad accedere a tutte le informazioni del database NIS…. O più semplicemente, con su -, diventare un utente del dominio e...**
- **Poi NIS+**
  - Struttura completamente diversa da **NIS**: sostanzialmente una struttura gerarchica con server principali e backup server, basata sempre su **RPC**.
  - **Autenticazione tramite password crittata** .
  - Non più supportato:
    - http://www.linux-nis.org/nisplus/
- Per approfondire: `https://www.freebsd.org/doc/it_IT.ISO8859-15/books/handbook/network-nis.htm`

Remote Procedure Call - RPC

- **RPC**: Attivazione da parte di un programma di una procedura o subroutine eseguita su un computer remoto ~~(quindi diverso da quello sul quale il programma viene eseguito).~~
- **RPC** *consente a un programma di eseguire procedure "a distanza" su computer remoti, accessibili attraverso una rete.*
- La proprietà principale dei servizi **RPC** è la **trasparenza**: l'esecuzione di una procedura remota deve essere analoga a quella di una procedura locale. I dettagli della comunicazione su rete devono essere "**nascosti**" (resi trasparenti) all'utilizzatore del meccanismo.
- Il paradigma **RPC** risulta particolarmente adatto per il **calcolo distribuito** basato sul modello **client-server (NFS).**

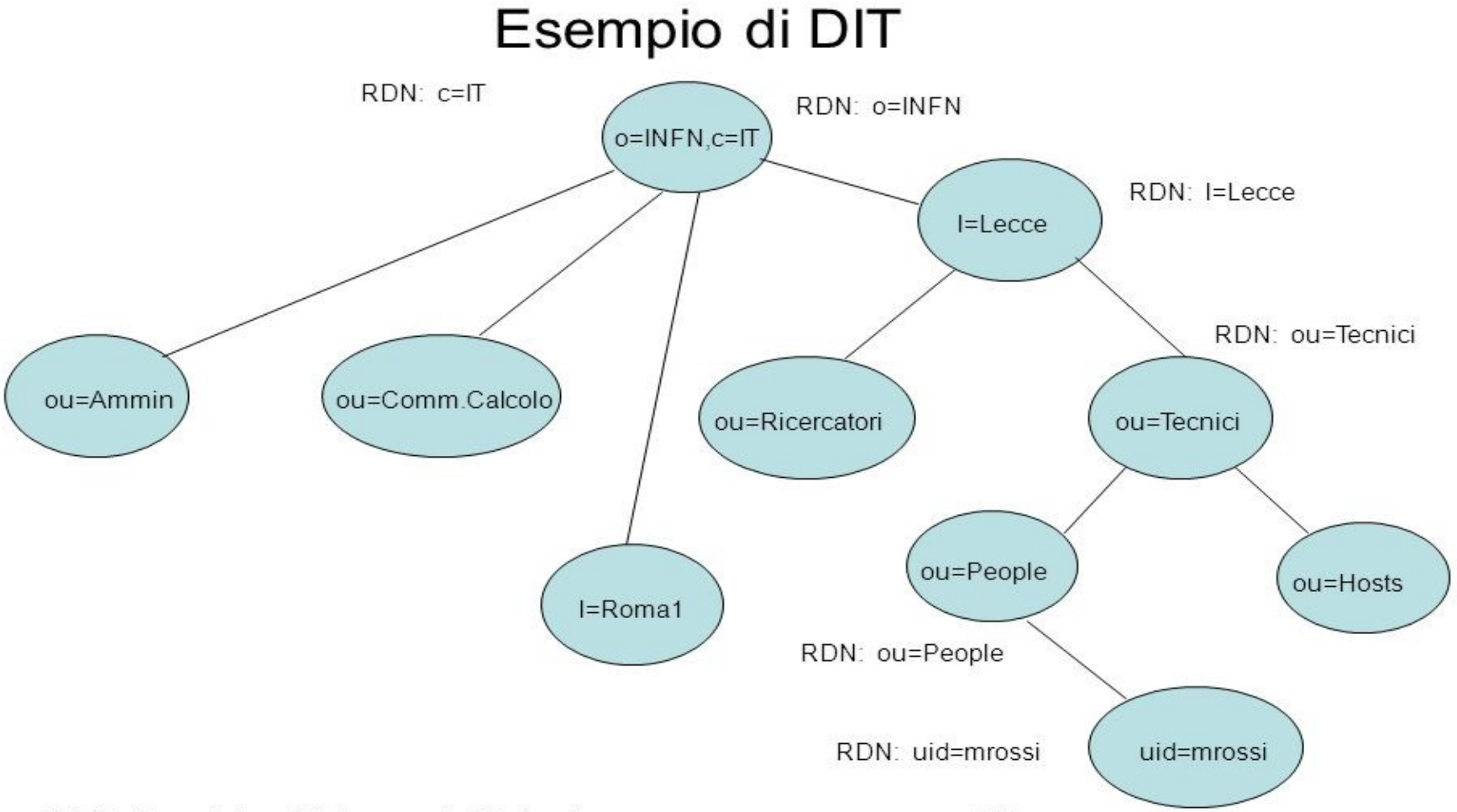
Lightweight Directory Access Protocol - LDAP

- Una **Directory** è una **base di dati che poggia su un modello gerarchico con struttura ad albero**.
- In quanto tale è, schematicamente, ben rappresentata da un albero detto **DIT (Directory Information tree)**.
- **LDAP** è **un protocollo per l'interrogazione e la manipolazione di Directory,** basato su un'architettura di tipo client/server.
- Utilizzato in ambiti in cui, **il numero delle operazioni di consultazione (lettura) può essere elevato, mentre le operazioni di modifica (scrittura) sono relativamente rare**.
- Per migliorare le performance di un sistema basato su **ldap** si usa **nscd,** demone che gestisce la cache di rete, che mantiene una cache locale per tutte le operazioni di rete configurate.

Lightweight Directory Access Protocol – LDAP - DIT

- **DIT (Directory Information Tree)**: è **una rappresentazione ad albero dei dati contenuti in un database ldap**. Rappresenta la **Directory**!
- I **nodi** rappresentano suddivisioni di vario tipo (ad esempio gli uffici di una organizzazione).
- Le **foglie** sono i dati veri e propri.
- Ogni **nodo** (**entry**) dell'albero ha un **relative distinguished name (RDN),** che è unico tra tutti i nodi fratelli.
- La sequenza (o percorso) di tutti gli **RDN** dal nodo interessato fino alla radice viene chiamato **Distinguished Name (DN)**, ed è unico per tutto il **DIT**.
- I nodi possono avere **informazioni strutturate** (ad esempio nome, cognome, email, numero di telefono…) **organizzate in attributi**, ed implementate uno o più schemi, che non sono altro che delle definizioni di tipi di dato.
- I nodi possono anche indicare dei riferimenti ad altre parti del **DIT**, o addirittura a **DIT** di altri server **LDAP** (**referral**).

Lightweight Directory Access Protocol – LDAP – DIT



Il Distinguished Name dell'utente con username mrossi è:

DN: uid=mrossi,ou=People,ou=Tecnici,l=Lecce,o=INFN,c=IT

Oppure

`mrossi@people.tecnici.lecce.infn.it`

Lightweight Directory Access Protocol – LDAP

- L'accesso alla **directory ldap** può essere **anonimo** oppure tramite **autenticazione** (**binding**). La fonte dei dati d'autenticazione può essere interna al **DIT** o esterna (ad esempio un file in `/etc`).
- Vi sono molte implementazioni di server **LDAP**. Le più note nei sistemi UNIX sono **OpenLDAP** (più usato) **e 389 Directory Server.** In Debian/Ubuntu l'implementazione di **OpenLDAP** è nel pacchetto `slapd`:

```
# apt install slapd
```

- Su **Windows LDAP** viene utilizzato in accoppiata con il protocollo **Kerberos** per la gestione dell'autenticazione utente (e non solo) in un sistema che prende il nome di **Active Directory**.

Lightweight Directory Access Protocol – LDAP - DIT

- **LDAP** è spesso usato per fornire le informazioni sugli utenti e sui gruppi ad un insieme di elaboratori. Il sistema **`nss`** dispone di un **plugin** (pacchetto Debian/Ubuntu: `libnss-ldap`) che, se attivato e configurato opportunamente, consente di usare dei **server LDAP** come fonte di dati per i vari database di sistema.
- **LDAP** può essere usato (tramite `libpam-ldap`) per l'autenticazione, facendo verificare al sistema locale se la password cifrata presente nel **DIT** (esattamente come in `/etc/passwd` o `/etc/shadow`) corrisponde a quella immessa dall'utente, oppure cercando di autenticarsi (**binding**) al **server LDAP** stesso usando quelle credenziali.
- Per installare i due pacchetti:

```
# apt install libnss-ldap libpam-ldap
```

LDAP vs rDBMS: Perchè non usare un db?

- **LDAP può essere considerato un database nosql.**
- Gli archivi dati **LDAP** sono destinati a sistemi con un numero elevato di letture rispetto alle scritture. I database **SQL** sono progettati per l'utilizzo di dati transazionali (lettura e scrittura elevate). Questo è il motivo per cui **LDAP** è un protocollo di directory. È adatto alle directory in cui vi sono molte letture e poche scritture.
- **LDAP** è caratterizzato come un servizio "scrivi una volta, leggi molte volte". Vale a dire, non ci si aspetta che il tipo **di** dati che verrebbero normalmente archiviati in un servizio **LDAP** cambi ad ogni accesso:
  - **LDAP** **NON** sarebbe adatto per conservare i record delle transazioni bancarie poiché, per loro natura, cambiano ad ogni accesso (transazione).
  - **LDAP** sarebbe tuttavia particolarmente adatto per mantenere i dettagli delle filiali bancarie, degli orari di apertura, dei dipendenti ecc..

LDAP vs rDBMS

- La risposta non è semplice ma possono essere utili le seguenti note:
  - Il calo delle prestazioni durante le scritture risiede nell'aggiornamento degli indici. Maggiore è il numero di indici (per una lettura più rapida), meno frequente sarà l'aggiornamento della directory. Rapporti di lettura:scrittura inferiori a 1.000:1 o superiori per directory LDAP ottimizzate per letture elevate.
  - La replica LDAP genera più transazioni per ogni aggiornamento, pertanto è necessario il carico di aggiornamento pratico più basso (1.000:1 o superiore).

LDAP vs rDBMS

- Se il volume di dati è grande (ad esempio > 10.000) il tempo per aggiornare anche un piccolo numero di indici potrebbe essere serio, quindi è opportuno mantenere gli aggiornamenti al livello più basso possibile (10.000:1).
- Se il volume di dati è relativamente piccolo (diciamo < 1.000 record), gli indici sono modesti e non viene utilizzata alcuna replica, non vediamo alcun motivo intrinseco per cui non sia possibile utilizzare LDAP in una forma che si avvicina a un sistema basato su transazioni, ovvero ogni 5 - 10 accessi comportano una lettura seguito dal ciclo di scrittura (una modifica nel gergo LDAP).
- Sospettiamo che la vera risposta a questa domanda sia (con le scuse alla memoria del compianto Douglas Noel Adams): il rapporto tra letture e scritture è 42!

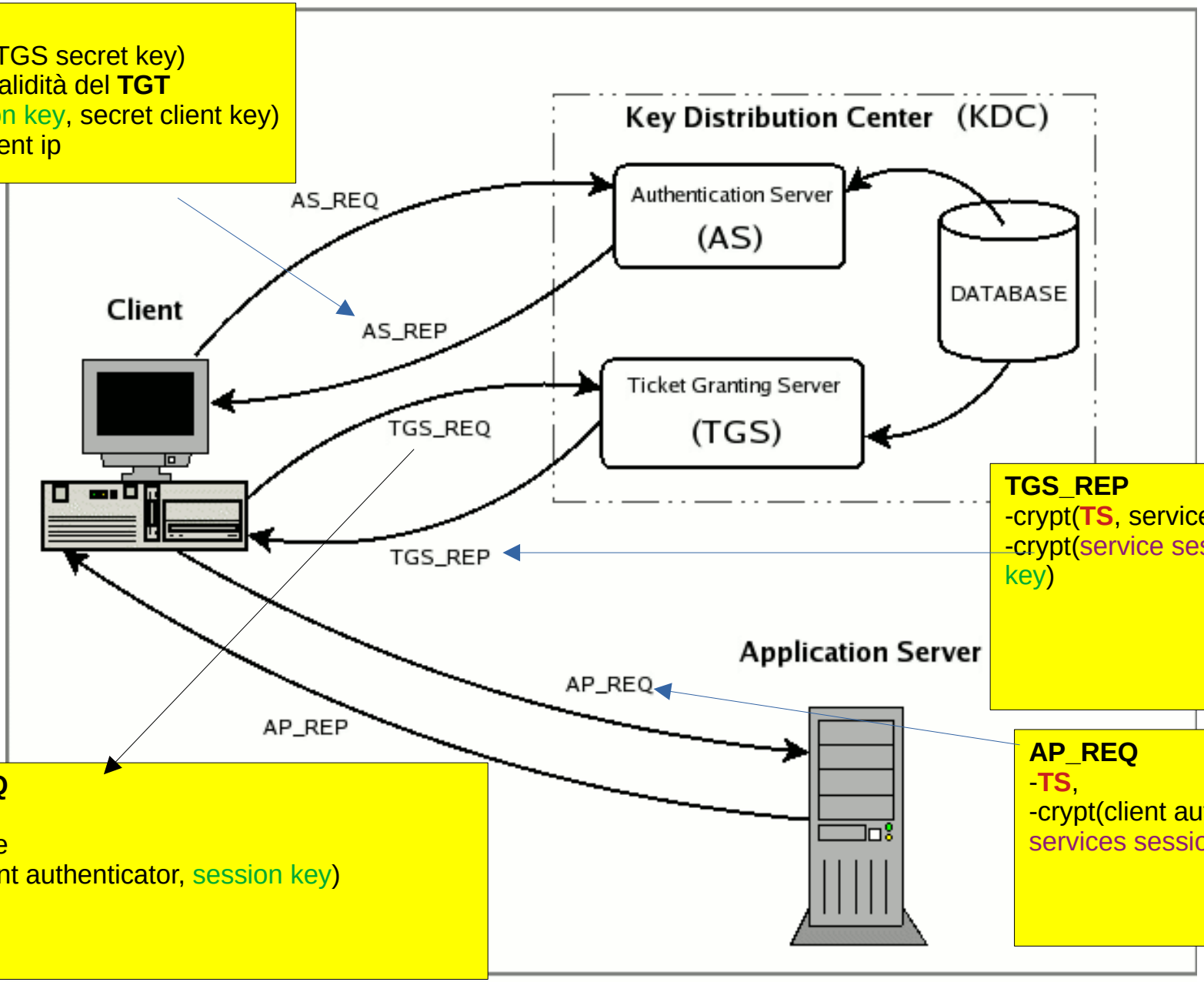
- Un dbms di solito ha più problemi di sicurezza...

Kerberos

- **Kerberos** è **un sistema per l'autenticazione distribuita basato sulla crittografia**.
- Il protocollo si affida ad un insieme limitato di nodi (**KDC**) fidati che fanno da arbitro alla situazione richiedendo e fornendo prove dell'avvenuta identificazione ed autenticazione delle parti.
- I sistemi che usano una stessa infrastruttura basata su **Kerberos** formano un **Realm**, che viene identificato tramite un nome ben definito, ad esempio: **`WIN.DSI.UNIVE.IT.`**
- Vediamo come funziona il protocollo **Kerberos**. Notate che si hanno principalmente 3 attori: un **client**, un **application server** e un **key distribution center** (**KDC**). Il **KDC** può essere integrato nel **application server**.
  - **KDC = (Autentication Server, Ticket Granting Server,DB)**

Kerberos - Funzionamento

AS_REP:
-crypt(**TGT**, TGS secret key)
-periodo di validità del **TGT**
-crypt(session key, secret client key)
-Client id, client ip



TGS_REP
-crypt(**TS**, service secret key)
-crypt(service session key, session key)

TGS_REQ
-**TGT**
-Id Service
-crypt(client authenticator, session key)

AP_REQ
-**TS**,
-crypt(client authenticator, services session key)

NB: Un application server non comunica mai direttamente con il Key Distribution Center.

Kerberos - Funzionamento

- **AS_REQ** è la richiesta iniziale di autenticazione (o di servizio) del client, non è crittata. Tale messaggio è diretto alla componente del **KDC (Key Distribution Center)** nota come **Authentication Server** (**AS**);
- **AS_REP** è la risposta dell'**Authentication Server** alla richiesta precedente. Sostanzialmente contiene:
  - il **TGT** (**Ticket-Granting Ticket**, criptato con la **chiave segreta del TGS**),
  - il **periodo di validità del TGT**,
  - la **chiave di sessione** (criptata con la **chiave segreta del client** richiedente),
  - l'identificativo del client e il suo indirizzo di rete.
- **TGS_REQ** è la richiesta da parte del client rivolta al **Ticket Granting Server** (**TGS**) per un **ticket di servizio(TS)**. Dentro questo pacchetto viaggia:
  - il **TGT** ottenuto dal messaggio precedente,
  - l'identificativo del servizio richiesto,
  - un **autenticatore** generato dal client e criptato con la **chiave di sessione**;

Fonte http://www.zeroshell.net/kerberos/Kerberos-operation/

Kerberos - Funzionamento

- **TGS_REP** è la risposta del **Ticket Granting Server** alla richiesta precedente. Essa contiene:
  - il **ticket di servizio** (**TS**) richiesto, criptato con la **chiave segreta del servizio**,
  - una **chiave di sessione di servizio** generata dal **TGS** e criptata con la precedente **chiave di sessione** generata dall'**AS**;
- **AP_REQ** è la richiesta che il client manda ad un **Applicaton Server** per accedere ad un **servizio**. Le componenti sono:
  - il **ticket di servizio (TS)** ottenuto dal **TGS** con la risposta precedente,
  - un **autenticatore,** generato sempre dal client, ma questa volta criptato con la **chiave di sessione del servizio** (generata dal **TGS**);
- **AP_REP** è la risposta che l'**Applicaton Server** invia al client per provare di essere veramente il server che il client si aspetta. Questo pacchetto non è sempre richiesto. Il client lo richiede al server solo quando è necessaria la mutua autenticazione.

Fonte http://www.zeroshell.net/kerberos/Kerberos-operation/

Kerberos – integrazione con LDAP

- Esistono diverse implementazioni di Kerberos. In ambiente UNIX le più note sono **MIT Kerberos** e **Heimdal**. Mentre il **protocollo Kerberos è standard**, e pertanto le implementazioni sono interoperabili, alcune funzionalità di amministrazione, che usano un protocollo ad hoc, possono non esserlo.
- Tramite il modulo **PAM** appropriato (pacchetto `libpam-krb5`) è possibile far autenticare un utente del sistema usando le sue credenziali **Kerberos**.
- Il modulo si occuperà anche di far avere un **TGT** all'utente, che così potrà usare altri servizi che partecipano al **REALM** senza dover fornire nuovamente la password (**Single Sign-On**).

Kerberos – integrazione con LDAP

- In Debian/Ubuntu per installare Kerberos:

```
# apt install krb5-kdc krb5-admin-
server libpam-krb5
```

- Sebbene anche i servizi possano usare **Kerberos** attraverso **PAM**, questo non permette la funzionalità di *single sign-on* appena descritta, né altre caratteristiche avanzate di **Kerberos**. Pertanto i servizi che lo richiedono spesso usano **Kerberos** per conto loro o attraverso ***GSSAPI*** (**Generic Security Service Application Program Interface**).

Active Directory - AD

- **Un Dominio è un insieme di Server, Workstation e Utenti che operano all'interno di uno stesso ambiente condividendo risorse gestite attraverso regole e permessi (policy).**
- Secondo la definizione di **Microsoft** un **Dominio** è **"un insieme di computer che condividono un database di risorse di rete e che vengono amministrati come un'unità con regole e procedure comuni".**

Active Directory - AD

- **Active Directory (AD)** è un **database ldap integrato nei server Windows** (>=2000), che fungono da **Domain Controller (o Controller di Dominio),** e consente di catalogare e gestire in modo centralizzato risorse di vario genere come: utenti, gruppi di lavoro, stampanti, cartelle condivise, ecc.
- **Active Directory** è quindi un **contenitore di oggetti**, ovvero utenti, gruppi, computer, server, stampanti, unità organizzative.
- Utilizza **Kerberos** per la gestione degli accessi.

Active directory

- **AD**, quindi, consente di gestire un **Dominio Windows**.
- La struttura del database **AD** è di tipo **gerarchico**, con contenitori che contengono oggetti e altri contenitori, sempre in una struttura ad albero.
- Per la realizzazione di un **Dominio Active Directory** è ~~sufficiente lanciare il comando~~ `dcpromo.exe` necessario installare il **ruolo** tramite **Server Manager**. Il sistema si occuperà da solo di installare tutto ciò che serve per l'autenticazione centralizzata, tra cui anche un server **DNS**.
- La gestione è demandata all'applicazione "**Active Directory Users and computers**".

!!! Warning !!!

- Importanti per un corretto funzionamento di **ldap**, **kerberos** e **active directory** sono:
  - Sincronizzazione degli orologi (**NTP**), perchè i ticket hanno una durata temporale.
  - Corretta configurazione dei **DNS** sia diretta che inversa, altrimenti i vari attori possono non riconoscersi tra loro.
  - Correttezza dei certificati **SSL**, se si parla di crittografia è essenziale che i certificati che garantiscono l'autenticità di client e server siano validi. (Vedremo **SSL** più avanti).
  - **Manutenzione costante** del database utenti/gruppi/client.

Kerberos + LDAP / Active Directory Eventuali Progetti

- **Progetto**: realizzare un sistema di autenticazione basato su kerberos+ldap utilizzando la vs distribuzione preferita.
  - Per ubuntu potete seguire le guide:
    - https://help.ubuntu.com/22.04/serverguide/openldap-server.html
    - https://help.ubuntu.com/22.04/serverguide/kerberos-ldap.html

      NB: NON USATE I NOMI kerberos.com e ssh.com in /etc/hosts!!!
- **Progetto**: realizzare un sistema di autenticazione basato su active directory utilizzando Windows 2012 server o Windows 2016 server.
- **Progetto**: realizzare un sistema di autenticazione utilizzando un database sql per la gestione degli utenti.
- Progetto: configurare sssd https://ubuntu.com/server/docs/service-sssd-ldap-krb

Avete un Sistema Autenticato!