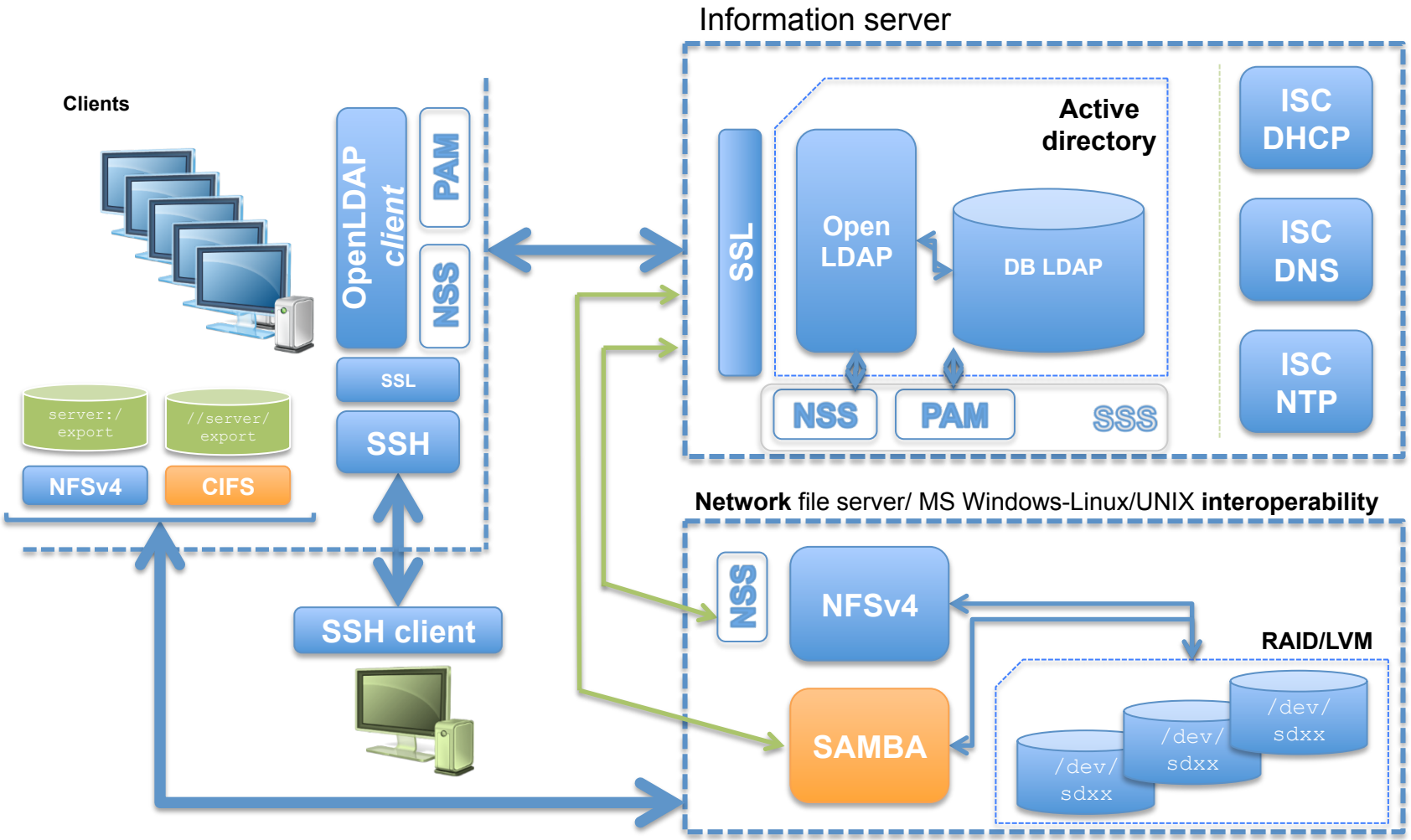
Computer System Design and Administration

Topic 8. Multi-platform interoperability and resource sharing service: SAMBA

José Ángel Herrero Velasco

Department of Computer and
Electrical Engineering

Secure information service: Puzzle

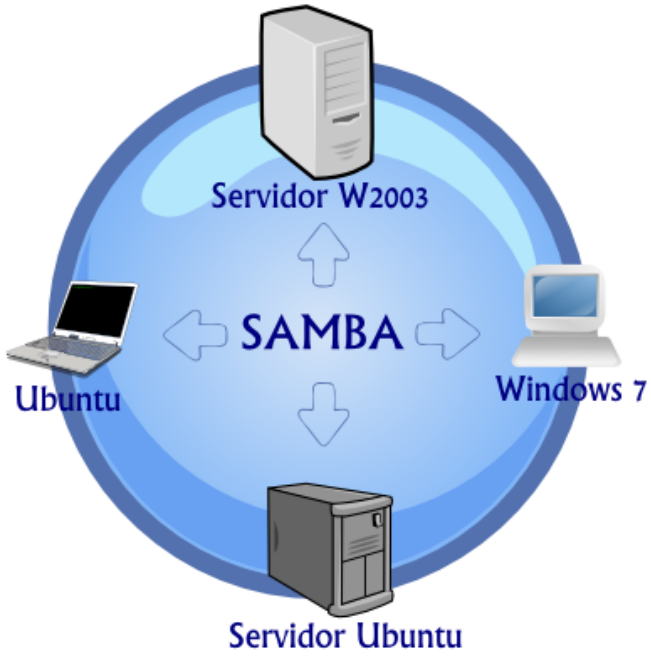


Target: …breaking down the boundaries of local data

- Deployment and development of a *secure* **network file and resource system**, based on **NFS** & **SMB/CIFS** (**SAMBA**) technology:
  - **Network file system:**
    - It is about **centralizing** the storage of (mainly) user files from a computer environment.
  - **Computational resources sharing system:**
    - [MS Windows] – [Linux/UNIX] interoperability:
      - **Heterogeneous network.**
    - *Introduction* to this important piece in the implementation of **LINUX *DOMAIN* controllers** for the integration of heterogeneous computing environments:
      - **WINDOWS – LINUX/Unix (transparency)**.

Multiplatform Interoperability Service

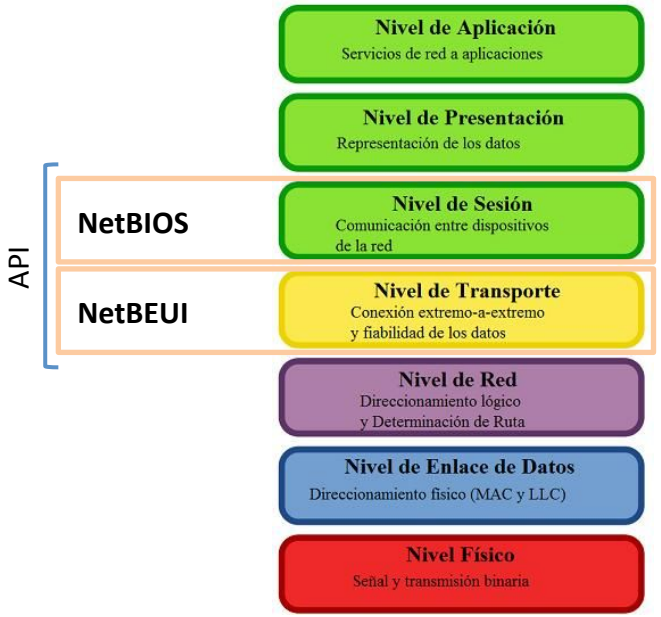
- **Deploys** a **Network Domain** *(WORKGROUP):*
  - Logical **grouping** of *heterogeneous* computers.
- **Shares** the available resources from Linux/Windows hosts over the network:
  - Local **file systems** (folders...).
  - **Printers.**
  - **Other devices** → CDroms, recorders...
- Enables users **to browse** **in the network** *(Intranet)*...:
  - ...Surfing across the **shared resources** in the network domain.
- **Validates** users (network) in the network **DOMAIN:**
  - For any computer to login into this network (domain):
    - **Windows** or **Linux** hosts, under the same credentials.
    - Unique "user space" in the *Intranet* (DOMAIN).
  - Requirements:
    - **Option A)** → SAMBA authentication:
      - **Samba backend:**
        - » Based on a *encrypted file* with usernames/passwords.
      - **Samba hostname resolution (winbind):**
        - » Similar to MS Windows "WINS" (`/etc/hosts` for Windows networks).
    - **Option B)** → Third-party service:
      - **An information service (*centralized*):**
        - » Using LDAP/Kerberos.
        - » It is able to deploy a Linux/UNIX **PDC/SDC** system:
          - **P**rimary/Secondary **C**ontroller **D**omain.
        - » Similarly to MS Windows "Active Directory".
      - **A local DNS service (bind9).**
      - **An NTP service.**



Windows **I**nternet **N**aming **S**ervice

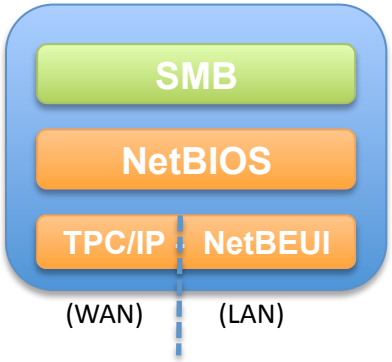
SAMBA: Underlying protocols

- **NetBIOS** (**N**etwork **B**asic **I**nput/**O**utput **S**ystem):
  - Designed by IBM in 1984.
  - It introduces a new **network interface** similarly to BIOS about hardware:
    - It allows **hosts on the same** network subnet to talk to one another using names instead of numeric address (IPs).
  - Basic features:
    - It provides session services (OSI layer 5).
    - It uses logic names for hosts (no IP address).
    - It supports secure data transfer:
      - **NetBIOS & SMB sessions.**
- **NetBEUI** (**NetB**IOS **E**xtended **U**ser **I**nterface):
  - Introduced by IBM in 1985:
    - Adopted by MS Windows (networking).
  - It is used by NetBIOS as **transport protocol** (OSI layer 4):
    - *LAN only*. On *WAN*, NetBIOS run over **TCP.**
  - Basic features:
    - Simple and efficient transport protocol:
      - **Up to 254 hosts.**
    - Limits:
      - **It uses the hostname as address and it doesn't support IP routing.**
    - The new MS windows versions support NetBIOS over TCP/IP.



SAMBA: Underlying protocols

- **SMB** (**S**erver **M**essage **B**lock): *"The core protocol":*
  - Designed by IBM (Barry Feigenbaum, IBM):
    - Developed (improved) later by **Microsoft** & **Intel (CIFS).**
  - It is a *resource sharing protocol* mainly:
    - It can run on top of a wide variety of *"underlying"* protocols.
    - → TCP (WAN) → NetBEUI (LAN) → IPX/SPX (NOVELL)
    - It usually runs over:
      - **TCP/IP directly (port 445).**
      - **NetBIOS +:**
        - » LAN networks:
          - **NetBEUI** API.
        - » WAN networks:
          - **UDP** → ports 137,138.
          - **TCP** → port 137,139.
        - » Novel and other networks:
          - Several legacy protocols such as NBF, IPX/SPX.
    - It is an "Inter-Process Communication" (IPC) system.
    - SMB serves as the basis for Microsoft's Distributed File System implementation:
      - **Introduced in MS-DOS to provide resource sharing.**
      - **MS Windows 7, 8 and 10 use SMB as their underlying protocol to implement the *"Windows Network Environment".***
  - Nowadays → **CIFS** (**C**ommon **I**nternet **F**ile **S**ystem):
    - Widely used by **Windows** systems (from 2000 to W10 → SMBv3…).



SAMBA: Underlying protocols

- **CIFS:** (**C**ommon **I**nternet **F**ile **S**ystem):
  - Based on SMB, evolved for WAN:
    - CIFS is a *particular implementation* of the SMB protocol, evolved by Microsoft.
  - **Linux UNIX:**
    - → SAMBA (SMBv3) uses CIFS.
  - **MS Windows:**
    - → Uses CIFS as its current *lingua franca* about file sharing.
  - It supports…:
    - File sharing.
    - "Network" printing.
    - User authentication and authorization.
    - Host name resolution.
    - Service announcement (Shared resources *browsing).*
  - Closed protocol:
    - MS license:
      - http://msdn.microsoft.com/en-us/library/ee442092.aspx.
      - **Open versions (Microsoft -- SAMBA):**
        - » http://www.samba.org/samba/ms_license.html.

SAMBA: Definition

- **SaMBa:** (huge) *suite* of **services** and **apps** for Linux/UNIX platforms:
  - **First target:** to make INTEROPERABILITY with MS Windows possible.
  - More than 2 million lines of code:
    - Developed originally by Andrew Tridgell.
    - As **"Open source"** → GPL license (**GNU**).
  - Protocols and services implementation:
    - Initially, it deploys the Microsoft protocol **SMB** (**S**erver **M**essage **B**lock):
      - **Nowadays → CIFS (Common Internet File System) for MS windows environments:**
        - » SAMBA implements the *"server side"* SMB/CIFS for UNIX/Linux platforms.
    - It "works" over MS-Windows **NETBIOS/NETBUIE API** (Windows networks):
      - **Which is used by Microsoft Windows and OS/2 (networking).**
  - **Scope:**
    - It enables users to use both Linux/UNIX & MS-Windows platforms in a **transparent** way (INTEROPERABILITY):
      - **Clients Windows (*hosts*) on (servers) UNIX/Linux networks.**
      - **Clients Linux/UNIX (*hosts*) on (servers) Windows networks.**
    - **Active Directory Authentication** Server – **PDC** (LDAP, kerberos, etc)
      - **User authentication is centralized by Linux server on Linux/Windows networks**
    - **Winbind**
      - UID and group *mapping*, for getting integration with LDAP or MS Windows AD
    - **Sharing** network resources:
      - **File Server**
      - **Print server**
      - **More devices.**

…but SMB is a *closed* protocol, how is it possible???

SAMBA: SAMBAv3 features

- **SAMBA** version **3.0** (**still in use**):
  - "Samba is an Open Source/Free Software **suite** that provides *seamless* file and print services to SMB/CIFS clients":
    - Source: samba.org.
    - Very popular: well supported and *active development.*
  - Originally created by Andrew Tridgell (1992):
    - Reverse-engineering.
  - It provides a stable, industrial-strength mechanism for integrating both Windows and Linux machines into UNIX network:
    - You only need to install one package on server.
    - No special software is needed on client side (Windows).
  - Features:
    - Includes *basic* functions of MS Windows **Active Directory** controller (Clients):
      - **Windows domain logins:**
        - » **LDAP** & **Kerberos 5.**
      - **Roaming windows user profiles.**
      - **CIFS print spooling.**
      - **Users login for deprecated MS windows (95/98/Me/XP...) and "currents" (7/8/10...).**
    - Unicode support:
      → **International languages.**
    - Main roles:
      - **File server.**
      - **Print server (print spooling).**
      - **Primary Domain Controller (PDC) → UNIX/Linux networks ONLY!!!**
      - **WINS server.**

    → Because many Windows protocols are closed and they are not fully documented, SAMBA developers have had to use *reverse-engineering* techniques:
      - **For this reason, Samba v3.0 can't deploy a FULL PDC on MS Windows networks yet.**

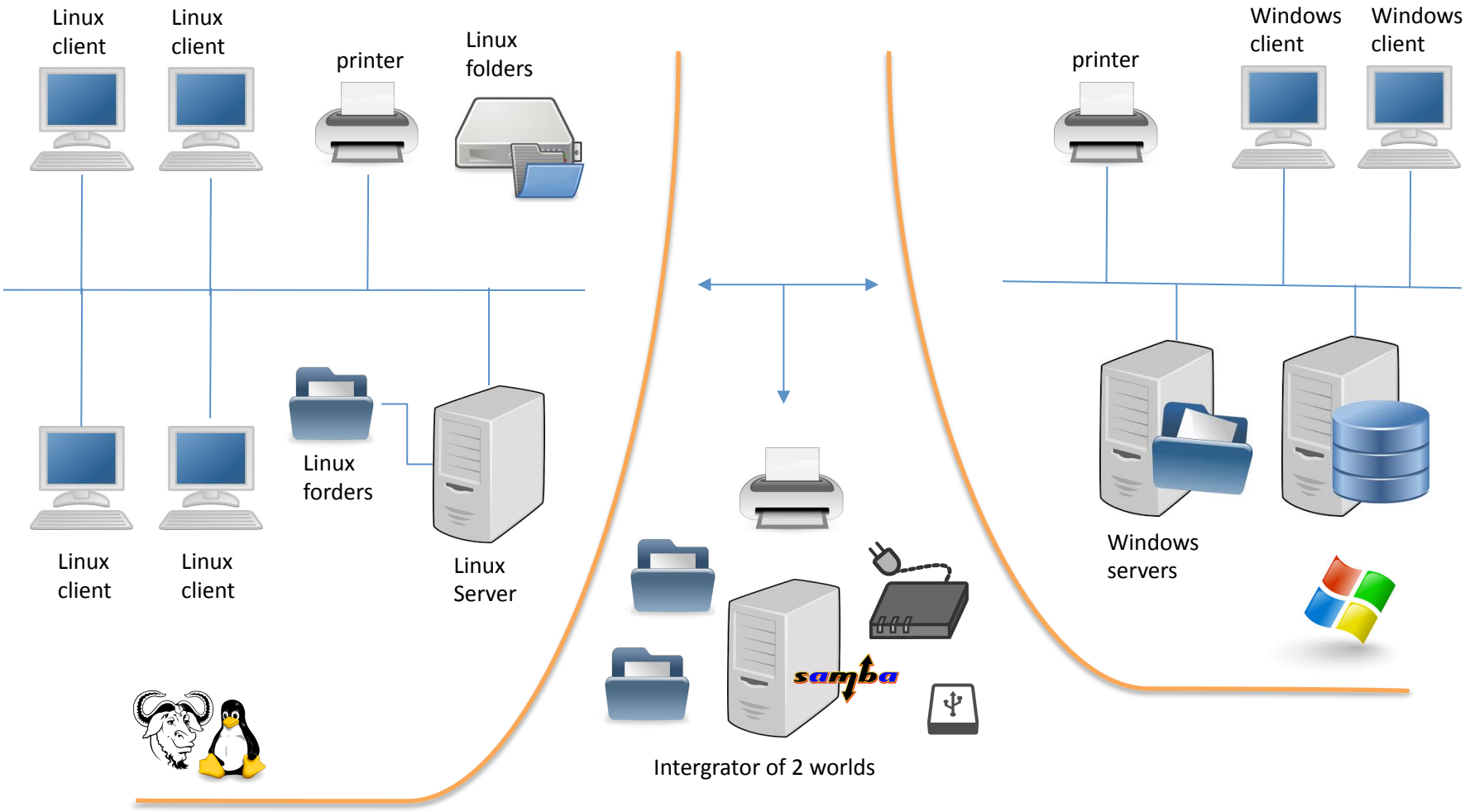
SAMBA: SAMBAv4 features

- **SAMBA** version **4.0** (**current**):

Current release (stable): **4.2.14**

  - It is a ***full replacement*** and upgrade of Samba 3:
    - 10 years' work (from 2003).
  - **Main target:**
    - To become an **Active Directory Controller** of Linux/Unix platforms for **UNIX/Windows networks:**
      - **Samba *AD CD.***
      - **Enable integrating MS windows machines to the Linux DOMAIN.**
  - Main features (regarding v3):
    - It supports:
      - **Active Directory authentication/authorization:**
        - » Users from Linux/Windows machines.
      - **Active Directory management protocols.**
    - Integration of:
      - **LDAP directory server.**
      - **Heimdal Kerberos authentication server.**
      - **Secure Dynamic DNS server.**
      - **Implementations of all necessary remote procedure calls for Active Directory.**

SAMBA: Architecture



SAMBA: Main daemons

- **smbd:**
  - Manages:
    - The **shared resources** between clients and SAMBA server.
    - All **notifications** between clients and SMB service.
  - Provides:
    - *File* service.
    - *Print* service (printer spooling).
    - Resources *search* service.
  - Carries out:
    - User **authentication** and **authorization.**
    - Shared resources **blocking.**
    - Data sharing through **SMB/CIFS** protocol.
- **nmbd:**
  - Manages:
    - The **local name resolution,** which provides a service similar to WINS (MS windows).
    - **Service announcement.**
  - Provides:
    - The **appropriate IP** address according to a client request.
    - A **list of resources** according to a client request.
  - It works together with **smbd**.

Both of them are run entirely as a **user proccess.**
(No *kernel-level* support needed).

SAMBA: Service installation

- **Previous:**
  - **Recommendations:**
    - Server must be physically safe.
    - Use SAMBA when the computational environment provides:
      - **Local DNS service**
      - **Local NTP service**
        - » Kerberos is very sensitive to time sync.
      - **LDAP service.**
    - Use **RAID** (5,6) mechanisms in your disk devices (store backend):
      - **Hardware/software.**
      - **+ Security and + Performance.**
    - Define the shared resources the server is going to share:
      - **File systems (shared directories):**
        - » Beware about shared directory rights.
      - **Devices:**
        - » Floppies, CDROM, DVD...
        - » **Printers.**

- **SAMBA service and tools** installation (as "root"):

server

```
$ apt-get update
$ apt-get install samba smbclient smbldap-tools
```

clients

```
$ apt-get install smbclient cifs-utils
```

SAMBA: Service configuration

- **Service** configuration (**server side**):

  `$ vi /etc/samba/smb.conf`

  (main flags):

  - It is divided into *sections* identified with brackets **[ ]**:
    - Every section is a **shared resource.**
    - There are 3 main sections:
      - **`global`, `homes` and `printers`.**
  - **`[global]` section:**

    → Parameters in this section are applied to the server as a whole.

    (more than 300 parameters).

    - `workgroup`: this controls what workgroup your server will appear to be in when queried by clients.

      → In our case: `LOCALDOMAIN`.
    - `netbios name`: NetBIOS name, name by which a Samba server is known (Windows networks).

      → In our case: `LOCALDOMAIN`.
    - `server string`: brief server description.
    - `encrypt passwords`: this boolean controls whether encrypted passwords will be negotiated with the client:
      - **The use of plain text passwords is NOT advised:**

      → In our case: `true/yes`.

SAMBA: Service configuration

- `passdb backend`: this option allows the administrator to choose which backend will be used for storing user and possibly group information:
  - **smbpasswd → The old plaintext passdb backend.**
  - **tdbsam → The TDB based password storage backend.**
  - **ldapsam → LDAP.**

  → In our case: …**LDAP.**
- `passwd program`: the name of a program that can be used to set UNIX user passwords.

  → In our case: = `/usr/sbin/smbldap-passwd -o %u`
- **`security`: this option affects how clients respond to Samba**
  - ***`user`* *(default)*: a client must first "log-on" with a valid username and password**
  - ***`domain`* : this mode will only work correctly if net(8) has been used to add this machine into a Windows NT Domain.**
  - ***`ads`*: Samba will act as a domain member in an Active Directory Domain (ADS) realm:**
    - » The machine running Samba will need to have **Kerberos** installed and configured and Samba will need to be joined to the ADS realm using the net utility.

  → In our case: = user.
- `allow hosts` or `hosts allow`: this parameter is a comma, space, or tab delimited set of hosts which are **permitted** to access a service.
- `deny hosts` or `hosts deny`: this parameter is a comma, space, or tab delimited set of hosts which are **denied** access to a service.
- `directory` or `path`: *PATH* to the shared resource.

SAMBA: Service configuration

→ LDAP integration:

- `ldap ssl`: it defines whether or not Samba should use **SSL** when connecting to the *ldap server* using *ads* methods:
  - **`Off`: NO SSL. "Plain" connection.**
  - **`start tls`: use the LDAPv3 StartTLS extended operation (RFC2830).**
- `ldap passwd sync`: this option is used to define whether or not Samba should sync the LDAP password with the NT and LM hashes for normal accounts (NOT for workstation, server or domain trusts) on a password change via SAMBA.
- `ldap admin dn`: it defines the **Distinguished Name** (DN) name used by Samba to contact the ldap server when retrieving user account information.
- `ldap sufix`: it specifies the *`distinguished name`* of the LDAP tree.
- `ldap user suffix`: this parameter specifies where **users** are added to the LDAP tree.
- `ldap group suffix`: this parameter specifies where **groups** are added to the LDAP tree.
- `ldap delete dn`: this parameter specifies whether a delete operation in the ldapsam deletes the complete entry or only the attributes specific to Samba.
- `ldap debug level`: controls the debug level of the LDAP library calls.

SAMBA: Service configuration

- **`[homes]` section:**
  - *If exists*, services connecting clients (users) to their **home directories** can be created on the fly by the server:
    - **These directories store the documents and user profiles (`homes`).**
- **`[printer]` section:**
  - This section works like [homes], but for printers.
  - *If exists,* users are able to connect to any printer specified in it.
- You can define other sections:
  - One section for each **shared resource.**
  - For instance:
    - **`[shared]` section: if it is a file system.**
    - **`[cdrom]` section: devices.**
- Relevant parameters:
  - `browseable` (yes): this controls whether this share is seen in the list of available shares in a net view and in the browse list.
  - `create mask/mode` (0770): when a file is created, the necessary permissions are calculated according to the mapping from DOS modes to UNIX permissions:
    - **In the same way, `directory mask` for directories.**
  - `guest ok` (yes): if this parameter is "yes" for a service, then no password is required to connect to the service.
  - `valid users`: list of users that should **be allowed** to login to this service.
  - `invalid users`: list of users that should **not be allowed** to login to this service.
  - `comment`: this is a text field that is seen next to a share when a client queries the server.

SAMBA: Server (daemons) configuration

- Parameters related to the definition of *shared printers:*
  - `print ok` or `printable`: if this parameter is yes, then clients may open, write to and submit spool files on the directory specified for the service.
  - `printer` or `printer name`: name of the shared resource.
- Parameters related to the definition of *shared directories:*
  - `preexec`: this option specifies a command to be run whenever the service is **connected.**
  - `postexec`: this option specifies a command to be run whenever the service is **disconnected.**

- **Server** (**daemons**) configuration (**server side**):

  `$ vi /etc/default/samba`

  This file *sets* the parameters associated with the execution of the service (**daemon**):
  - `RUN_MODE`: run mode for SAMBA service:
    - `daemons`.
    - `inetd`.

SAMBA: SAMBA-LDAP integration

- SAMBA can use **LDAP** *service* as **authentication *backend.***
- For this **it is necessary** to adapt the LDAP database:
  - A new schema for *managing* SAMBA information:
    - `samba.schema.`
  - Adjustment of LDAP directory (DB):
    - New index for new LDAP attributes (`eq`):
      - **sambaSID, sambaPrimaryGroupSID, sambaGroupType, sambaSIDList, sambaDomainName…**
    - New access lists for certain attributes:
      - **userPassword,sambaNTPassword,sambaLMPassword:**
        - "cn=admin,dc=localdomain" → write / Anonymous → auth / self → write.
    - New LDAP control entries about SAMBA:
      - **sambaDomainName=LOCALDOMAIN,dc=localdomain.**
      - **uid=root,ou=people,dc=localdomain.**
      - **uid=nobody,ou=people,dc=localdomain.**
      - **cn=Domain Admins,ou=groups,dc=localdomain.**
      - **cn=Domain Users,ou=groups,dc=localdomain.**
      - **…**
- **`smbldap-tools:`**
  - Suite of de *perl* scripts that enables you:
    - To adapt the LDAP directory to support SAMBA:
      - **Creating the new LDAP control entries:**
        - **`smbldap-populate.`**
    - To manage the LDAP directory from SAMBA service:
      - **User management:**
        - `smbldap-useradd, smbldap-userdel, smbldap-usermod, smbldap-userinfo…`
        - `smbldap-passwd.`
      - **Group management:**
        - `smbldap-groupradd, smbldap-groupdelm, smbldap-groupmod…`

SAMBA: SAMBA-LDAP integration

- **smbldap-tools suite configuration** (part 1):
  `$ vi /etc/smbldap-tools/smbldap.conf`
  (Main flags):
  - It is divided into *sections* identified with (#):
    - It defines the parameters that will determine the mode of behavior of the **smbldap** utility.
  - **General Configuration:**
    - `SID=`: secure Identifier Domain.
      → you can get the SID for your domain using: `net getlocalsid`.
    - `sambaDomain`: Samba Domain the Samba server is in charge.
  - **LDAP Configuration:**
    - `masterLDAP`: FQDN of the *master* LDAP server.
    - `masterPort`: TPC/IP port for the master LDAP server (default: 389).
    - `slaveLDAP`: FQDN of the *slave* LDAP server.
    - `slavePort`: TPC/IP port for the slave LDAP server (default: 389).
    - `ldapTLS`: should we use TLS connection to contact the ldap servers.
    - `cafile, clientcert and clientcert`: PATH to files which establish the SSL keys and SSL certificates of CA and LDAP service.
    - `verify`: how to verify the server's certificate (none, optional or require).
    - `suffix`: the distinguished name of the search base.
    - `usersdn`: branch in which users account can be found or must be added.
    - `computersdn`: branch in which computers account can be found or must be added.
    - `groupsdn`: branch in which groups account can be found or must be added.
    - …
  - **Unix account Configuration:**
    - `userLoginShell`: default shell given to SAMBA users.
    - `userHome`: default directory `$HOME` where user home directory is located.
    - …
  - **SAMBA Configuration:**
    - …
  - **Tools Configuration:**
    - `with_smbpasswd`: should we use the *smbpasswd* command to set the user's password (instead of the *mkntpwd* utility)?
    - `smbpasswd`: PATH to the *smbpasswd* binary (default: "`/usr/bin/smbpasswd`").
    - `with_slappasswd`: should we use the *slappasswd* command to set the Unix user's password (instead of the *Crypt:* libraries)?
    - `slappasswd`: PATH to the *slappasswd* binary (default: "`/usr/sbin/slappasswd`").

SAMBA: SAMBA-LDAP integration

- **smbldap-tools suite configuration** (part 2):
  `$ vi /etc/smbldap-tools/smbldap_bind.conf`
  (Main flags):
  - It sets the **credentials** for LDAP directory access:
    - `masterDN`: the distinguished name (`dn`) of root LDAP account used to bind to the master server:
      - `cn=admin, dc=…`
    - `masterPw`: the credentials (password) to bind to the master server.

(**) Beware of the UNIX permissions of this file because of it contains the *administration password* for the LDAP directory.

SAMBA: Client side configuration (Linux)

- From clients, we can use the following commands to *dialog* with SAMBA server:
  - Connection from **Linux/UNIX client** to SAMBA file server:
    - `$ smbclient //<SAMBA server> -U <username>`
      `smb> ls`
      `smb> get <file>`
      `smb> put <file>`
      `smb> cd <dir>`
      `smb> help`
  - **Shared resources list:**
    - `$ smbclient -L <SAMBA server>`
  - **Static mount** (Not permanent) on Linux/UNIX client machines:
    - Enable SMB support in Kernel:
      - **`$ modprobe smbfs/cifs`**
    - Mount SAMBA resource using SMBFS protocol (deprecated…):
      - **`$ mount -t smbfs -o username=<username>,passwd=<password>,workgroup=<workgroup name> //ip_servidor/resource_name /mounting point`**
    - Mount SAMBA resource using **CIFS** protocol:
      - **`$ mount -t cifs -o username=<uername>,noexec //server_ip/resource_name /mounting point`**

SAMBA: Client side configuration (Linux)

- **Static mount** (permanent):

  `$ vi /etc/fstab`

  - On this file we can define the mounting points on boot, remote file system and resources included:
    - **Every time system boots, the resources are mounted.**
    - **When system halts, the resources are unmounted.**
  - For NFS/SAMBA: one more entry.

    `$ mount -a`
  - Syntax (**File-based security**):

```
//server_ip/<resource_name>   /<local_directory>  cifs credentials=/etc/samba/user,noexec  0    0
```

Only if **kerberos** is not integrated by SAMBA.

- **We need to create the *credentials file:***
  - `$ vi /etc/samba/user`
  - Add the following lines:

    `username=<user name>`
    `password=<password>`
- **Advice → Change the file rights:**
  - `$ chmod 0600 /etc/samba/user`

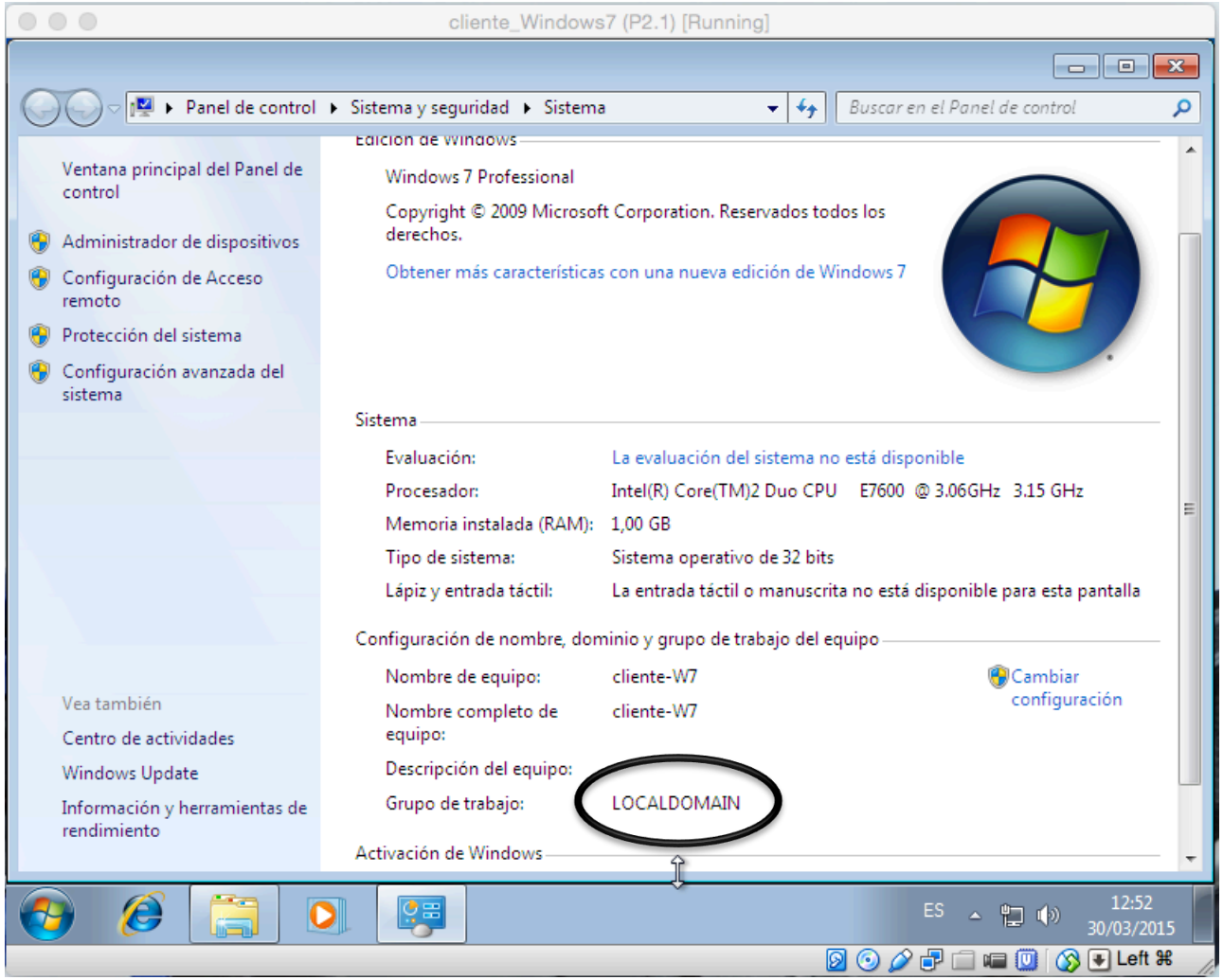
SAMBA: Client side configuration (Linux)

- Syntax (**with kerberos 5 security**):

```
//server/<shared directory>   /<local directory>  cifs sec=krb5,noexec  0    0
```

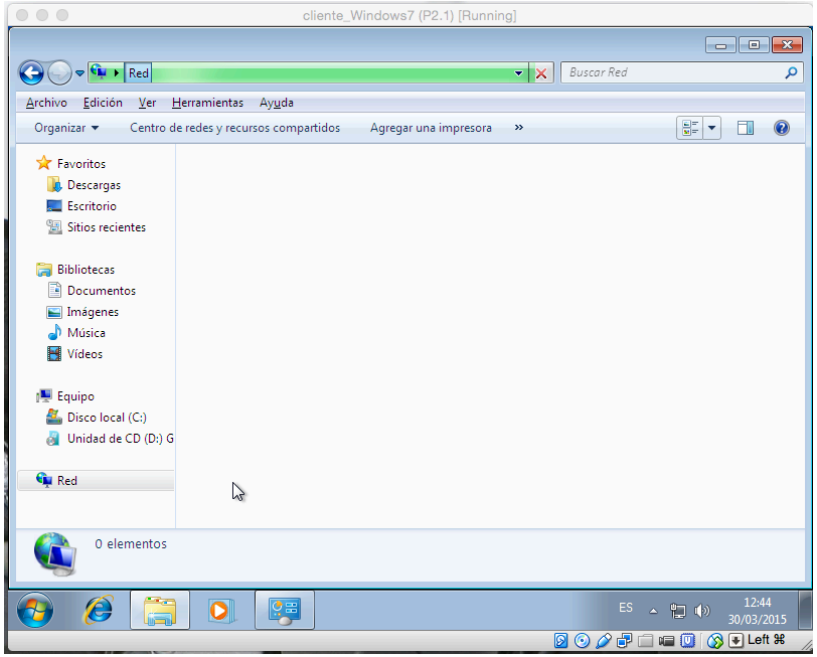
  - Remember that in this case, since it is a *"kerberized"* service (SAMBA + kerberos 5 security), we need the **service credential (kerberos) file** on client for that host (`keytab`):
    - The kerberos service key for this host:
      → **`cifs/hostname@DOMAIN`**

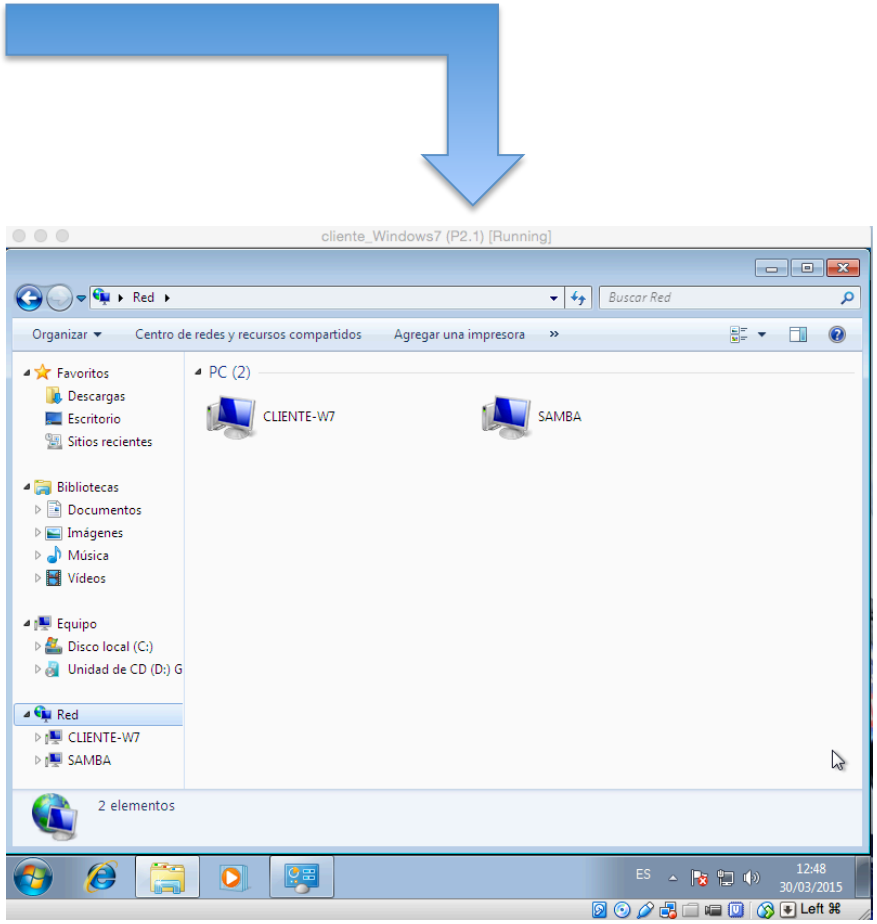
SAMBA: Client side configuration (Windows)



José Ángel Herrero Velasco

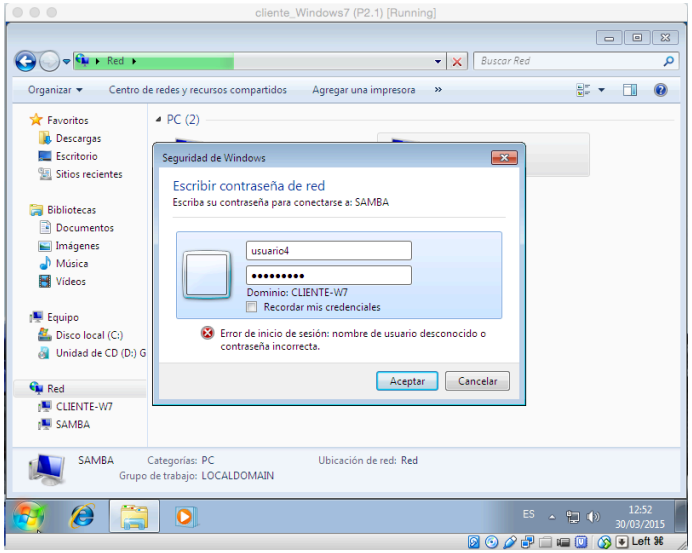
SAMBA: Client side configuration (Windows)

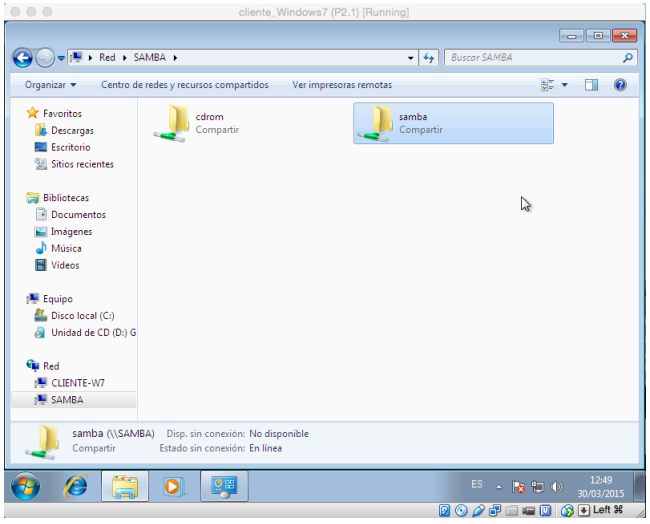


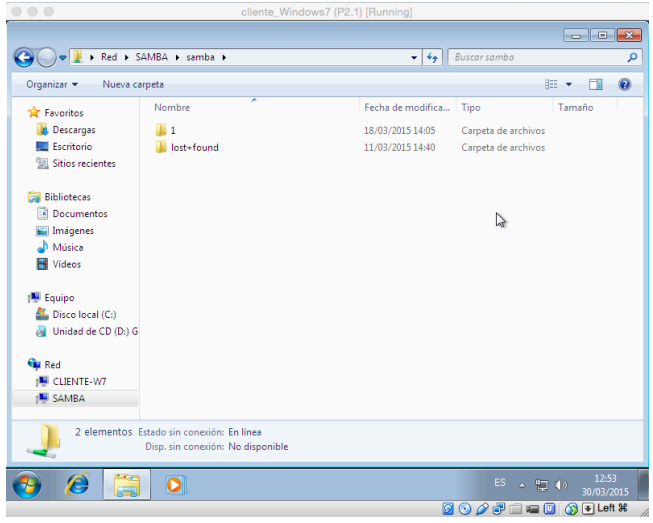


José Ángel Herrero Velasco

SAMBA: Client side configuration (Windows)







José Ángel Herrero Velasco

SAMBA: Basic management

- **Checking:**
  - Syntax of service configuration file:
    - `$ testparam`
- **To create, edit or delete local SAMBA users (*smbldap* tools):**
  - Create:
    - `$ smbldap-populate`
    - `$ smbldap-useradd -a -c "Samba user"…`
    - `$ smbldap-passwd <username>` → To change passwords.
  - Delete:
    - `$ smbldap-userdel <username> -x <username>`
- **More tasks…**
  - Authentication using **kerberos (credentials):**
    - `$ kinit <username>`
  - List SAMBA resources of an user:
    - `$ smbclient -k -L //<SAMBA server>/<resource>/ -U <username>`
  - Connection to SAMBA service share:
    - `$ smbclient -k //<SAMBA server>/<resource>/ -U <username>`