Networks

René Serral-Gracià   Xavier Martorell-Bofill[1]

[1] Universitat Politècnica de Catalunya (UPC)

May 26, 2014

Lectures

1. System administration introduction
2. Operating System installation
3. User management
4. Application management
5. System monitoring
6. Filesystem Maintenance
7. Local services
8. **Network services**
9. Security and Protection
10. Virtualization

Outline

1. Introduction
   - Goals
   - Previous Considerations
   - Network Address Translation
   - Firewall
2. Servers
3. Services

Goals

Knowledge

- Main services and networking protocols
  - Superserver, portmapper, DNS, FTP, WWW, e-mail
-

Abilities

- Service configurations
  - Superserver
  - DNS
  - FTP
  - WWW
  - E-Mail

Network admin considerations (I)

Security measures

- Never execute services with superuser privileges
- Expose only necessary services – firewalls
- Configure carefully all the offered services
  - Never leave default configurations
  - Disable/Remove unused services
- Monitor the service's logs
- Check for security issues – **be up to date**

Network admin considerations (and II)

Port classification

- Privileged ports: 0 - 1023
  - Controlled and assigned by IANA
  - Only privileged users (`root`) mai install services to those ports
- Registered ports: 1024 - 49151
  - Not controlled but registered by IANA
  - Registry about services using those ports – `/etc/services`
- Dynamic ports: 49152 - 65535
  - Used for temporary connections

/etc/services

- Relates services with corresponding port number
  - various applications use it (netstat,...)

```
servicename    port/protocol    alias list
```

```
echo            7/tcp
echo            7/udp
systat          11/tcp          users
systat          11/udp          users
ftp-data        20/tcp
ftp-data        20/udp
# 21 is registered to ftp, but also used by fsp
ftp             21/tcp
ftp             21/udp          fsp fspd
ssh             22/tcp
ssh             22/udp
telnet          23/tcp
telnet          23/udp
# 24 - private mail system
smtp            25/tcp          mail
smtp            25/udp          mail
domain          53/tcp
domain          53/udp
http            80/tcp          www www-http
http            80/udp          www www-http
```

Network Address Translation – NAT

- Router translates internal addresses by one (or various) of its own
  - Allows using a reserved IP (pool) and keep connectivity to the outside
- The router remembers the output connections to identify its answers
  - Output connection:
    - 192.168.1.25 (port 1085) → 212.106.192.142 (11086)
  - Reply connection:
    - 212.106.192.142 (11086) → 192.168.1.25 (1085)

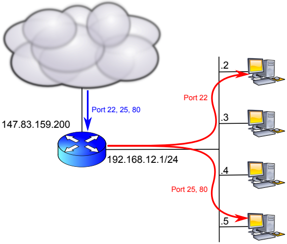
Tools: `iptables` (SNAT), `dnsmasq`

NAT collateral effects

- Private addresses are not visible from the outside
  - Attacks may only fall to the router – except over ongoing connections
- Network security depednds on router security
- Internal machines cannot offer services to the outside
  - Ecxept when using Port Address Translation (PAT)
- Important performance penalty for the network
  - All external connections go through a single router
  - Each packet requires some CPU time for processing
- Some services do not behave properly when using NAT
  - Those establishing connections to the inside
  - FTP, IRC, Netmeeting, . . .

Port Address Translation (PAT)

- Indicate to the NAT router it must forward some input connections to a particular machine
- Map router ports to some internal machine

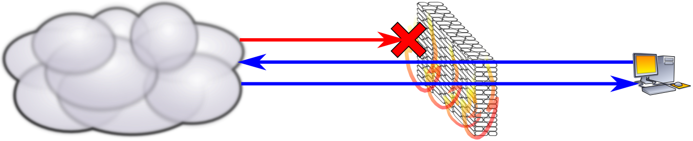


Eines: `iptables` (DNAT)

Firewall

Server that determines which connections may be established between two networks

- It typically works at network and transport layers
  - In general application details are not known
- It can keep connection status (Connection Tracking)
  - It allows related connections: “replies“



Firewall == Security?

- A firewall is another piece of the overall security of a system
- Its use can potentially offer a false security feeling
- Other aspects cannot be neglected
  - Correct application configuration
  - Perform regular security updates on installed software
  - Limit concurrent connections
- Other security tools in the private network and servers are still necessary

Outline

1 Introduction

2 Servers

- Server types

3 Services

Server types

- Connection oriented
  - The server keeps status about the different sessions
  - Better performance
  - Less error resilience
- Connectionless
  - There is no status about the client connections
  - There are no sessions
  - Requests must be self contained
  - Client request must contain all the required information
  - Better failure resilience and recovery

Server types – Depending authority

- Primary
  - They keep a copy of all the information
  - If there is mismatch in the stored information the primary takes precedence
  - There is one per service
- Secondary
  - Keep copies of the information
  - Performing periodic updates with the primary
  - There can be more than one per service
  - Load balancing
  - Are an implicit backup of the primary
- Cache (and/or proxies)
  - Keep –partial– copies of the most used information
  - More than one per service
    - Better performance
  - They can add security checks, filtering, log, . . .

Outline

1 Introduction

2 Servers

3 Services

- Remote Procedure Calls (RPC)
- Domain Name System (DNS)
- Dynamic Host Configuration Protocol (DHCP)
- Hypertext Transfer Protocol (HTTP)
- File Transfer Protocol (FTP)
- Simple Mail Transfer Protocol (SMTP)
- E-mail reception
- Secure Shell
- Network File System (NFS)
- Samba (SMB)

Superserver

- A service even when idle uses resources
  - Many services are requested only from time to time: `telnet`, `ftp`, `ssh`, ...
- Superserver listens to all the ports and activates the service only when needed
  - It detects the request
  - Initiates the service
  - Passes the message
- Limitations
  - Between connections it is not possible to keep information in memory
  - Overhead caused by process creation

Implementations: `inetd`, `xinetd`

/etc/xinetd.conf, /etc/xinetd.d

Indicates the services offered by the superserver

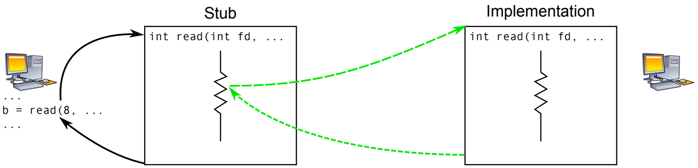
`Service, Protocol, User/group, Server, Parameters`

```
$ cat /etc/xined.conf
includedir /etc/xinetd.d
```

```
$ cat /etc/xined.d/ftp
service ftp
{
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/vsftpd
        log_on_success         += HOST DURATION
        log_on_failure         += HOST
        disable                 = no
}
```

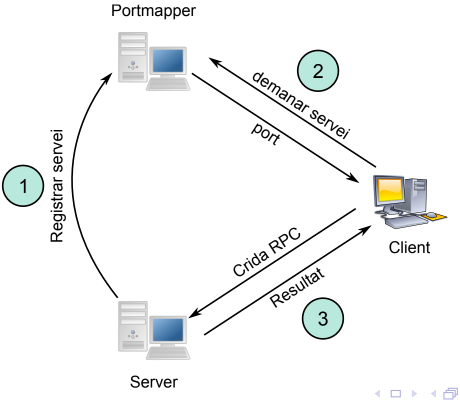
Remote Procedure Calls (RPC)

- Remote subroutine invokation
    - Identified by a service number ID
- RPC Servers
    - They implement a set of remote connections
    - Listen in a dynamic port
- Portmapper
- Registers the RPC servers
    - Maps the port with the subroutines
- Needed by other services
    - NFS, . . .



Portmapper

- All the status is kept on memory
  - If the process fails, is not enough restarting it
  - All RPC servers must be restarted
- All services must be registered upon portmapper start



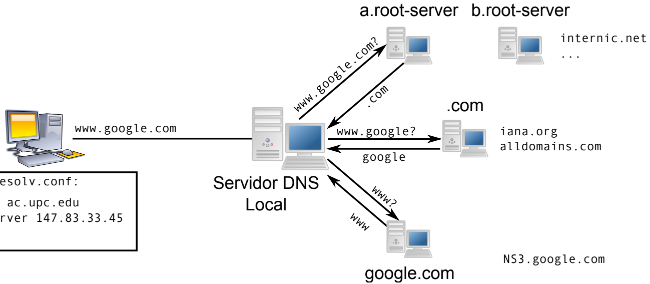
Domain Name System (DNS)

- Name resolution service
  - Hostname → IP address
  - IP Address → hostname
- Issues
  - Large amount of machines
  - Large number of changes
- Solution
  - Hierarchical distribution of the information (domains)
  - Authority delegation

DNS Internals

Authority delegation

- Each domain administers its own server
- Everybody knows the higher servers in the hierarchy (root)
- Everybody knows the server for their domain
- Name resolution is iterative



```
/etc/resolv.conf:
search ac.upc.edu
nameserver 147.83.33.45
```

DNS: RFCs 1034/1035

Service performance

Using ”caches“ is convenient

- High temporal locality
  - Avoids repeating the same query
- High spacial locality
  - Avoids going up to the root servers too often
  - Avoids some steps of the iterative search

DNS can be used for load balancing

- We can have several IPs for the same name
  - Each query returns different values: Round Robin or ”geographical“ criteria

```
$ nslookup www.google.com
Name:    www.google.com
Address: 212.106.221.23
Name:    www.google.com
Address: 212.106.221.27
Name:    www.google.com
Address: 212.106.221.25
...
```

DNS client configuration

- `/etc/host.conf`
  - Where a name is searched and its order
- `/etc/hosts`
  - Locally translated machines
- `/etc/resolv.conf`
  - Automatic domains to be searched
  - IP addresses of the DNS servers

DNS Server configuration

- `/etc/bind/named.conf`
  - What are we administering?
    - DNS Domains
    - IP addresses ranges
  - Indicates primary, secondary, or cache
- Direct translation files
  - Name.domain → IP address
  - 1 file for each administered domain
- Inverse translation file
  - IP Address → name.domain
  - 1 file for each administered IP range

DNS type of registers

- SOA (Start of Authority)
  - Serial number
  - Refresh time and retries
  - Expiration times
  - Minimum TTL
- A - Direct translation
  - Name → IP address

```
romeu IN A 147.83.32.4
```

- CNAME - synonyms
  - Name → name

```
romeu IN CNAME lp_romeu
```

DNS type of registers

- PTR - inverse translation
  - IP Address → DNS name

```
4 IN PTR romeu.ac.upc.edu.
```

- NS - Domain delegation
  - DNS Domain→ server

```
ac IN NS 147.83.32.3
```

- MX - mail exchanger
  - DNS Domain → server

```
ac IN MX 147.83.33.10
```

- I altres. . .
  - HINFO, WKS, . . .

DNS configuration example

Zone ”cluster.craax.upc.edu“, as primary.

```
$ cat /etc/bind/named.conf
options {
        directory "/var/cache/bind";
        forwarders {
                147.83.159.217;
        };
        auth-nxdomain no;    # conform to RFC1035
        listen-on-v6 { any; };
};
zone "cluster.craax.upc.edu"  {
  type master;
  file "/etc/bind/cluster.zone";
};

zone "1.1.10.in-addr.arpa"  {
  type master;
  file "/etc/bind/cluster.rev";
};
```

DNS configuration example

```
$ cat /etc/bind/cluster.zone
$TTL    604800
@       IN      SOA     cluster. cluster.craax.upc.edu. (
                       20101220                ; Serial
                         604800                ; Refresh
                          86400                ; Retry
                        2419200                ; Expire
                         604800 )              ; Negative Cache TTL
;
@       IN      NS      gandalf
$ORIGIN                 cluster.craax.upc.edu.
gandalf         IN      A       10.1.1.1
boromir-1       IN      A       10.1.1.2
```

```
$ cat /etc/bind/cluster.rev
$TTL    604800
@       IN      SOA     cluster. cluster.craax.upc.edu. (
                       20101220                ; Serial
                         604800                ; Refresh
                          86400                ; Retry
                        2419200                ; Expire
                         604800 )              ; Negative Cache TTL
;
@       IN      NS      gandalf
$ORIGIN                 cluster.craax.upc.edu.
1       IN      PTR     gandalf.cluster.craax.upc.edu.
2       IN      PTR     boromir-1.cluster.craax.upc.edu.
```

Exercise

- We have 3 services at (`server1`, `server2` i `server3`) with these registers

```
server1 IN A 123.123.123.1
server2 IN A 123.123.123.2
server3 IN A 123.123.123.3
```

- We want to add the following services
  - `www` at `server1` (`server2` is the backup server)
  - `ftp` at `server1` and `server2`
  - incoming/outgoing mail at `server3`

Which registries would you add?

DNS Related tools

- `whois domain`
  - Provides contact information for a domain
- `dig [@server] query`
  - Performs a DNS query
  - It allows controlling different resources
    - Server, type of register, iterative/recursive resolution, . . .
  - Returns the registers corresponding to the query
    - It supports debugging

Dynamic Host Configuration Protocol (DHCP)

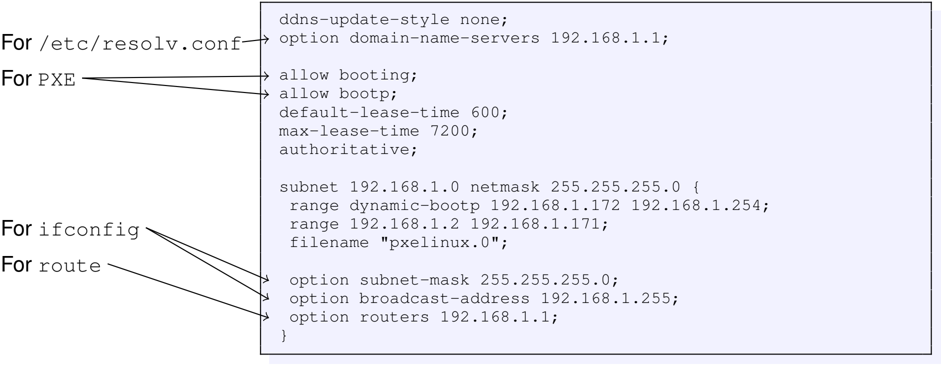
- It delivers automatically the network configuraiton to a host
  - IP assignation, Gateway and DNS
- Machine trustfulness is not verified
  - By default it is assumed that if the host can reach connectivity then it is legitimate
  - It can provide MAC address verification
- IP addresses are assigned from a predefined range

Dynamic Host Configuration Protocol (DHCP)

Remote boot support through BOOTP and PXE

- Preboot Execution Environment (PXE)
- Network card uses BIOS to get network information
- It allows to decide the kernel image to boot
  - Downloaded through TFTP
  - A remote root system can be mounted

Dynamic Host Configuration Protocol (DHCP)



For /etc/resolv.conf

For PXE

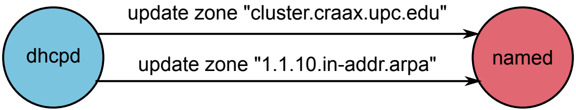
For ifconfig

For route

```
ddns-update-style none;
option domain-name-servers 192.168.1.1;

allow booting;
allow bootp;
default-lease-time 600;
max-lease-time 7200;
authoritative;

subnet 192.168.1.0 netmask 255.255.255.0 {
 range dynamic-bootp 192.168.1.172 192.168.1.254;
 range 192.168.1.2 192.168.1.171;
 filename "pxelinux.0";

 option subnet-mask 255.255.255.0;
 option broadcast-address 192.168.1.255;
 option routers 192.168.1.1;
}
```

Dynamic Host Configuration (DHCP)

DHCP and DNS can work together



dhcpd

update zone "cluster.craax.upc.edu"

update zone "1.1.10.in-addr.arpa"

named

/etc/dhcpd/dhcpd.conf

```
ddns-update-style interim;
key DHCP_UPDATER {
   algorithm HMAC-MD5.SIG-ALG.REG.INT;
   secret pRP5FapFoJ95JEL06sv4PQ==;
};
zone ac.upc.edu. {
   primary 192.168.1.1;
   key DHCP_UPDATER;
}
```

/etc/bind/named.conf

```
key DHCP_UPDATER {
   algorithm HMAC-MD5.SIG-ALG.REG.INT;
   secret pRP5FapFoJ95JEL06sv4PQ==;
};
zone ac.upc.edu. {
   type master;
   file "ac.zone";
   allow-update { key DHCP_UPDATER; };
};
...
```

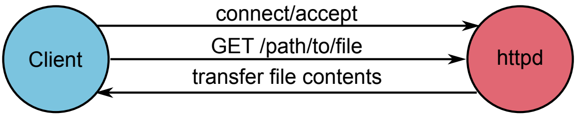
Exercise

In group

- Which potential problem can be caused by a DHCP server crash?
- Propose an implementation to solve it

Hypertext Transfer Protocol (HTTP)

- Data transfer service
- Connectionless
  - There is no state between connections
  - Each petition is self-contained
- Nevertheless it uses TCP



Apache Web Server

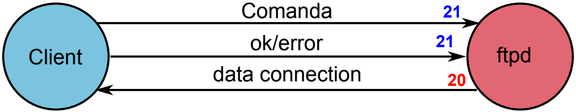
- Implements support for HTTP
- `/etc/apache/httpd.conf`

Main features

- Unprivileged user execution
- Queries are served using memory separated processes/threads
  - Memory sharing configurable by the administrator
  - Maximum concurrent processes limit
- Configuration options in a per directory basis
- Virtual Host configuration
  - By IP address
  - By DNS name

File Transfer Protocol (FTP)

- Data transfer service
- Connection oriented
- Control connection
  - There is state between connections: `cwd`
- Data connection
  - active: does not support NAT
  - passive: NAT is supported
  - There is a new data connection per transfer



FTP Configuration

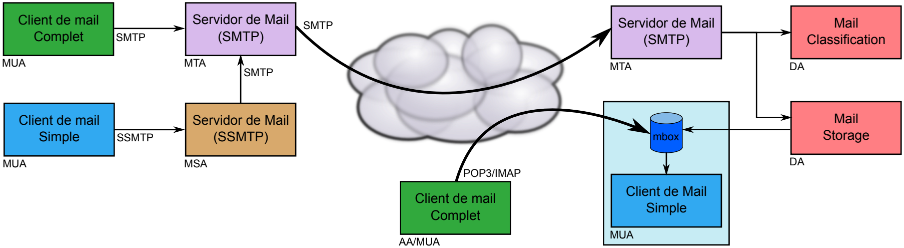
- There are many server implementations
  - `wu-ftpd`, `proftpd`, `vsftpd`, ...
- User level based authorization: `/etc/ftpusers`
  - List of the users that **CAN'T** access FTP
- Use `chroot` for security in Aonymous FTP
  - Changes the root of the process
  - Extra configuration
  - Requires install basic commands and configuration files
    - `/etc/passwd`, `/etc/shadow`
    - `/bin/ls`, `/lib/libc.so`, ...
  - Use it even for regular users

Simple Mail Transfer Protocol (SMTP)

Parts composing the mail system

- MUA - Mail User Agent
  - User application to read/write e-mails
- MSA - Mail Submission Agent
  - Application to transmit the mail from the client to the MTA
  - It make all previous error checking
- MTA - Mail Transport Agent
  - It sends the e-mail between servers
- Delivery Agent
  - Application to store mails into the user's mailbox
  - Sometimes the mails are stored into a database
- Access Agent
  - Application allowing the user to access its e-mail

Mail system components



Internals of an e-mail

- Envelope
  - Message destination
  - Source
  - Not received by the clients – only for servers
- Headers
  - Set of message properties
    - Sending date
    - Source and destination (shown by the e-mail clients)
    - List of servers the message has crossed
- Message body
  - Uses 7 bits ASCII
  - Attachments use Base-64

Mail client configuration

Mail reception

- Access to local mailbox
  - Mailbox/maildir format interpreter
- Remote mailbox access
  - POP3
  - IMAP

Mail sending

- Using an SMTP server

E-Mail server configuration

Mail sending – sendmail/postfix

- Sending direct to the destination
  - Search for MX record in DNS – local destination
- Sending through a Relay
  - No direct access to the destination

Mail reception

- Store the mails locally
  - POP3, DIMAP
- Store the mails in the remote server
  - IMAP

E-Mail server configuration

Mail aliases

- Redirect mail to other users
  - In a remote machine
- Users with multiple names
  - `root`, `postmaster`, `webmaster` → `usuari@host`
- Send a mail to a file instead of a user
  - spam: `/dev/null`
- Send the e-mail to a program
  - autoftp: ”| `/usr/bin/ftpserver`“
- Mailing list definition
  - Is better to use: Majordomo, Mailman, ListProc, SmartList, ...

Aliases defined in `/etc/aliases` or `/etc/mail/aliases`

Security considerations

User authentication

- By default the server does not ask for credentials
  - SASL can be used
- Envelope can be forged — SPAM . . .
- Trust mail relays
  - The server always tries to send the message
  - Even if the headers do not belong to the domain (Open Relays)

Security considerations

Mail privacy

- Mail is sent in plain text
  - Use of TLS (SSL) only between MUA and MTA
- PGP - Pretty Good Privacy
  - Message cyphering and signing
  - Based in public key cryptography
- S/MIME

Filter installation

- Anti-spam
  - `Spamassasin`, gray lists, black lists, . . .
- Anti-virus
  - `Clam AV`, `Amavis`, `f-prot`,. . .

Exercise – In group

We just set up a filter to control spam

- Which action would you take as a server when you detect a spam message?
- And if the filter is an anti-virus?

E-mail reception

Post Office Protocol (POP)

- It allows users to access their mailbox
- It downloads the messages to the local machine
- Authentication without encryption
  - `pop3s` secure alternative using SSL

Internet Message Access (IMAP)

- It allows users to manage their mailbox
- Management is performed remotely
- User authentication
  - Allowing encryption
- `imaps` even more secure alternative using SSL

Secure Shell

- It substitutes rsh/rlogin and telnet
- Adding security
  - It performs authentication based on RSA, DSA, ECDSA
    - Session key is signed by the client's private key
    - The server uses the public key as stored in (`.ssh/authorized_keys`) to check if the signature is correct
    - password based authentication is also supported
  - Connection is fully encrypted
    - Confidentiality: `3DES`, `Blowfish`, `...`
    - Integrity: `hmac-md5`, `...`
- The server runs the specified command or offer a shell
- Transparent session
  - Whenever a pseudo-terminal is not requested
  - It can be used to transfer binary files
- login Session
  - X11 Protocol forwarding can be configured

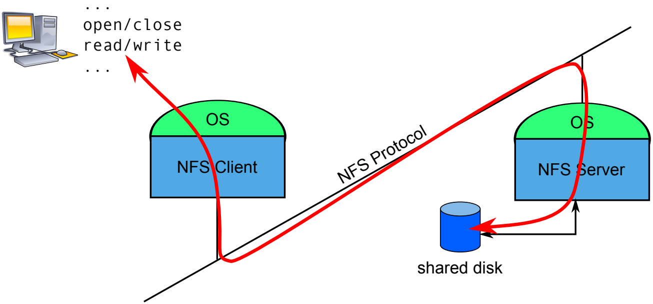
Exercise – In group

Secure Shell actions

- How would you implement secure copy and secure FTP directly with ssh?

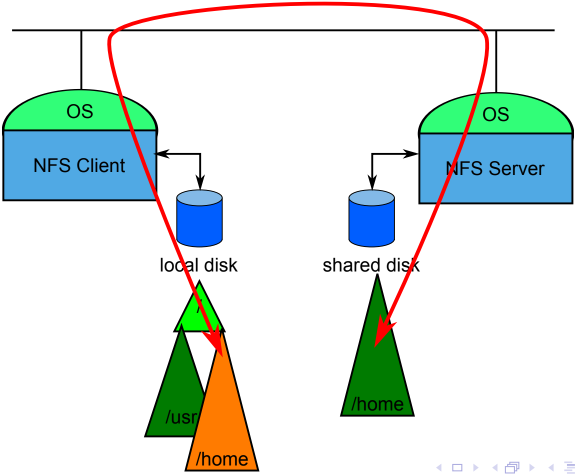
Network File System (NFS)

- File access in a remote server
  - Keeping the semantics (privilege wise) of the local filesystem
- It is transparent to the user
  - Implemented using RPC's



Remote mounting for NFS

The mounted directory is presented as local



Access privileges

- UIDs in the remote machines must be the same as used in local
  - Filesystems store UID rather than usernames
  - This can be adapted by using `idmapd`
- UID automatic translation (`idmapd`)
  - `root`, `nobody`
- Options
  - `no_root_squash`, root can su to any user!
  - `all_squash`, all users become `nobody`
  - We can decide who nobody is

```
anonuid=UID,anongid=GID
```

NFS Configuration

- Determine which resources to export
- Hosts to export to
- Configuration flags

/etc/exports

```
/              master(rw) trusty(rw,no_root_squash)
/projects      proj*.local.domain(rw)
/usr           *.local.domain(ro) @trustedgroup(rw)
/home/joe      pc001(rw,all_squash,anonuid=150,anongid=100)
/pub            (ro,insecure,all_squash)
```

SMB — Samba

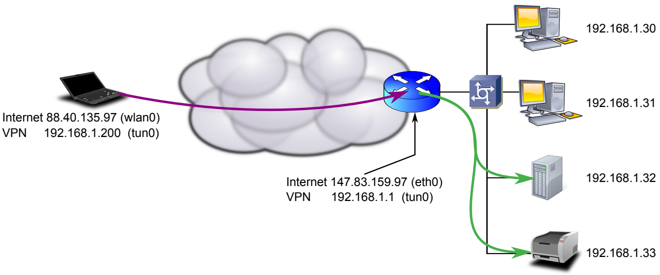
- It allows sharing files and printers
- User level access control
    - Authentication using login and password
        - Based on username not UID
        - Encripted and plaintext password transmission
    - Machine based access restriction
        - It does not allow to change permissions depending on the source
        - One must use different share names

Lightweight Directory Access Protocol (LDAP)

- It provides access to users database
  - Directory format (X.500)
- It offers user authentication methods
  - `/etc/passwd`, `/etc/shadow`, `/etc/group`, . . .
  - . . . they can be dumped to the LDAP database
- Besides regular files, login can also be controlled through the database

Virtual Private Networks (VPN)

- Server and client negotiate a secure connection
- An internal IP is offered through a secure tunnel
  - It grants access to all the internal services


Internet 88.40.135.97 (wlan0)
VPN 192.168.1.200 (tun0)
Internet 147.83.159.97 (eth0)
VPN 192.168.1.1 (tun0)
192.168.1.30
192.168.1.31
192.168.1.32
192.168.1.33