Computer Engineering Degree *Computer Engineering* Year 2017/18	Practical classes **Lab1**	**CSDA** **Unit II** INTRANET

Integration of global services in enterprise environments I:

The INTRANET

Deployment of a secure information server I

*Active directory secure service (**Single sign-on**)*

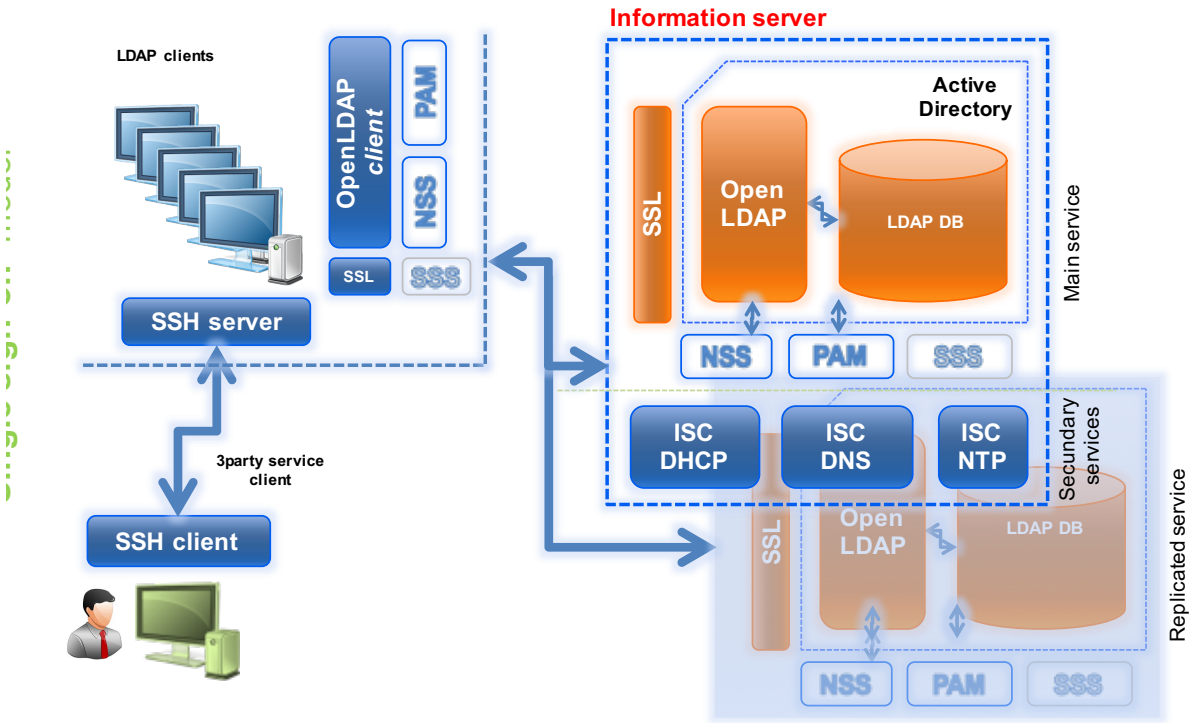


Table of contents

Table of contents 2
Main goals 3
Getting started: Creating the clone for lab1 4
Assignment 1: The Setting. Updating and initial configuration for `server-01` 5
Assignment 2: The Core. Installation and configuration of the secure information service: OpenLDAP (over SSL) 6
Assignment 3: LDAP Replication. Configuration of the openLDAP service in "Multi-Master" mode of replication (Syncrepl) 11
Assignment 4: All together. Integration of openLDAP and NSS/PAM to build an identification and authentication service 13
References and Resources 14

Main goals

- To learn about processes for adapting basic servers to certain needs. In this case:
  - Installation and configuration of **OpenLDAP** as a centralized *active directory* implementation for the **identification** and **authentication** (validation) of users, groups, hosts, domains, etc.
  - Integration of the LDAP directory service as a global information service for a specific computational environment, through PAM and NSS (SSS) software.
- Adaptation, integration and configuration of client services for the centralized information service.
- To become familiar with and handle different techniques and tools for administration and testing of said services.

Getting started: Creating the clone for lab1

1. Create a new clone from the initial base system "**core**".
    a. Select this option: "**Restart MAC address**"
    b. Type: **full**
    c. Select **all** the branches from the snapshot "tree".
2. Create an initial snapshot for that clone before starting the lab class.
    a. Remember to keep the VM off
    b. Call it **snapshot_P1**

Assignment 1: The Setting.

Updating and initial configuration for `server-01`

1. First, update the system from debian repositories.
2. Then, you will have to adapt your clone_P1 to turn it into a **secure information server**. So, carry out the required tasks as follows:
    a. Hostname: `server-01`.
    b. Local name resolution:
        1. Hostname: `server-01.localdomain`
        2. Alias: `server-01`
    c. Networking: "*bonding channel*" mode
        1. Make sure that both of the `clone_P1` network interfaces are connected to "type NAT" network *network_1.*
        2. Build and configure a "*bonding channel*" network type using both network interfaces (eth0 and eth1) and one of the available algorithms:
            - *IP*: (example)
                - (**bond0**): 192.168.0.**11**
            - *Network mask:* 255.255.255.0
            - *Network*: 192.168.0.0
            - *Broadcast*: 192.168.0.255
            - *Gateway*: 192.168.0.1
            - **High ability** algorithm: *`Active-Backup`*
    d. Disable all those services that you are not going to use. At your own discretion.

Note:

The default operation of **channel bonding** makes all the cards work with the same MAC (that of the primary slave) and interferes with the internal routing methods of VirtualBox between the host and the guest of the virtual machine. This forces you to configure the bonding kernel module so that the MAC address of the master interface is always the MAC address of the active slave card, each slave interface remaining with its own MAC address. This operation will cause slight delays, as the equipment has to update its ARP tables when the bonding channel fails. The parameter to add in `/etc/network/interfaces` to achieve this is as follows:

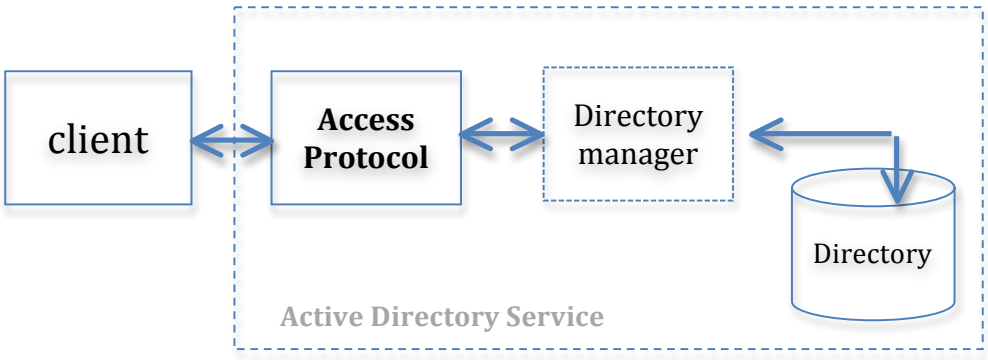
- `bond-fail-over-mac 1`

More details in [7] [8]

Assignment 2: The Core.

Installation and configuration of the secure information service: OpenLDAP (over SSL)

Now, the idea is to build a **secure** information server that enables us to manage the users, groups, hosts and local service data in a centralized and secure mode, as well as to manage the validation and authentication process. The core of that system will be the LDAP database where the user (*username*, *passwords*, GECOS, shell…) host and service data will be stored. We will call this system "LDAP Active Directory Service".



OpenLDAP (WITHOUT **TLS/SSL security**): Now, we will have to install the OpenLDAP software which enables us to build the "LDAP Active Directory Service":

1. *Installation*[1]:
   - Openldap (slapd)
   - Tools to manage the LDAP database
2. *Configuration*;
   - Password for the LDAP directory (manager) `cn=admin`: "ldap"
   - Database engine: HDB
   - LDAPv2 support
   - Domain name: "**localdomain**"
   - Root distinguished name (DN): **`dc=localdomain`**
3. *Post-configuration;* Once installed and pre-configured initially, configure it to verify the following features:
   - **Active Directory:**
     - Add, at least, a new "index" for the "`uidNumber`" field (attribute). That enables to us to speed up the searches. Index to add: `eq`
     - Through the LDAP "*access list*", make sure that "admin" and "owner" users of each user LDAP entry have enabled the write rights for these fields:

[1] Use the official debían software repositories

1. `userPassword`
2. `shadowLastChange`
3. `loginShell`
4. `gecos`.

- **LDAP Service:**
  - Initially, your LDAP service must run without secure **SSL** connections (`ldap`). This means that service will use only the 389 TCP port and `server-01` as hostname.
  - What does the `ldapi:///` instance mean and what is it for?

4. *Checking*;
   - Make sure that the LDAP service is running and "listening to" LDAP *client* requests on the right TCP port.
   - Check the *base* LDAP structure using the installed LDAP client `tools`. What can you see?
5. *Creating the LDAP active directory tree*;
   - Make a backup of the LDAP database in a *ldif* file using openLDAP server tools.
   - Once this is done, create that organizational tree for the LDAP database (`dc=localdomain`), using *.ldif* files:

dc=localdomain

cn=admin

```
dn: cn=admin,dc=localdomain
dn: ou=people,dc=localdomain
dn: ou=groups,dc=localdomain
dn: ou=machines,dc=localdomain
dn: ou=services,dc=localdomain
```

More details in [4] [5]

6. *Loading data;*
   - Now, you should populate the LDAP database, adding the corresponding data:
     - User groups: `group1 (2000), group2 (2001)`
     - User entries:
       a. `user1` (UID `2000`) (grupo `group1`) (password: `temporal1`)
       b. `user2` (UID `2001`) (grupo `group2`) (password: `temporal2`)
     - Host entries:
       a. `cliente` (IP `192.168.0.20`)
       b. `server-01` (IP `192.168.0.11`)

c. `server-02` (IP `192.168.0.12`)

d. `server-03` (IP `192.168.0.13`)

e. `server-04` (IP `192.168.0.14`)

`Note`: Use the **FQDN**: Full host name, including the domain name: `cliente.localdomain`

- Domain entry: `localdomain (*)`

- Check the new entries have been correctly added, using the LDAP *server* `tools`.

More details in [5]

B. **OpenLDAP** (WITH **SSL/TLS security**): Now it is time to add SSL/TLS security to the LDAP service to make encrypted communications:

1. *Installation:* Install the libraries and needed software to support **TLS,** the successor to SSL:
   - You should use the **GnuTLS** implementation to manage TLS certificates
2. *Generating TLS keys/certificates:* To carry out a process as similar as possible to real life, it will be necessary to create a CA[2] (Certificate Authority). This method is secure and easy to scale, but requires more work initially and more long-term maintenance. With that certificate, you will be able to sign and validate service TLS certificates, like the LDAP service certificate:
   - Generate the *CA certificate* (*self-signed*) to sign and validate the LDAP service certificate:
     - Generate a *CA private key*:
       a. File name: `CA_server-01.localdomain.key(2).`
       b. Key generated by default.
     - Generate the CA certificate and sign it yourself using the private key.

       → the certificate public key will be integrated into the CA certificate (automatically)
       a. *File name:* `CA_server-01.localdomain.cert (1).`
       b. *Sign mode:* **self-signed**
       c. Profile:
          i. *This certificate will be a CA certificate*
          ii. *This certificate will be used to sign other certificates*
       d. Other data[3]:

[2] A CA (**sign**) is a trusted entity that issues electronic certificates (docs) that verify a digital entity's identity on the Internet. In our case, we will act as a CA who will validate service certificates such as the LDAP certificate.

[3] You can use a template: `CA_server-01.localdomain.info`

- *Certificate type: X.509 (default)*
- *Expiration days*: 365 days
- *Country of the subject:* ES
- *State:* Cantabria
- *Locality: Santander*
- *Organization:* UC
- *Unit:* CSDA
- *"Common name"*: **server-01.localdomain** (3)
- *e-mail:* sistemas@localdomain

- Generate the *LDAP service certificate* that you will sign using the CA certificate (*private key*):
  - Generate a *LDAP certificate private key*:
    a. File name: `ldap_server-01.localdomain.key (2).`
    b. Key generated by default.
  - Generate the LDAP service certificate and sign it using the CA certificate and its *private key.* Save it as `ldap_server-01.localdomain.cert (1).`
    a. *Sign mode:* **signed by a CA**
    b. Profile*:*
       - This certificate will be used to encrypt data
       - This certificate will be used for a TLS server
    c. Other data[4]:
       - *Certificate type*: X.509 (default)
       - *Expiration days*: 365 days
       - *Country of the subject:* ES
       - *State:* Cantabria
       - *Locality: Santander*
       - *Organization:* UC
       - *Unit:* CSDA
       - *"Common name"*: **server-01.localdomain** (3)
       - *e-mail:* sistemas@localdomain

(1) PATH `/etc/ssl/certs`

(2) PATH `/etc/ssl/private`

→ Make sure that the ldap service (slapd) user is the owner (UNIX permissions) of the *LDAP certificate private key* file

(3) **It is very important** to use the FQDN and not its IP or another value.

More details in [5]**[6]**

[4] You can use a template: `ldap_server-01.localdomain.info`

3. *Re-configuration;* Enable secure LDAP connections using **TLS**:
    - **System level**:
        - LDAP service must be run under the **openldap** user permission, which will also belong to the *ssl-cert* group.
        - Modify the group property permission of the LDAP service certificate (`ldap_server-01.localdomain.key`) to the *ssl-cert* group.
            i. Check the access rights for that file.
    - **LDAP directory level**:
        - Add to LDAP directory configuration (OLC → `cn=config`) the required entries about the PATHs of CA certificate, LDAP service certificate and its private key (service).
    - **Service level**:
        - Modify the LDAP service (***slapd***) configuration to enable SSL/TLS security:
            i. Hostname: `server-01.localdomain`[5]
            ii. TCP port: 636
4. *Checking*:
    - **Service**:
        - Check that the LDAP service (slapd) is running and it is listening in the secure TLS port (636).

    b. **TLS certificate**:
        - Check that the LDAP service certificate (TLS) is “ok” through the same LDAP service.

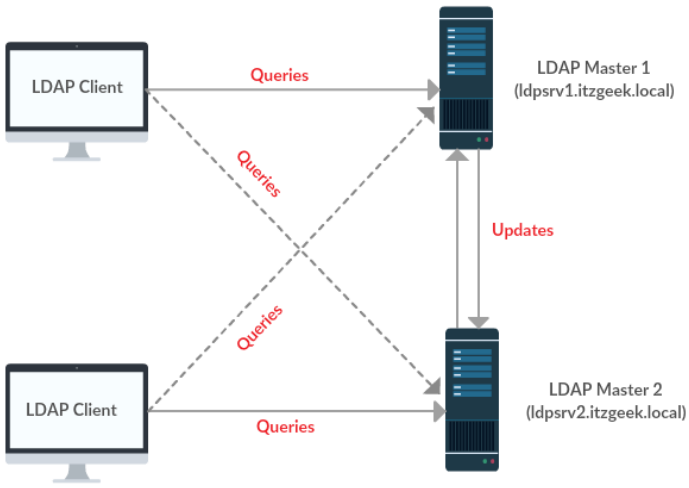
    c. **LDAP directory**:
        - Check that the LDAP service is operative (`ldaps`), using the OpenLDAP *client* tools.

[5] It is very important that you establish here the same name that we used in LDAP service certificate creation. It is the "Common name" field: FQDN

Assignment 3: LDAP Replication.

Configuration of the openLDAP service in "Multi-Master" mode of replication (Syncrepl)

In this module, we are going to configure the LDAP service to deploy an interesting feature; **Syncrepl replication.** We will consider a simple **N-Way Multi-Master** stage, where we will use 2 VMs running 2 `slapd` services, both in "master" mode. So we will get both of them can handle LDAP queries in reading and writing mode, in addition to replicate data one each other.



Configure OpenLDAP Multi-Master Replication

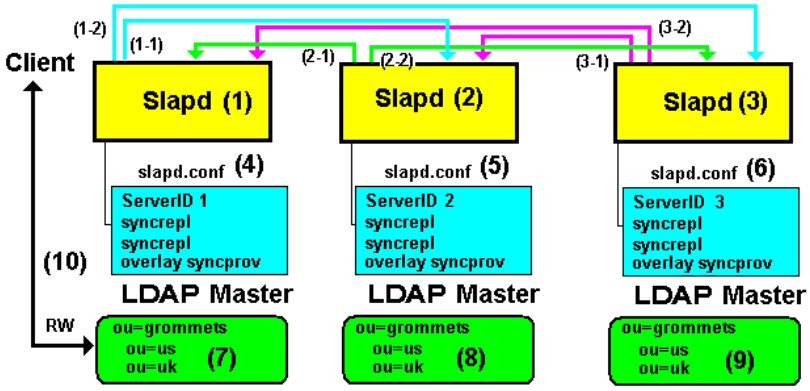
As in previous work, create a new snapshot before starting this assignment.

1. Remember to keep the VM off
2. Call it **snapshot_P1-A3**

A. *Servers schema:* To configure the LDAP service in replication mode, we need the service to be running on 2 different servers. So you have to clone the *clone_P1* VM. *(clone_P1-replica)*:
    1. Clone type: "link"
    2. *clone_P1-replica features*:
        a. Modify the hostname to `server-02`.
        b. *IP:* 192.168.0.**12**
        c. Local name resolution:
            1. Check the `/etc/hosts` file and add the hostname/IP of both VMs.
        d. Set the (*daemon*) LDAP service configuration to start up the `ldap`, `ldaps` and `ldapi` instance services.

B. *New configuration* in both LDAP servers: Add to the configuration of both LDAP services (LDAP configuration DIT), everything required to ensure that they are able to synchronize their DITs

(Configuration and DATA) in **N-way Multi-Master** mode. You should use only the `ldap` instance (unsecure) to sync both slapd services.



(Source: zitrax.com)

At least, try to replicate the DIT configuration (`cn=config`). If you achieve this, you just have to add a new config item, as an index, for checking.

C. *Checking*:

1. Using a *.ldif* file, create a new user in the LDAP directory on `server-01`. Once this has been done, check the result of a global search (ldapsearch) of all the LDAP entries, from both LDAP servers.

More details in [9] [10]

Once you have finished the assignment, revert to the previous snapshot **snapshot_P1-A3.**

Assignment 4: All together.

Integration of openLDAP and NSS/PAM to build an identification and authentication service

The goal now is to learn how to use the secure LDAP service from a client host. To be specific, you must make it possible for any LDAP user on `server-01` to open a **SSH** session on the `client`.

1. Make sure that the SSH server is installed and running on `client`.
2. This host should be able to use the LDAP service on `server-01` (over SSL) and thus **identify** and **authenticate** local and remote (LDAP) users.
    a. First, configure `client` as a LDAP client, so that it can access LDAP data using the *client* LDAP tools (`ldapserach, ldapadd, ldapmodify…`). Remember that the LDAP session should be secure, over **SSL** protocol.
        - LDAP server URI:
            1. `ldaps://server-01.localdomain`
            2. `ldaps://server-02.localdomain`
        - DN of the search base: `dc=localdomain`
        - LDAP version to be used: 3
        - LDAP account for root: `cn=admin,dc=localdomain`
        - LDAP root account Password: 'ldap'
        - Allow the LDAP administration account to behave like the local root
        - LDAP data base doesn't require login

    b. Secondly, add the LDAP service as another *identify* system on `client`. To do this, you need to correctly modify the **NSS** (Name service switch) configuration.
    c. Finally, configure `client` to use the LDAP service on `server-01` as an *authentication* system for only the SSH service. For this, you need to correctly modify the **PAM** (Pluggable authentication modules) configuration.
    d. Consider also the following aspects:
        - For LDAP users who connect to `client` for the first time, the system should create their HOME directories automatically.
3. Make sure that client and server-01 are property linked to each other through LDAP.
    a. **Identification** checking:
        1. `$ ldapsearch -x`
        2. `$ id <username>`
        3. `$ getent passwd`
    b. **Authentication** checking and `client` access:
        1. SSH access to `client` for "no root" users.

References and Resources

1. man
2. Google
3. **Slides:**

   → https://gitlab.com/herreroja/G679
4. More:

   OpenSSL
   [1] http://www.openssl.org/docs/

   GnuTLS
   [2] http://www.gnutls.org/manual/gnutls.html
   [3] https://help.ubuntu.com/community/GnuTLS

   OpenLDAP
   [4] https://wiki.debian.org/LDAP/OpenLDAPSetup
   [5] http://www.zytrax.com/books/ldap/ch5/index.html#step1-dit
   [6] https://help.ubuntu.com/community/GnuTLS

   Bonding Channel
   [7] https://wiki.debian.org/Bonding
   [8] https://www.kernel.org/doc/Documentation/networking/bonding.txt

   OpenLDAP Replication
   [9] http://www.zytrax.com/books/ldap/ch7/
   [10] http://www.linuxlasse.net/linux/howtos/OpenLDAP_N-Way_MultiMaster_Replication
5. More docs and web links are available in the GIT repository.