Computer Engineering Degree *Computer Engineering* Year 2017/18	Practical classes **Lab3**	**CSDA** **Unit II** INTRANET

Integration of global services in enterprise environments I:

The INTRANET

Deployment of a secure network file system and shared resources server

*Computer systems for **network file systems** and **resource sharing** management*

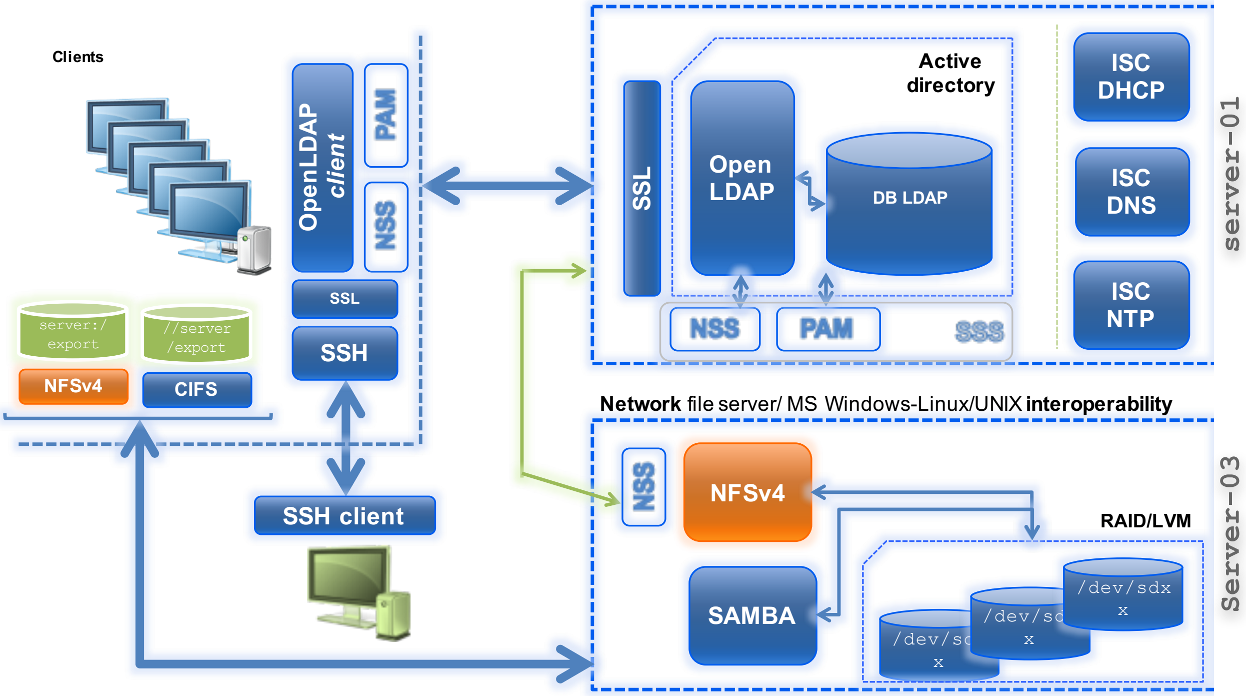


Table of contents

Main goals

- To learn about processes for "adapting” basic servers to certain needs. In this case:
  - Installation and configuration of **NFS** as a centralized *network file system* implementation for the **distribution** and **sharing** of user files among hosts.
  - Installation and configuration of **SAMBA** as a centralized *resource sharing* system for the **distribution** and **sharing** of computing resources (from user files to a printer and more) among heterogeneous systems (UNIX-Windows).
  - <u>Integration</u> of both systems with the LDAP service to avoid NFS and SAMBA access to resources and files by non-validated users.
- Adaptation, integration and configuration of client services for these services.
- To become familiar with and handle different techniques and tools for administration and testing of said services.

Getting started: Creating the clone for lab3

1. Create a new clone from the initial 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_P3**
3. For **client_LINUX** clone, create a new initial snapshot to complete this lab class.
    a. Remember to keep the VM off
    b. Call it **snapshot_P3**

Assignment 1: The Setting.

Updating and initial configuration for `server-03`

1. First, update the system from debian repositories.
2. Then, you will have to adapt your clone_P3 to turn it into a **secure network file server**. So, carry out the tasks required as follows:
    a. Hostname: `server-03`.
    b. Local name resolution:
        1. Hostname: `server-03.localdomain`
        2. Alias: `server-03`
    c. Networking:
        1. Make sure that both of the `clone_P3` network interfaces are connected to "type NAT" network *network_1.*
        2. Required data:
            - *IP*: (example)
                - (**eth0**): 192.168.0.**13**
            - *Network mask:* 255.255.255.0
            - *Network*: 192.168.0.0
            - *Broadcast*: 192.168.0.255
            - *Gateway*: 192.168.0.1
    d. DNS servers:
        1. *DNS1*: 8.8.8.8
        2. *DNS2*: 8.8.4.4
        3. *Search domain*: *localdomain*
    e. Disable all those services that you are not going to use. At your own discretion.
    f. Upgrade the server to last available software versions.

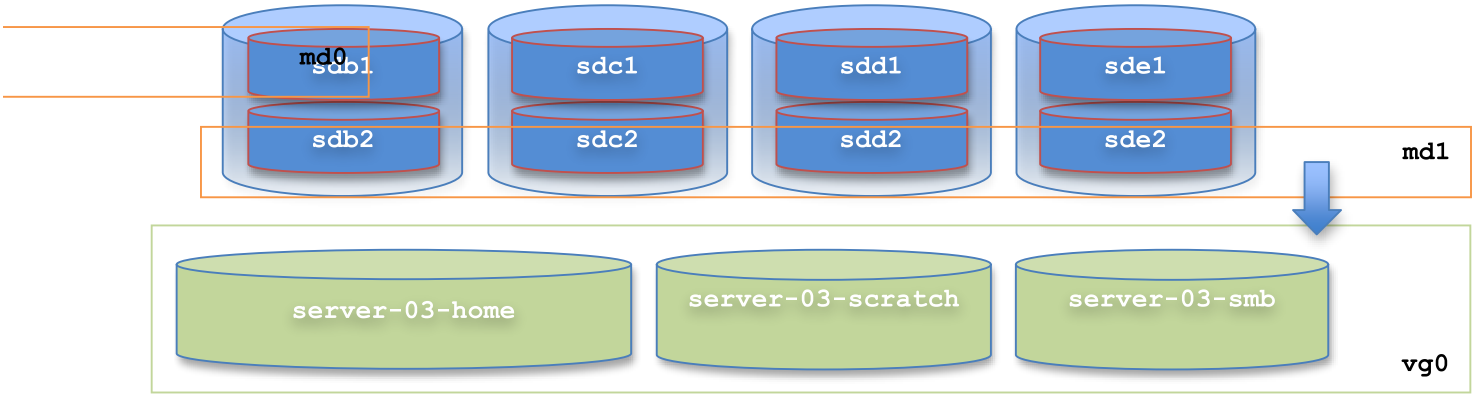
3. You have to configure `server-03` as client of service supplied by `server-01` (lab 1):
    a. **NTP Client.** Time (date) of our file server should be automatically synchronized by the NTP `server-01`. Use the `ntpdate-debian` app in a "client-server" model and decide the sync interval.
    b. **DNS client**. Add `server-01` as secondary DNS for your server.
    c. **LDAP client**. Our new server will be able to use the LDAP directory in a safe way (ssl) to **identify**[1] users who are managed by LDAP on `server-01/server-02`.

4. Build a **secondary storage system** for `server-03` using the following design:

/deb/sdb **/deb/sdc** **/deb/sdd** **/deb/sde**

[1] It is not necessary for users to be able to connect, in this case, to `server-03` by SSH (PAM)

RAID 5



   a. 4 disks (`/dev/sd{b,c,d,e}`)
   b. 2 partitions per disk, with equal size.
   c. 2 RAID devices configured in *RAID5* mode
      1. `md0` → `/dev/sd{b1,c1,d1,e1}`
      2. `md1` → `/dev/sd{b2,c2,d2,e2}`
   d. 1 LVM group volume `vg0`, using as physical volumes the RAID devices.
   e. 3 logical volumes:
      1. `server-03-home` → 1 GB
      2. `server-03-scratch` → 1 GB
      3. `server-03-smb` → 512 MB

5. Make the following file systems on each logical volume:
   a. `server-03-home` → **ext4**.
   b. `server-03-scratch` → **xfs**.
   c. `server-03-smb` → **ext4**.

6. Configure the system to mount these file systems permanently on the following directories:
   a. `server-03-home` → **/export/home**
   b. `server-03-scratch` → **/export/scratch**
   c. `server-03-smb` → **/export/samba**

   Mount options:
   - `noatime,nodiratime,`*`errors=remount-ro`*[2]
   - no dumps
   - no checks

7. Change the access rights of **/export/samba** and **/export/scratch** so that everybody can read and write on them.

[2] Only for ext4

Assignment 2: The Core (Linux).

Installation and configuration of the network file system: NFSv4

The idea is now to deploy a *secure* and *centralized* network file system that enables users to keep their files and data in a safe place and which can be accessed remotely.

NFSv4 (WITHOUT **security features**): We will configure `server-03` to export user files and directories (`/export/home`) and the directory "*scratch*" (`/export/scratch`)

1. Pre-Installation:
    a. Check the current kernel supports NFSv4.
    b. Copy the full content of user's homes created in Lab1 (**LDAP**) from `client:/home/` to `server-03:/export/home/`. Also, check the access and ownership permissions are ok (`0700`)
    c. Move `/home` to `/home.ini`
    d. Make a symbolic link between `/home` and `/export/home` (source)
2. *Installation*[3]:
    a. NFS server (nfs-kernel-server)
    b. NFS support files common to client and server (nfs-common)
3. *NFS configuration*:
    a. Enable IDMAPD
        1. IDMAPD domain: `localdomain.`
    b. Table of exported NFS file systems:
        1. All our exported FS will be located in `/export`
            a) Counfigure `/export` as the "**root**" of our table of exported NFS file systems.
        2. Exported file systems:
            a) `/export`
            b) `/export/home`
            c) `/export/scratch`
        3. Export options:
            a) **RW** mode
            b) Syncronization mode: **async**
4. *Checking* (on `server-03)`:
    a. Create a temporary directory: /`tmp/nfs`
    b. Using NFS, mount the NFS file system `home` on the temporary folder.
    c. Make sure everything is ok (NFS mounting)
    d. Unmount the NFS file system and remove the temporary folder.

[3] Use the official debian software repositories

NFSv4 (WITH **security features**): Now we have to re-configure `server-03` to export the NFS root directory (`/export`), user files and directories (`/export/home`) and the directory "*scratch*" (`/export/scratch`) safely, **under specific security controls**. To do this, perform the following tasks:

1. **Host level** security:
   a. Configure NFSV4 on `server-03` to export only to the LOCAL subnet (192.168.0.0). That is, only hosts within this network will be able to access (*mount to*) the file systems exported by `server-03`.
2. **User level** security:
   a. On client hosts, force NFS to distinguish "*root*" from NFS server "root", considering it as *nobody* user. This is very useful when exporting directories to "unreliable" client computers.
   b. Similarly, force all users of NFS client hosts with UIDs between 2001 and 3000 to be limited in their *anonymous* user access permissions.

3. **File level** security:
   a. Add ACL support for `/export/home/.` Build an access list (ACL) for the exported directory `/home/user2`
      i. For the owner, TOTAL Control
      ii. For the owner's group, accesses "Rxtcy"
      iii. For all other users, accesses "Rxtcy"
      iv. For "`user1`", **allow** any access

   Remember that NFSv4 ACL system is only available from nfs client side, through local directory.

   More details in [4] [5]

4. **Size control (**growing**)** for the NFS file systems, using a *users/groups* **quota** system.
   a. Configure your server so that you can make use of a quota system in the NFS exported file system `/export/home`:
      1. A limit of 100 MB will be established for each user (*soft limit*).
      2. If the user exceeds this limit, it will have 5 days to remove content until the account is blocked.
      3. Under no circumstances will any user be able to exceed the 120 MB of space in his `$HOME` (*hard limit*).

Assignment 3: NFSv4 client.

Configuration and checking of `client` VM as a NFSv4 client

Now it is time to configure the `client` VM as a NFSv4 client:

1. Pre-Installation: As with `server-03`, you have to configure `client` as client of service supplied by `server-01` (lab 1):
   a. **NTP Client.** Time (date) of our file server should be automatically synchronized by the NTP `server-01`. Use the `ntpdate-debian` app in a "client-server" model and decide the sync interval.
   b. **DNS client**. Add `server-01` as secondary DNS for your server.
   c. **LDAP client**. Our new server will be able to use the LDAP directory in a safe way (ssl) to **identify and authenticate** users who are managed by LDAP on `server-01/server-02`.
      1. Configure `client` as LDAP client host so that we can access the LDAP directory using the LDAP Client Command utility (`ldapserach, ldapadd, ldapmodify` …). Remember that the connection must be secure, over SSL and that we have a replicated LDAP system.
      2. Add LDAP services on `server-01/server-02` as user/host identification method.
      3. Reconfigure the client system to enable LDAP authentication for the SSH service.
2. *Installation*: First, we must install and pre-configure the NFS client "side"
   a. nfs-common
3. *NFSv4 configuration* on `client`:
   a. *NFS static configuration***.** Configure the `client` VM to enable the initial **static** mount (booting time) of `/export/scratch (server-03)` on `/remote/scratch (client)`. Ensure this is done permanently. Consider the *default* mount options.
   b. *NFS dynamic (automount) configuration. Now,* configure the `client` VM to mount **dynamically** (on demand) in the local directory `/remote/home/`, the user HOME directories copied to the remote directory `/export/home/`, located on `server-03`:
      1. On the `client`, `/home` should be a symbolic link to `/remote/home`.
      2. Mount options:
         a) `acl`
4. *Operation checking:*
   a. The `/remote/scratch` directory must be mounted on `client`.

b. One LDAP user (user2) will be able to open an SSH session on `client`, and also have its $`HOME` directory (located on `server-03)` available. That directory will be mounted on demand and automatically.

c. On `client`, check the default "acls" for user2 $`HOME`.

  - o Check that the changes made to the previous module are maintained:
    → For "`user1`", **allow** TOTAL access to user2 $`HOME`.
    → For "`user2`", **deny** TOTAL access to user1 $`HOME`.

More details in [4] [5]

d. Verify that the quota system operates properly.

  - o Login as user2 and create a file image of size 1024 MB. Use the `dd` command.

Assignment 4: The Core (Windows).

Installation and configuration of the muilti-platform UNIX-Windows system: SAMBA

In addition to NFSv4, we should install and set up an SMB/CIFS service on `server-03`, using SAMBA software. Also, we will need LDAP validation for the SAMBA service, in order to use the SAMBA resources available using credentials. So the pre-requisite is an OpenLDAP server configured with a directory that can accept authentication requests.

More details in [8] [9] **[10]** [11]

1. *Pre-installation*: Configure the LDAP service on `server-01` to integrate SAMBA and enable LDAP validation with the SAMBA service:

   a. In order for OpenLDAP to be used as a backend for SAMBA, logically, the DIT will need to use attributes that can properly describe Samba data. Such attributes can be obtained by introducing a Samba LDAP schema.
      i. Install the samba package that contains the LDAP schema, necessary to **integrate** with SAMBA.
         1. SAMBA documentation.
      ii. Now, import that schema (*.ldif* format) into the config DIT of LDAP directory:
         1. `/usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz`

   b. Set up new LDAP "*access lists*" so the "admin" and "owner" users only have write permissions on the following fields:
      i. `userPassword`
      ii. `sambaLMPassword`
      iii. `sambaNTPassword`

   c. Also, we can set up new indexes based on these SAMBA attributes to improve performance when a client performs a filtered search on the DIT:
      i. `sambaSID eq`
      ii. `sambaPrimaryGroupSID eq`
      iii. `sambaGroupType eq`
      iv. `sambaSIDList eq`
      v. `sambaDomainName eq`

2. *Installation*: Install the SAMBA service on `server-03`. Required packages:
   a. SAMBA
   b. SAMBA and SMB tools
   c. SAMBA documentation
   d. SMB-LDAP tools (`samba-common-bin`).

3. *Configuration*: Now it's time to configure SAMBA on `server-03` as a **resource sharing service,** in order to export the local directory “samba” and the CD/DVD device to the LOCAL network (192.168.0.0). Remember to use `server-01` as validation backend in SAMBA:

a. Work group: “**LOCALDOMAIN”**

b. Security level: **“user”** (ROLE_STANDALONE)

c. Net bios: **SAMBA CSDA**

d. *OpenLDAP (over SSL) backend:*

i. backend**: ldapsam:ldap://server-01.localdomain:389”**

1. **Disable SSL.** It is not supported by this SAMBA version

ii. ldap suffix: **dc=localdomain**

iii. ldap admin dn: **cn=admin,dc=localdomain**

iv. ldap suffix: **dc=localdomain**

v. ldap group suffix: **ou=groups**

vi. ldap user suffix = **ou=people**

vii. ldap machine suffix: **ou=machines**

e. Shared directory: **`/export/samba`**

i. SMB name: “**samba**”

ii. Sharing option: “**public**”

iii. Do not allow clients to modify the contents of shared directory

iv. The directory will be **browseable**

f. Shared device: CD/DVD

i. SMB name: “**cdrom**”

ii. Sharing option: “**public**”

iii. Do not allow clients to modify the contents of shared directory

iv. The directory will be **browseable**

g. Shared printer:

i. SMB name: “**CSDA Printer**”

ii. Sharing option: “**public**”

iii. The directory will be **browseable**

iv. Spool directpry: **`/vat/spool/samba`**

h. Configure the samba daemon to become a “standalone” service.

i. Store the LDAP “cn=admin” password in a local secret file. Use `smbpasswd` command.

2. *Post-configuration*: SAMBA-LDAP **INTEGRATION**: Use the **smbldap-tools** to populate the LDAP directory. In particular, use it to add *functional structures* to LDAP to enable LDAP authentication in SAMBA. Thus, we can manage and take control over access to SAMBA resources.

a. Smbldap-tools configuration: Take a look at ANNEX 1.

i. Get **SID** using:

```
$ net getlocalsid
```

b. Now, use smbldap-tools to:

i. Adapt LDAP database for SAMBA support ("repopulate")

ii. Add a new LDAP-SAMBA user (`user4`) which we use to check the SAMBA resources access.

iii. After creating `user4`, don’t forget to make a *password* for that user.

3. *Start-up*:
    a. Check that the main file of SAMBA configuration is syntactically all right.
    b. Restart the SAMBA service.

4. *Checking*:
    a. Check the new SAMBA service (from `server-03`) according to the following commands:
        i. **Command 1:** (Anonymous) It shows (lists) the different resources that the SAMBA service has available for sharing on the intranet:

```
$ smbclient –L //server-03/
```

        ii. **Command 2:** It provides an interactive "shell" to browse and access SAMBA shared resources:

```
$ smbclient –U user4 //server-03/samba
```

Assignment 5: SAMBA client.

Configuration and checking of `client-linux` and `client-w7` VMs as SAMBA clients

Finally, we focus on the SAMBA client side, both on the linux client (`client`) and windows client (`client-w7`) that we should create and install using MS WINDOWS 7.

1. **Linux SAMBA Client.** The `client` VM should mount the shared directory `/export/samba` using the SMB/CIFS protocol on `server-03`:
   a. Install on `client` the packages required by SAMBA in order to deploy the SAMBA client side on `client` that allows us to carry out resource sharing between both UNIX/Linux and MS-Windows platforms.
   b. On `client`, list the SMB shared resources of `server-03`.
   c. Connect to `server-03` from `client` using SMB and *user4* credentials. Browse (navegate) the `samba` directory.
   d. Try to make a new directory in the `samba` directory.
   e. Connect using the **CIFS protocol**:
      - Create a new directory called `/remote/samba` on `client`.
      - Install the packages needed to mount a SMF file system on `client`
        1. `cifs-utils`
      - **Mount manually** the shared directory `/export/samba` (`mount.cifs`) on`/remote/samba` (`client`) as *user4*:
        1. Mount options:
           a. File system type: `cifs`
        2. Make sure the mounting is *permanent*.

2. [OPTIONAL] **MS Windows SAMBA Client.** First, build a new VM on VirtualBox. Then, install MS window 7 on it:
   a. Create a clone VM from **client** and call it *cliente-w7*
   b. Make an initial snapshot for using in the practical class. Call it **snapshot_P2**.
      - Remember to power off the VM beforehand.
   c. Install MS Windows 7 using an installation CD/ISO.
   d. Configure the new virtual host using these parameters:
      - Host name: **client-w7**
      - Domain: **LOCALDOMAIN**
   e. Connect to `server-03` from `cliente-w7` using the *network environment* of Windows:

- When you open the *network environment,* the system will detect the SAMBA server (`server-03)` automatically.
- Connect to it using *user4* credentials.
- Browse (navigate) the shared resources of the SAMBA service …

References and resources

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

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

   NFS

   [1] http://www.nfsv4.org/

   [2] http://ditec.um.es/laso/docs/tut-tcpip/3376c410.html

   [3] https://help.ubuntu.com/community/NFSv4Howto

   [4] https://linux.die.net/man/1/nfs4_setfacl

   [5] http://wiki.linux-nfs.org/wiki/index.php/ACLs

   SAMBA

   [6] http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/

   [7] http://www.samba.org/samba/docs/using_samba/toc.html

   [8] https://wiki.debian.org/LDAP/OpenLDAPSetup

   [9] http://wiki.samba.org/index.php/Samba_AD_DC_HOWTO

   [10] http://siddou.tk/2013/06/install-sambaopenldap-on-debian-7-wheezy/

   [11] https://help.ubuntu.com/lts/serverguide/samba-ldap.html

   Automount

   [12] http://web.mit.edu/Kerberos/krb5-1.7/#documentation

5. More docs and web links are available in the GIT repository.

Annex 1

/etc/smbldap-tools/smbldap.conf

```
SID="S-1-5-21-2985063129-2976061446-3412244960" # (**)
sambaDomain="LOCALDOMAIN"
masterLDAP="server-01.localdomain"
masterPort="636"
# slaveLDAP="ldap_srv_name"
# slavePort="636"
ldapSSL="1"
cafile="/etc/ssl/certs/CA_server-01.localdomain.cert"
verify="require"
suffix="dc=localdomain"
usersdn="ou=people,${suffix}"
computersdn="ou=machines,${suffix}"
groupsdn="ou=groups,${suffix}"
#idmapdn="ou=Idmap,${suffix}"
scope="sub"
hash_encrypt="SSHA"
crypt_salt_format="%s"
userLoginShell="/bin/bash"
userHome="/homes/%U"
userHomeDirectoryMode="700"
userGecos="User"
defaultUserGid="513"
defaultComputerGid="515"
skeletonDir="/etc/skel"
defaultMaxPasswordAge="3650"
with_smbpasswd="0"
smbpasswd="/usr/bin/smbpasswd"
with_slappasswd="0"
slappasswd="/usr/sbin/slappasswd"
```

(**) This SID will be the SAMBA ID that was obtained by running the command "`net getlocalsid`". Each one of you should own one SID.

/etc/smbldap-tools/smbldap_bind.conf

```
masterDN="cn=admin,dc=localdomain"
masterPw="ldap"
```