Serveis web i de transferència de fitxers

Eduard Canet i Ricart

Índex

Introducció

Segurament els serveis més populars que estudiarem en el mòdul *Serveis de xarxa i Internet* són els serveis d'HTTP i FTP tractats en aquesta unitat. Aquests són els serveis que permeten la creació de llocs web i de servidors de descàrrega de fitxers.

El servei més popular avui en dia a Internet és el servei web, que utilitza HTTP. La seva popularitat, basada en el tractament d'hipertext que ha acabat incloent vídeo, àudio i multimèdia en general (hipermèdia), l'ha convertit en una eina a l'abast de tothom. L'ús dels navegadors web i HTTP ha eclipsat molts dels altres protocols d'Internet, que han acabat veient com les seves funcionalitats s'integraven en el servei web (els usuaris baixen fitxers pel web en lloc de per l'FTP). El servidor intermediari (*proxy server*) és un servei HTTP que proporciona capacitats de memòria cau i filtratge dels continguts web que sol·liciten els clients.

En l'apartat **"Instal·lació i administració de servidors web"** es descriuen els fonaments i protocols en els quals es basa el funcionament d'un servidor web, el protocol HTTP. S'explica la sintaxi d'aquest protocol i es descriu un diàleg petició/resposta entre un client (per exemple, un navegador) i un servidor web. També es mostra com instal·lar i configurar servidors web i s'examina la configuració per defecte.

La funcionalitat del servidor es pot ampliar a través de mòduls dinàmics. Així doncs, es mostra com activar i configurar mòduls dinàmics, com per exemple els que proporcionen SSL, PAM, estadístiques i monitorització del servidor, etc. S'explica com crear i configurar llocs web virtuals, segurament un dels elements més importants de la configuració de servidors web. Els llocs virtuals permeten disposar de múltiples llocs web en un mateix servidor.

Un altre aspecte molt important és com gestionar l'accés als llocs web, qui té o no té permís per accedir a on. En aquest apartat aprendreu a instal·lar i configurar els mecanismes d'autenticació i control d'accés del servidor. Una de les preocupacions principals a Internet és la seguretat. Es mostra com obtenir i instal·lar certificats digitals i com establir mecanismes per assegurar les comunicacions entre el client i el servidor.

També es presenta com realitzar proves de monitoratge del servei, analitzar els registres del servei per a l'elaboració d'estadístiques, resoldre incidències i generar documentació.

En l'apartat **"Instal·lació i administració de serveis de transferència de fitxers"** s'estudien els serveis FTP i TFTP, que permeten penjar i baixar fitxers en la xarxa. L'FTP utilitza TCP i proporciona fiabilitat en les transferències. Permet l'accés tant d'usuaris identificats com d'anònims. El TFTP utilitza UDP i és un mecanisme sense fiabilitat, però molt usat per a descàrregues en àrees locals. Els clients lleugers o els sistemes que s'inicien de xarxa utilitzen TFTP per transferir la informació.

Es realitza una descripció del protocol i s'analitza un diàleg complet client/servidor. També es mostra com instal·lar i configurar servidors de transferència de fitxers, examinar la configuració per defecte i personalitzar-la per tal de satisfer els requeriments del lloc FTP.

Un altre aspecte és aprendre a gestionar els usuaris i l'accés als recursos. Es mostra com crear usuaris i grups per a l'accés remot al servidor i com configurar l'accés anònim. També s'indica com establir limitacions en els diferents modes d'accés.

Així mateix, es repassa exhaustivament cada un dels modes de connexió, tant en mode actiu com en mode passiu. I també es realitzen proves amb clients en línia d'ordres i amb clients en mode gràfic. Especialment es tracta la utilització del navegador com a client del servei de transferència de fitxers.

Per oferir seguretat, integritat i confidencialitat als serveis que originàriament no en proporcionen, s'han desenvolupat tècniques com l'SSL i el TLS, que han donat lloc als serveis HTTPS i FTPS. També han sorgit protocols com l'SSH, que permeten un model de transferència d'informació xifrada.

Resultats d'aprenentatge

En finalitzar aquesta unitat, l'estudiant:

1. Administra servidors web aplicant criteris de configuració i assegurant el funcionament del servei.
    - Descriu els fonaments i protocols en els quals es basa el funcionament d'un servidor web.
    - Instal·la i configura servidors web.
    - Amplia la funcionalitat del servidor activant i configurant mòduls.
    - Crea i configura llocs web virtuals.
    - Configura els mecanismes d'autenticació i control d'accés del servidor.
    - Obté i instal·la certificats digitals.
    - Estableix mecanismes per assegurar les comunicacions entre el client i el servidor.
    - Realitza proves de monitoratge del servei.
    - Analitza els registres del servei per a l'elaboració d'estadístiques i la resolució d'incidències.
    - Elabora documentació relativa a la instal·lació, configuració i recomanacions d'ús del servei.
2. Administra serveis de transferència de fitxers assegurant i limitant l'accés a la informació.
    - Estableix la utilitat i el mode d'operació del servei de transferència de fitxers.
    - Instal·la i configura servidors de transferència de fitxers.
    - Crea usuaris i grups per a l'accés remot al servidor.
    - Configura l'accés anònim.
    - Estableix limitacions en els diferents modes d'accés.
    - Comprova l'accés al servidor, tant de manera activa com passiva.
    - Realitza proves amb clients de línia d'ordres i amb clients gràfics.
    - Utilitza el navegador com a client del servei de transferència de fitxers.
    - Elabora documentació relativa a la instal·lació, configuració i recomanacions d'ús del servei.

1. Instal·lació i administració de servidors web

L'**HTTP** (*Hypertext Transfer Protocol* o **protocol de transferència d'hipertext**) és un protocol de capa d'aplicació que proporciona transferència de documents d'hipertext al web. El protocol HTTP és omnipresent: el World Wide Web (WWW) permet baixar hipertext, multimèdia i altres tipus de dades.

L'HTTP és un protocol sense estat i no orientat a la connexió permanent.

L'HTTP està basat en un esquema client/servidor en què el client es connecta al port 80 del servidor i fa una sol·licitud (una pàgina web, per exemple), i el servidor emet la resposta corresponent i tanca la connexió. Es tracta, per tant, d'un protocol sense estat. La connexió entre client i servidor sovint s'inicia i es tanca en cada petició/resposta. L'HTTP utilitza habitualment el protocol de transport TCP per obtenir fiabilitat en la comunicació.

L'**HTTP** és un protocol de capa d'aplicació que proporciona transferència de documents d'hipertext a la web. Utilitza un mecanisme client/servidor al port 80 basat en TCP.

L'especificació actual del protocol HTTP és la 2 (HTTP/2), descrita al document RFC 7540, de maig de 2015 (que en descriu l'estàndard), tot i que hi ha l'esborrany de la versió 3 (HTTP/3). Aquesta especificació és un alternativa a l'anterior especificació, la 1.1 (descrita al document RFC 2616, de juny de 1999) i que no deixa obsoleta. L'HTTP sorgeix als anys noranta com a protocol per transferir documents hipermèdia "en cru" per Internet (versió 0.9). La versió HTTP 1.0 (RFC 1945) permet el pas de missatges utilitzant un format tipus MIME (usat en el transport de correu). Originàriament, el contingut dels documents a transferir era text, però amb la popularització del WWW s'ha acabat convertint en un protocol de transport de contingut multimèdia i no únicament hipertext. A més, l'HTTP s'utilitza sovint com a protocol de comunicacions entre clients i altres sistemes d'Internet diferents del WWW, com per exemple NEWS, SMTP, NNTP, FTP, Ghoper, servidors intermediaris (*proxies*) i d'altres, per accedir a aquests recursos.

MIME

El *Multipurpose Internet Mail Extension*, o extensió de propòsit múltiple per al correu, és el mecanisme utilitzat per descriure el contingut dels fitxers, saber si són una imatge, un full de càlcul, un executable o altres, i permetre al navegador obrir l'aplicació pertinent.

En parlar d'HTTP ens venen immediatament al cap els navegadors web (tipus Firefox, Google Chrome, Safari...), que permeten visualitzar des d'entorns gràfics contingut d'hipertext i multimèdia, també anomenat hipermèdia. De fet, però, també existeixen navegadors en mode text (no gràfics) per a contingut únicament de text (per exemple Lynx). Habitualment usem els navegadors per obtenir contingut HTTP, però la majoria d'ells ens permeten accedir a recursos d'altres tipus.

Per accedir als documents publicats en el WWW o a documents interns de la xarxa corporativa, cal un mecanisme d'adreçament universal. L'URI (Uniform Resource Identifier o identificador uniforme de recursos) és el mecanisme d'identificació de recursos universal i té la sintaxi *schema:identifier* (esquema:identificador).

URI

Sovint s'utilitzen indistintament URI i URL, tot i que no són el mateix. Un URI es pot classificar com un URL, un URN o ambdós. L'URI permet identificar elements globalment, l'URL localitzar-los i l'URN proporciona un mecanisme d'assignació de noms únic.

L'esquema descriu la sintaxi utilitzable per l'identificador i pot ser HTTP, HTTPS, FTP o Ghoper, entre d'altres. L'identificador permet determinar el recurs concret dins d'aquest esquema. Usualment, en HTTP s'utilitza un subconjunt de l'URI anomenat URL (Uniform Resource Locator o localitzador uniforme de recursos) per localitzar un recurs. Així, en les barres de navegació trobem URL com http://www.uoc.es o ftp://ftp.rediris.es, per exemple.

La identificació de recursos es realitza mitjançant URI, URL o URN segons correspongui. La sintaxi és la següent:
URI = Uniform Resource Identifier (esquema:identificador)
URL = Uniform Resource Locator (http://www.escoladeltreball.org)
URN = Uniform Resource Name (ietf:rfc:2616)

1.1 Funcionament del servei web

El protocol HTTP estructura el diàleg client/servidor en un esquema molt bàsic de petició/resposta. Fins a la versió HTTP 1.0, cada petició/resposta implicava una connexió que s'obria i es tancava en finalitzar la resposta. Amb les millores introduïdes en la versió HTTP 1.1, s'introdueix un mecanisme de connexions persistents. La connexió establerta es pot mantenir un temps oberta per realitzar més peticions dins de la mateixa connexió (per exemple, baixar altres components de la pàgina). La versió 2 incorpora millores per tal de reduir la latència en la càrrega de pàgines web, a apart de mantenir una alta compatibilitat amb la versió anterior.

Connexions persistents

Les connexions persistents permeten que els múltiples elements d'una pàgina web (que es troben en fitxers diferents) es puguin baixar sense que calgui una connexió per a cada element.

El protocol HTTP és un protocol sense estat (*statless*). Ni client ni servidor mantenen un estat de sessió a nivell de protocol. Segurament us heu connectat a un servidor de correu web (*webmail*) i heu establert una sessió d'usuari mentre consulteu el correu. Aquesta sessió no s'implementa a nivell del protocol HTTP, sinó que és responsabilitat del desenvolupador web mantenir l'estat. Això es fa generalment utilitzant tècniques com l'ús de galetes (*cookies*), passant paràmetres per l'URL, amb camps ocults (típic dels formularis)...

L'**HTTP** és un protocol sense estat que usualment tanca la connexió per a cada petició/resposta.

1.1.1 Descripció del diàleg petició/resposta

En els diàlegs HTTP el client usualment emet una petició al servidor indicant algun dels mètodes que permet el servidor (no necessita implementar-los tots). El servidor emet una resposta i l'acompanya d'un valor d'estatus que indica el tipus de resposta (OK, error...).

Petició ('request')

El diàleg HTTP s'inicia quan un client fa una petició (usualment d'una pàgina web) a un servidor (usualment al port 80). Aquest missatge de petició consta d'una primera línia anomenada *línia de petició*, seguida de capçaleres, una línia en blanc i el cos de la petició:

Components d'una petició

Els components d'un missatge de petició HTTP són quatre:

- Línia de petició
- Capçaleres
- CRLF
- Cos

- **Línia de petició**: la primera línia d'una petició sempre té la mateixa estructura, per exemple: GET /docums/fitxa.html HTTP/1.1. El primer camp és el mètode a usar (GET significa "petició"), el camp següent és el document a obtenir i el tercer indica la versió del protocol HTTP que s'utilitza. Aquesta primera línia ha d'acabar sempre amb els caràcters CRLF.
- **Capçaleres (*headers*)**: a continuació es troben les capçaleres de la petició. Les capçaleres permeten descriure opcions del client i opcions preferibles del servidor. Per exemple, el client pot indicar el sistema operatiu i el navegador que utilitza, i el servidor ho pot tenir en compte a l'hora d'efectuar la resposta. Hi ha multitud de capçaleres i es recomana consultar el document RFC 2616 (que descriu l'estàndard HTTP) per ampliar-ne la informació. La capçalera *Host:* és obligatòria en HTTP 1.1 i indica l'URL del servidor al qual s'adreça la petició.

Diferència entre HTTP1.0 i HTTP1.1

Una de les diferències entre HTTP/1.0 i HTTP/1.1 és que en HTTP/1.1 hi ha una capçalera obligatòria (Host: <nom_servidor>) i en HTTP/1.0 no. Això li permet al servidor saber si la petició és per a ell, i permet implementar seus virtuals.

* **CRLF (línia en blanc)**: una línia en blanc separa la part de capçaleres de la petició de la part del cos. Aquest mecanisme està manllevat del format dels missatges de correu, on també s'utilitza una línia en blanc per separar les capçaleres del cos dels missatges.

- **Cos (body)**: el cos del missatge és opcional i no s'utilitza usualment en les peticions.

Mètodes de les peticions

Les peticions HTTP contenen un mètode en el primer camp de la primera línia. Aquest acostuma a ser GET o POST en les peticions, però n'hi ha més:

- **HEAD**: igual que GET però únicament sol·licita la capçalera del document. S'utilitza per comprovar l'existència del document.
- **GET**: petició al servidor per obtenir el document sol·licitat.
- **POST**: envia al servidor informació que ha d'incorporar al recurs de destinació especificat. Un ús habitual és en els formularis, on les dades es passen per POST perquè el servidor les incorpori en el document de destinació indicat.
- **PUT**: permet posar en el servidor el document indicat. En lloc de baixar un document, és un mètode per penjar un document en el servidor.

- **DELETE**: elimina el document indicat del servidor. Si es deixa, és clar.
- **TRACE**: el servidor retorna com a missatge una còpia del missatge tal com li ha arribat. És molt útil per al monitoratge del servei per part del client, ja que pot veure quines transformacions ha patit la seva petició en creuar passarel·les o *gateways*, servidors intermediaris...
- **OPTIONS**: és una sol·licitud d'informació de les opcions de transferència del servidor. El servidor contesta indicant quines són les seves capacitats.

Resposta

El servidor respon les peticions del client amb missatges que tenen una estructura similar a les peticions. Consten d'una primera línia, anomenada *línia d'estatus*, seguida de les capçaleres, una línia en blanc i la resposta, que va al final:

- **Línia d'estatus**: la primera línia d'una resposta té sempre un format com *HTTP/1.1 403 Accés prohibit*. El primer camp indica el protocol HTTP usat. El segon camp és un valor numèric de tres dígits que indica el tipus de resposta donada. Hi ha una llista exhaustiva de valors d'estatus i de significats. L'últim camp és un text descriptiu de l'estatus.
- **Capçaleres (*headers*)**: la resposta conté totes les capçaleres que el servidor consideri oportú incloure.
- **CRLF**: una línia en blanc separa les capçaleres del cos de la resposta.
- **Cos (*body*)**: aquesta part conté el contingut "real" de la resposta pròpiament dit. Així si per exemple s'ha sol·licitat una pàgina web, el contingut a mostrar es troba aquí (tota la pàgina web, no us confongueu amb les etiquetes HEADER i BODY del llenguatge HTML).

Estatus de les respostes

Les respostes contenen un primer camp amb un valor numèric d'estatus. En el document RFC 2616 se'n pot trobar la llista completa, però segons quin sigui el primer dígit es pot fer la classificació següent:

- 1xx: informació genèrica
- 2xx: acció amb èxit, *successful*
- 3xx: redirecció
- 4xx: error del client
- 5xx: error del servidor

1.1.2 Exemples de connexions HTTP

Tot el diàleg client/servidor té forma d'ordres i respostes, com en l'exemple següent de connexió HTTP 1.0 al servidor local. Vegeu com es realitza una connexió HTTP per mitjà d'un Telnet al port 80 d'un servidor HTTP per fer una petició GET d'una pàgina web:

```
root@server:~# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 14 Sep 2020 07:59:09 GMT
Server: Apache/2.4.38 (Debian)
Last-Modified: Mon, 07 Sep 2020 20:01:44 GMT
ETag: "a8-5aebeb015df61"
Accept-Ranges: bytes
Content-Length: 168
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

<html>
    <head>
        <title>Pàgina principal</title>
    </head>
    <body>
        <h1>Pàgina principal</h1>
        <p>Servidor Apache</p>
    </body>
</html>
Connection closed by foreign host.
root@server:~#
```

En l'exemple es pot fer un seguiment dels elements que intervenen en una comunicació HTTP. La petició client és una petició GET, seguida d'una línia en blanc i sense cos (el GET no en requereix). La resposta del servidor comença amb una primera línia d'estatus (el valor 200 indica "OK"), seguida de vuit capçaleres i finalment el cos. El cos de la resposta és la pàgina web HTML que visualitzarà el navegador.

Vegeu ara un exemple on el client demana al servidor quines són les ordres o mètodes que implementa:

```
root@server:~# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
OPTIONS /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 14 Sep 2020 08:01:06 GMT
Server: Apache/2.4.38 (Debian)
Allow: GET,POST,OPTIONS,HEAD
Content-Length: 0
Connection: close
Content-Type: text/html

Connection closed by foreign host.
```

```
root@server:~#
```

Finalment vegeu la simulació d'una petició POST. S'ha emplenat un formulari amb uns camps (nom, cognom1 i cognom2) i aquests valors es transfereixen per POST al servidor, segurament a un script tipus CGI, JavaScript o ASP.

```
root@server:~# telnet www.ioc.cat 80
Trying 10.0.0.2...
Connected to www.ioc.cat.
Escape character is '^]'.
POST /cgi-bin/script-06.sh HTTP/1.1
Host: www.ioc.cat
Content-Type: text/html
Content-Length: 33

nom=pere&cognom1=pou&cognom2=prat
HTTP/1.1 200 OK
Date: Mon, 14 Sep 2020 08:01:06 GMT
Server: Apache/2.4.38 (Debian)
Connection: close
Transfer-Encoding: chunked
Content-Type: text/html; charset=UTF-8
<h1> Llistat dels arguments rebuts</h1>
<h2> POST arguments rebuts per sdtin <h2>
nom=pere&cognom1=pou&cognom2=prat

Connection closed by foreign host.
```

1.2 Instal·lació i configuració de servidors web

El protocol HTTP està estructurat en forma de servei client/servidor. Per tant, cal disposar del programari apropiat per representar cada un d'aquests rols. El programari que fa la funció de client usualment ja està disponible en el sistema operatiu amb aplicacions com, per exemple, els navegadors gràfics Firefox i Chrome o navegadors d'entorn de text com Lynx. És a dir, per disposar de la part client del servei HTTP normalment no cal instal·lar res, perquè tots els sistemes operatius proporcionen almenys un navegador.

Així, quan parlem d'instal·lar el servei HTTP fem referència al procés d'instal·lació i configuració del programari del servidor. La instal·lació del programari que proporciona el servei HTTP es fa de manera molt similar a la d'altres serveis de xarxa (com els serveis DHCP, DNS o FTP). Es tracta d'instal·lar el programari de l'aplicació servidor i fer-ne la configuració apropiada. Senzill, oi?

Per fer això cal fer les reflexions i passos següents:

- Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?
- Obtenir l'aplicació que proporciona el servei HTTP.
- Observar l'estat de la xarxa actual. El servei està ja en funcionament? Existeix ja un servidor HTTP instal·lat i actiu?

- Instal·lar l'aplicació servidor.
- Comprovar que la instal·lació s'ha efectuat correctament.
- Configurar el servei en el servidor i comprovar que els clients hi poden accedir.
- Comprovar que el servei funciona correctament.

1.2.1 Aplicacions de servidor HTTP

Sempre que l'administrador vol posar en funcionament un nou servei de xarxa cal que primerament analitzi quines aplicacions hi ha al mercat que ofereixen aquest servei. És feina seva estudiar les característiques de les diverses aplicacions, com per exemple avaluar-ne l'eficiència, el cost, el que en diuen altres usuaris... Això es pot fer navegant per Internet, consultant les revistes especialitzades o demanant consell a un expert.

Usualment, però, l'administrador acaba utilitzant l'aplicació de servidor HTTP que li proporciona el mateix sistema operatiu. Si utilitzeu Windows, l'empresa Microsoft ofereix una aplicació pròpia, però també en podeu trobar d'altres a Internet. Igualment, si utilitzeu GNU/Linux, segurament la mateixa distribució ja proporciona un servidor HTTP o bé n'existeix algun de clàssic provinent d'Unix. De totes maneres en podeu obtenir d'altres a Internet.

Cerca d'HTTP a Internet

Usualment, l'administrador s'informa per mitjà del seu cercador preferit (per exemple, Google) i de webs com la Viquipèdia. Proveu de buscar "HTTP" o "HTTP server" en aquests serveis.

L'**Apache Server** és un programari de servidor HTTP omnipresent en tots els sistemes operatius avui en dia. Tot i que està basat en GNU/Linux, també és utilitzat pels sistemes operatius de Mac i Windows.

Podeu trobar tota la informació d'aquest servidor a **www.apache.org**.

1.2.2 Instal·lació de l'aplicació servidor

Els usuaris de GNU/Linux poden buscar fàcilment per Internet paquets de servidor HTTP usant eines com yum o apt-get i els repositoris de paquets apropiats segons la distribució que utilitzin. A més, sempre es pot recórrer als cercadors per localitzar el que faci falta.

Un cop instal·lat el programari caldrà identificar què s'ha instal·lat. Quins paquets i què contenen. A vegades no s'instal·laran paquets sinó fitxers .tar, el contingut dels quals també caldrà saber examinar. És important saber identificar quins dels components instal·lats corresponen a fitxers executables, quins a fitxers de configuració i quins a fitxers de documentació.

Tot servei instal·lat s'ha de configurar apropiadament i posar en marxa. Per tant, caldrà saber gestionar l'estat del servei (engegar, aturar, recarregar...) i definir l'estat que ha de tenir en els diferents *runlevels* (nivells d'execució) del sistema.

En definitiva, el procediment d'instal·lar inclourà usualment:

- Buscar el programari del servei (sigui en format de paquets .deb, .rpm o .tar) i descarregar-lo utilitzant l'eina apropiada segons quina sigui la distribució que utilitzem.
- Examinar el sistema per identificar quin programari, quins paquets, hi ha instal·lats relacionats amb el servei.
- Identificar els components del servei. Quins són els fitxers executables, quins els de configuració i quins els de documentació.
- Consultar i establir l'estat del servei (engegar i aturar) i saber establir l'estat per defecte per a cada *runlevel*.

1.2.3 Configuració per defecte

El servei HTTP té, en instal·lar-se, una configuració per defecte que acostuma a ser l'apropiada per a un servidor web bàsic. De vegades té els fitxers de configuració buits, de manera que caldrà editar-los abans de posar el servei en funcionament.

En qualsevol cas, cal saber identificar cadascun dels conceptes que es descriuen a continuació (es mostren els valors apropiats per al servidor Apache):

- Nom del servei: apache2, localitzat a /etc/init.d/apache2.
- Fitxer de configuració: /etc/apache2/apache2.conf.
- Directori de configuracions particulars de mòduls externs: /etc/apache2/mods-available i /etc/apache2/mods-enabled.
- Directori de treball del servidor: /etc/apache2.
- Directori de publicació del servidor: /var/www/html.
- Ubicació de fitxers d'exemple, documentació i pàgines de manual d'on poder obtenir una configuració inicial bàsica.

La configuració d'un servidor web pot ser molt senzilla o terriblement complexa, tot depèn dels objectius que ens proposem. Per publicar un senzill web estàtic no cal fer altra cosa que copiar els fitxers al directori indicat i utilitzar la configuració per defecte del servidor. Si volem utilitzar diverses seus webs virtuals amb certificats digitals per permetre connexions segures i amb contingut dinàmic, la configuració del servidor esdevé una mica més entretinguda.

La configuració del servidor web Apache s'estructura en:

- Secció 1: configuració global
- Secció 2: configuració de la seu web principal
- Secció 3: configuració de seus virtuals

Configuració global

En aquesta secció es descriuen aspectes generals del funcionament del servidor, entès com un servei (com un dimoni) del sistema. S'hi descriuen les característiques següents, entre d'altres:

- Definir l'arrel on hi ha els fitxers de configuració Apache.
- Localitzar i observar on es troba el fitxer del PID.
- Definir per quines adreces IP i ports escolta el servidor.
- Carregar els mòduls dinàmics.
- Definir l'usuari i grup amb el qual s'executa Apache.

Les principals directives del servidor de configuració global del servei són:

- **ServerRoot**: descriu el directori de treball del servei. Dins d'aquest directori és on hi ha els fitxers de configuració i on s'han generat enllaços simbòlics que permeten enllaçar amb els mòduls, el PID, els *logs* i els fitxers de configuració particulars de mòduls externs.

```
1 ServerRoot "/etc/apache2"
```

- **Include**: descriu el directori per defecte on hi ha més fitxers de configuració a incloure. En lloc de generar un fitxer de configuració molt gran, es crea un fitxer particular per a cada aspecte addicional que cal configurar. Generalment n'hi ha un per a cada mòdul extra que es carrega, per exemple per al SSL, l'LDAP...

```
1 Include conf-enabled/*.conf
```

- **Listen**: indica les adreces IP i els ports pels quals escolta el servidor. Per defecte, el servidor escolta pel port 80, corresponent al protocol HTTP, però també se sol fer pel port 8080, pel port 443 en comunicacions HTTPS i per qualsevol altre port diferent que es vulgui usar. Per defecte, escolta per totes les adreces IP del servidor.

```
Listen *:80
```

Es poden indicar plantilles per a les adreces IP i per als ports usant el caràcter *. Així, *10.0.0.1:** indica escoltar per qualsevol port per a la IP indicada. En canvi, **:8080* significa escoltar pel port 8080 per a totes les adreces IP del servidor. Es poden posar tantes directives *Listen* com facin falta.

```
Listen *:80              #escoltar pel port 80 per a totes les adreces IP
Listen 10.0.0.1:*        #escoltar per tots els ports per a aquesta IP
Listen 192.168.1.30:443  #escoltar pel port del protocol HTTPS per a l'adreça
    IP indicada
```

- **User** i **Group**: permeten definir l'usuari i el grup amb els quals s'executa el servidor. En aquest cas s'executa com a usuari Apache i grup Apache.

```
User apache
Group apache
```

Vegeu l'estructura dels directoris del servidor amb:

```
oot@server:~# tree /etc/apache2
/etc/apache2
|-- apache2.conf
|-- conf-available
|   |-- charset.conf
|   |-- javascript-common.conf
|   |-- localized-error-pages.conf
|   |-- other-vhosts-access-log.conf
|   |-- security.conf
|   '-- serve-cgi-bin.conf
|-- conf-enabled
|   |-- charset.conf -> ../conf-available/charset.conf
|   |-- localized-error-pages.conf -> ../conf-available/localized-error-pages.
    conf
|   |-- other-vhosts-access-log.conf -> ../conf-available/other-vhosts-access-
    log.conf
|   |-- security.conf -> ../conf-available/security.conf
|   '-- serve-cgi-bin.conf -> ../conf-available/serve-cgi-bin.conf
|-- envvars
|-- magic
|-- mods-available
|   |-- access_compat.load
|   |-- actions.conf
|   |-- actions.load
<retallat>
|-- mods-enabled
|   |-- access_compat.load -> ../mods-available/access_compat.load
|   |-- alias.conf -> ../mods-available/alias.conf
|   |-- alias.load -> ../mods-available/alias.load
|   |-- auth_basic.load -> ../mods-available/auth_basic.load
<retallat>
|-- ports.conf
|-- sites-available
|   |-- 000-default.conf
|   '-- default-ssl.conf
|-- sites-enabled
|   |-- 000-default.conf -> ../sites-available/000-default.conf
|   '-- default-ssl.conf -> ../sites-available/default-ssl.conf
'-- ssl
    |-- apache.crt
    '-- apache.key

```

```
7 directories, 196 files
root@server:~#
```

Configuració de la seu web principal

La segona secció del fitxer de configuració descriu les opcions de funcionament i de publicació de la seu web per defecte o principal del servidor. Les directives usades aquí afecten al web per defecte i s'hereten per a totes les altres seus web (virtuals) que es defineixin en el servidor. S'ha de tenir clar que el servidor web pot servir múltiples seus webs anomenades seus **virtuals** o *vhosts*. Existeix sempre una seu que és la seu web **principal** o per defecte, a part de les altres seus virtuals que es puguin definir.

El servidor web configura sempre almenys una seu web **principal** amb independència del fet que es configurin altres seus **virtuals**.
Les opcions definides en aquesta segona secció:

- Afecten a la seu web principal.
- Les hereten per defecte totes les altres seus webs virtuals.

S'hi descriuen les característiques següents, entre d'altres:

DocumentRoot: estableix el direcori de publicació de la seu web principal o per defecte. Els clients que es connectin a l'URL indicat per *ServerName* podran accedir al contingut de *DocumentRoot*. És dins d'aquest directori que hi haurà el típic fitxer index.html i la resta de fitxers i directoris que formen el web principal.

```
DocumentRoot "/var/www/html"
```

Directory: per a cada directori que calgui configurar es pot definir un bloc d'opcions de configuració agrupades en aquesta directva. Evidentment, les opcions afecten al directori i també s'hereten per als seus subdirectoris. Cal parar atenció en el fet que el directori a indicar és una ruta absoluta de l'arbre de directoris físic del sistema i no una ruta relativa lògica de l'estructura del web.

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Options +Includes
    XBitHack On
</Directory>
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
    XBitHack On
</Directory>
<IfModule mod_userdir.c>
    UserDir disabled
</IfModule>
```

DirectoryIndex: en l'exemple següent es pot veure com es defineixen els documents a mostrar per defecte quan se solicita un URL i no s'especifica el document.

```
DirectoryIndex index.html index.html.var
```

htaccess: a banda de les directives *Directory* es pot usar un altre mètode per establir opcions de configuració per a un directori determinat i per a tots els seus subdirectoris. Consisteix a posar un fitxer de configuració .htaccess a cada directori a configurar específicament. El fitxer conté les opcions específiques per al directori i els seus subdirectoris. Evidentment, aquest fitxer s'ha de protegir perquè no sigui descarregat pels clients.

```
AccessFileName .htaccess
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
```

mime: indica com s'identifiquen els tipus MIME.

```
TypesConfig /etc/mime.types
DefaultType text/plain
<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>
```

Logs: defineix el fitxer de registre o *logs*, el nivell dels *logs* o *loglevel* i el format en el qual s'hi han de desar les entrades.

```
ErrorLog logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
    combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog logs/access_log combined
```

cgi-bin: defineix el directori que conté els scripts executables CGI i n'especifica les opcions de funcionament.

```
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
```

server-status i **server-info**: activen i defineixen el funcionament del monitoritge integrat en el servidor web Apache. Permeten observar detalladament la configuració del servidor i el seu estat actual. Cal activar aquestes funcionalitats per a cada seu web de la qual es vulgui fer el seguiment.

```
<Location /server-status>
    SetHandler server-status
```

```
    Order deny,allow
    Deny from all
    Allow from www.ioc.cat portatil localhost
</Location>
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from www.ioc.cat portatil localhost
</Location>
```

Depenent de la versió d'Apache i distribució de Linux, ens hem d'assegurar que aquests mòduls estan habilitats. Es pot consultar i habilitar amb la comanda *a2enmod*.

```
root@server:/etc/apache2# a2enmod info
Enabling module info.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@server:/etc/apache2# a2enmod status
Module status already enabled
root@server:/etc/apache2#
```

Configuració de seus virtuals

Les seus virtuals o *virtualhosts* es descriuen àmpliament en l'apartat "Creació i configuració de llocs web virtuals".

En aquesta secció es descriuen les seus virtuals que ha d'atendre el servidor. Cal una entrada VirtualHost per a cada web a servir.

VirtualHost: descriu una seu web virtual indicant la seva adreça IP i port associats. Es defineix el nom del servei i el directori de publicació per a aquest servei.

```
<VirtualHost www.ioc.cat:80>
    ServerAdmin webmaster@ioc.cat
    DocumentRoot /var/www/html
    ServerName www.ioc.cat
    ErrorLog logs/www.ioc.cat-error_log
    CustomLog logs/www.ioc.cat-access_log common
</VirtualHost>
```

1.2.4 Exemple de configuració bàsica

Un cop instal·lat el servidor és molt fàcil posar en funcionament la seu web principal del servidor. Simplement cal:

- Establir el lligam o *bind* amb la directiva *Listen* per indicar les IP i ports per on servir. De fet, es pot deixar el valor per defecte "*:80" si es vol atendre per totes les IP.
- Indicar el nom del servei amb la directiva *ServerName*.
- Poblar el directori de publicació amb els continguts del web.

- Assegurar-se que la resolució de noms DNS identifica correctament el nom del servei amb alguna de les IP del servidor.

Per fer proves es pot usar la resolució de noms locals via /etc/hosts:

```
root@server:~# cat /etc/hosts
192.168.1.30  ioc www.ioc.cat
# ping www.ioc.cat
```

Configuració del servidor:

```
Listen *:80
ServerAdmin root@localhost
ServerName www.ioc.cat:80
UseCanonicalName Off
DocumentRoot "/var/www/html"
```

Arrencada del servei:

```
root@server:~# service apache2 start
root@server:~# service apache2 status•
 apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset:
        enabled)
   Active: active (running) since Tue 2020-09-15 08:48:33 CEST; 6s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 3711 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/
       SUCCESS)
 Main PID: 3715 (apache2)
    Tasks: 55 (limit: 1138)
   Memory: 10.0M
   CGroup: /system.slice/apache2.service
           |--3715 /usr/sbin/apache2 -k start
           |--3716 /usr/sbin/apache2 -k start
           '--3717 /usr/sbin/apache2 -k start

de set. 15 08:48:32 server.ioc.cat systemd[1]: Starting The Apache HTTP Server
     ...
de set. 15 08:48:33 server.ioc.cat apachectl[3711]: AH00557: apache2:
     apr_sockaddr_info_get() failed for
de set. 15 08:48:33 server.ioc.cat apachectl[3711]: AH00558: apache2: Could not
      reliably determine the se
de set. 15 08:48:33 server.ioc.cat systemd[1]: Started The Apache HTTP Server.
root@server:/etc/apache2#
```

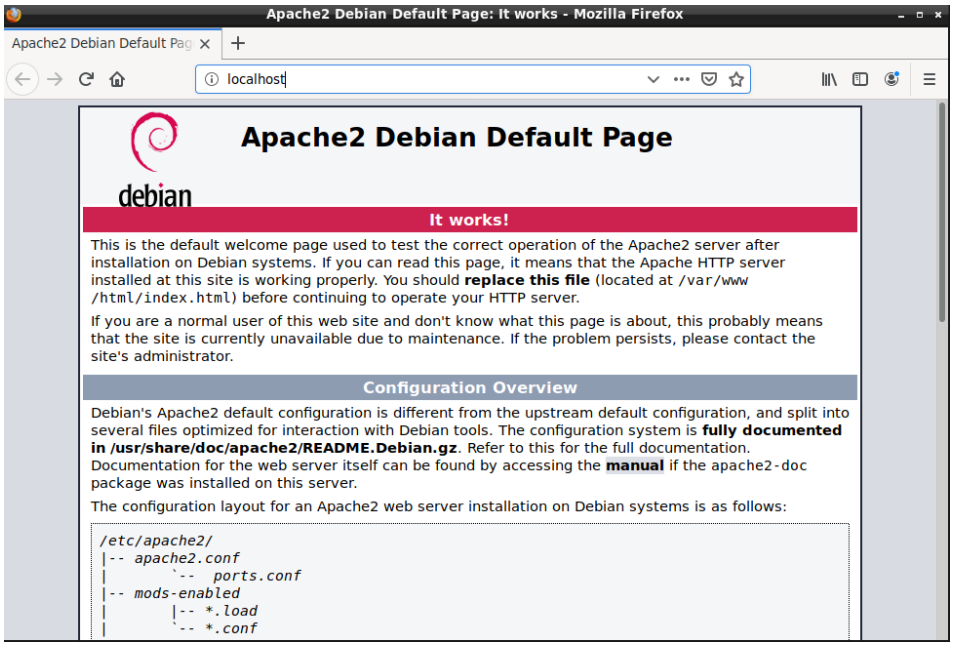
Comprovació del funcionament

Per validar el funcionament n'hi ha prou d'utilitzar qualsevol navegador client i connectar-se localment a qualsevol de les adreces IP del servidor o al *ServerName* definit per al servidor principal (l'únic configurat actualment).

```
[user@host]# telnet www.ioc.cat 80
```

Per defecte, el servidor web Apache mostra una pàgina preparada expressament per quan encara no hi ha contingut web en el directori de publicació. Aquesta pàgina serveix de test per verificar el funcionament del servidor, tal com es pot observar en la figura 1.1. Aquesta pàgina es mostra quan es contacta el servidor i encara no s'ha definit cap seu web pròpia.

FIGURA 1.1. Pàgina per defecte del servidor web Apache



Pàgina de prova pròpia

Finalment, es pot realitzar una pàgina HTML de prova pròpia per verificar que el servidor accedeix al directori de publicació i la mostra correctament. La pàgina ha de tenir el nom *index.html* o un dels noms definits com a nom de document per defecte.

El llistat de la pàgina i la seva ubicació:

```
[root@host]# cat /var/www/html/index.html
<html>
<head><title>Prova</title></head>
<body>
  <h1>Això és una prova de pàgina web</h1>
  <p>Aquí es pot escriure un paràgraf molt més interessant que aquest.<p>
</body>
</html>
```

1.3 Mòduls dinàmics

En fer la instal·lació del servidor s'han identificat els fitxers de configuració i l'executable del servei, httpd.conf i a l'httpd respectivament. Però aquests no són els únics fitxers de configuració i programari executable del servidor web. Sovint la funcionalitat del servidor web s'incrementa afegint-li noves funcions, com per exemple l'autenticació d'usuaris via PAM o LDAP, la incorporació de certificats digitals, comunicacions segures amb SSL... Cada una d'aquestes noves funcionalitats pot requerir programari addicional i noves directives de configuració.

Antigament, els fitxers de configuració creixien i creixien fins a "rebentar", cosa que dificulta la capacitat de l'administrador per governar-los i sobretot per tenir-los estructurats i fàcilment modificables. Avui en dia la majoria de serveis permeten estendre la seva funcionalitat en mòduls separats i amb fitxers de configuració que es mantenen a part i es carreguen mitjançant un *Include* en el fitxer de configuració principal.

Els **mòduls** permeten estendre la funcionalitat del servidor web proporcionant noves "peces" de programari.
La configuració del servei per mitjà de mòduls permet:

- Carregar peces de programari, mòduls encarregats de fer funcions específiques que extenen les funcionalitats del servidor web.
- Disposar de fitxers de configuració separats per a cada mòdul, cosa que facilita l'organització estructurada de la configuració.

S'han d'entendre els mòduls com un mecanisme de bocins de programari (a la manera del joc de construcció Lego) que es poden afegir i treure de la configuració actual per tal de seleccionar les prestacions i funcions que es volen proporcionar pel servidor. Podem dividir els mòduls en dues categories:

- **Estàtics**: el servidor web Apache que s'ha posat en funcionament ja té diversos mòduls carregats i executant-se des de bon principi. De fet, l'executable del servei, el dimoni httpd, s'ha compilat i se li han incorporat uns determinats mòduls (els responsables de fabricar el paquet per a la distribució que s'estigui utilitzant són qui els han seleccionat). Si es volgués disposar d'altres mòduls caldria compilar de nou l'executable del servidor.

```
# Llistat dels mòduls compilats
root@server:~# apache2 -l
Compiled in modules:
  core.c
  mod_so.c
  mod_watchdog.c
  http_core.c
  mod_log_config.c
  mod_logio.c
  mod_version.c
  mod_unixd.c
root@server:~#
```

- **Dinàmics**: a part dels mòduls estàtics que incorpora el servidor es poden afegir els mòduls dinàmics o **Dinamyc Shared Objects** que calguin. Des del fitxer de configuració global es poden afegir mòduls i també es poden afegir des del directori de configuracions específiques.

```
# Llistat de mòduls carregats: estàtics i dinàmics
root@server:~# source /etc/apache2/envvars
root@server:/root# apache2 -M
Loaded Modules:
```

```
 core_module (static)
 so_module (static)
 watchdog_module (static)
 http_module (static)
 log_config_module (static)
 logio_module (static)
 version_module (static)
 unixd_module (static)
 access_compat_module (shared)
 alias_module (shared)
 auth_basic_module (shared)
 authn_core_module (shared)
 authn_file_module (shared)
 authz_core_module (shared)
 authz_host_module (shared)
 authz_user_module (shared)
 autoindex_module (shared)
 deflate_module (shared)
 dir_module (shared)
 ...
```

És convenient saber usar les eines que proporciona el servei per interrogar-lo. Hem de ser capaços de:

- Identificar la versió del servidor.
- Identificar les opcions amb què s'ha compilat el servidor.
- Llistar els mòduls estàtics.
- Llistar els mòduls dinàmics.
- Llistar les directives actives.
- Monitorar tot el servei usant el recurs web propi *server-status*.

```
# Versió d'HTTP
root@server:~# apache2 -v
Server version: Apache/2.4.38 (Debian)
Server built:   2019-10-15T19:53:42

# Llistat de la versió de servidor i les opcions amb les quals s'ha compilat
root@server:~# apache2 -V
[Tue Sep 15 09:17:49.993943 2020] [core:warn] [pid 5272] AH00111: Config
    variable ${APACHE_RUN_DIR} is not defined
apache2: Syntax error on line 80 of /etc/apache2/apache2.conf:
    DefaultRuntimeDir must be a valid directory, absolute or relative to
    ServerRoot
Server version: Apache/2.4.38 (Debian)
Server built:   2019-10-15T19:53:42
Server's Module Magic Number: 20120211:84
Server loaded:  APR 1.6.5, APR-UTIL 1.6.1
Compiled using: APR 1.6.5, APR-UTIL 1.6.1
Architecture:   64-bit
Server MPM:
Server compiled with....
 -D APR_HAS_SENDFILE
 -D APR_HAS_MMAP
 -D APR_HAVE_IPV6 (IPv4-mapped addresses enabled)
 -D APR_USE_SYSVSEM_SERIALIZE
 -D APR_USE_PTHREAD_SERIALIZE
 -D SINGLE_LISTEN_UNSERIALIZED_ACCEPT
 -D APR_HAS_OTHER_CHILD
 -D AP_HAVE_RELIABLE_PIPED_LOGS
 -D DYNAMIC_MODULE_LIMIT=256
```

```
 -D HTTPD_ROOT="/etc/apache2"
 -D SUEXEC_BIN="/usr/lib/apache2/suexec"
 -D DEFAULT_PIDLOG="/var/run/apache2.pid"
 -D DEFAULT_SCOREBOARD="logs/apache_runtime_status"
 -D DEFAULT_ERRORLOG="logs/error_log"
 -D AP_TYPES_CONFIG_FILE="mime.types"
 -D SERVER_CONFIG_FILE="apache2.conf"

# Llistat de les directives
root@server:~# source /etc/apache2/envvars
root@server:/root# apache2 -L
<Directory (core.c)
  Container for directives affecting resources located in the specified
      directories
  Allowed in *.conf only outside <Directory>, <Files>, <Location>, or <If>
<Location (core.c)
  Container for directives affecting resources accessed through the specified
      URL paths
  Allowed in *.conf only outside <Directory>, <Files>, <Location>, or <If>
<VirtualHost (core.c)
  Container to map directives to a particular virtual host, takes one or more
      host addresses
  Allowed in *.conf only outside <Directory>, <Files>, <Location>, or <If>
<Files (core.c)
  Container for directives affecting files matching specified patterns
  Allowed in *.conf anywhere and in .htaccess
  when AllowOverride isn't None
...
```

1.3.1 Examinar els mòduls dinàmics

En instal·lar els paquets del servei s'han instal·lat tot de mòduls en el directori específic de mòduls del servei httpd. Usualment aquest directori és /usr/lib/apache2/modules. Es pot fer un llistat d'aquest directori per observar quins mòduls externs hi ha instal·lats en el sistema.

```
root@server:/# ls -l /usr/lib/apache2/modules | head -5
total 4040
-rw-r--r-- 1 root root  15742 Oct 15  2019 httpd.exp
-rw-r--r-- 1 root root  14384 Oct 15  2019 mod_access_compat.so
-rw-r--r-- 1 root root  14384 Oct 15  2019 mod_actions.so
-rw-r--r-- 1 root root  18480 Oct 15  2019 mod_alias.so
root@server:/#
```

Tots aquests mòduls estan carregats al servidor? No necessàriament. Estan instal·lats, però que estiguin actualment en funcionament en el servidor depèn de si s'han carregat o no des de la configuració del servidor. La directiva **LoadModule** permet carregar mòduls dinàmics des d'algun dels fitxers de configuració del servei.

```
LoadModule auth_basic_module modules/mod_auth_basic.so
```

Aquest és un extracte dels mòduls carregats en el fitxer de configuració principal apache2.conf. Podem observar, per exemple, que es carreguen els mòduls d'autenticació bàsica, *digest*, *file*, LDAP...

```
LoadModule auth_basic_module modules/mod_auth_basic.so
```

```
LoadModule auth_digest_module modules/mod_auth_digest.so
```

No tots els mòduls que es carreguen s'indiquen en el fitxer de configuració principal apache2.conf. Per facilitar l'administració del servei, el fitxer de configuració es pot repartir en petits fitxers que en configurin aspectes concrets. Dividir la configuració per funcionalitats separades és molt pràctic perquè li permet a l'administrador governar cada aspecte per separat i perquè evita que el fitxer de configuració principal esdevingui un fitxer massa extens per ser manipulat amb facilitat.

Tal com s'ha pogut observar en fer la instal·lació, existeixen dos directoris, **apache2/conf-available** i **apache2/conf-enabled**, que contenen els fitxers de configuració de mòduls i de funcionalitats que s'han segregat del fitxer de configuració principal, essent el darrer el de la configuració activa. No només té fitxers de configuració de mòduls, sinó que l'administrador també pot decidir segregar en fitxers a part aquells aspectes que vol governar per separat. Això li permet la flexibilitat d'incorporar-los o no a la configuració en execució simplement incloent-los o no.

Apache proporciona un sistema disponible/actiu (available/enabled) que permet tenir diferents configuracions preparades per a configuracions, mòduls i seus virtuals que permet activar i desactivar d'una manera més còmode (en comptes d'esborrar, reanomenar, etc.). al mateix fitxer de configuració principal està explicat com a comentari, i les respectives comandes per activar i desactivar:

```
# * Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/
#   directories contain particular configuration snippets which manage modules,
#   global configuration fragments, or virtual host configurations,
#   respectively.
#
#   They are activated by symlinking available configuration files from their
#   respective *-available/ counterparts. These should be managed by using our
#   helpers a2enmod/a2dismod, a2ensite/a2dissite and a2enconf/a2disconf. See
#   their respective man pages for detailed information.
```

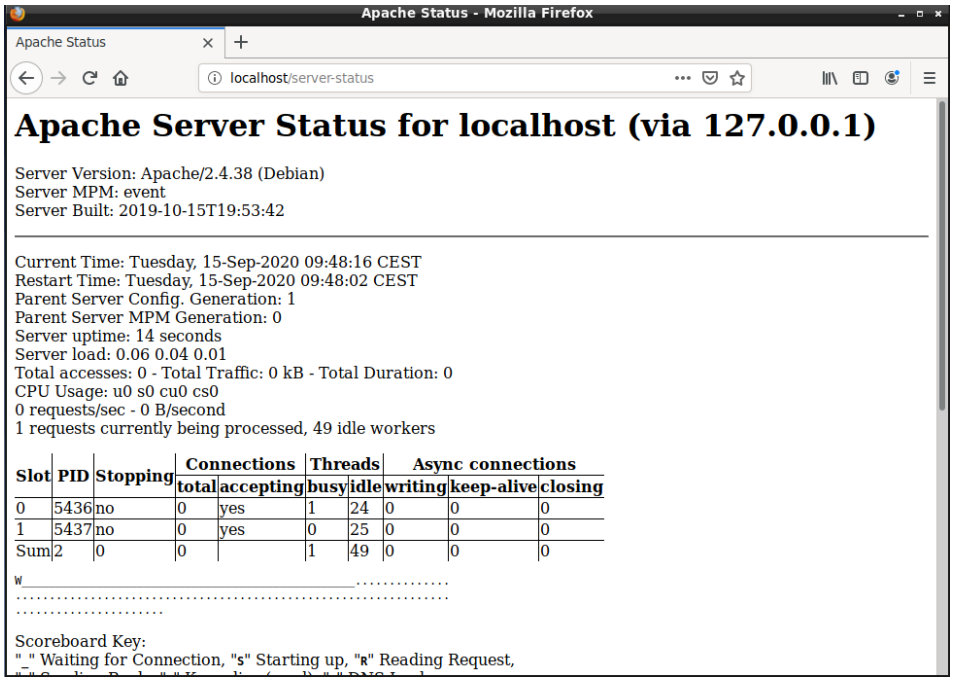
La directiva *Include* del fitxer de configuració global és l'encarregada de carregar tots els fitxers de configuració extres que hi ha al directori apache2/conf-enabled. En l'exemple ve amb la directiva *IncludeOptional* que fa el mateix que *Include*, però que si hi ha algun error en no trobar els fitxers de configuració, ho ignora.

```
# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf
```

En resum, podem dir que els mòduls **instal·lats** es troben en un directori específic tipus *usr/lib/apache2/modules*. Que estiguin instal·lats no significa que estiguin carregats i en funcionament.
Els mòduls es carreguen directament des del fitxer de configuració apache2.conf mitjançant directives **LoadModule**, indicant el nom del mòdul i la seva ubicació.
També es poden carregar des de fitxers de configuració específics, situats típicament en el directori apache2/conf-enabled. Els fitxers que conté poden configurar diversos aspectes de la funcionalitat del servidor i també poden, si cal, carregar mòduls usant la directiva *LoadModule*.

El millor mecanisme per observar els mòduls carregats i la configuració de les directives i opcions que proporcionen és consultar el mateix web de monitoratge que proporciona el servidor web a l'adreça ***server-status***, tal com podeu veure en la figura 1.2.

FIGURA 1.2. Pantalla d'informació de l'estat del servidor



1.4 Creació i configuració de llocs web virtuals

El servidor web s'ha configurat mitjançant el fitxer de configuració global per escoltar per un conjunt de ports i per a un conjunt d'adreces IP amb la directiva *Listen* i ha rebut un nom mitjançant la directiva *ServerName*. Aquest és el nom amb el qual s'identifica el servei principal o per defecte. La majoria de servidors en tenen prou amb una configuració com aquesta, ja que disposen d'una sola seu web. Ara bé, el servidor pot tenir més d'una seu web, ja sigui perquè té múltiples adreces IP o perquè té una adreça IP amb múltiples seus web.

En la terminologia d'Apache s'anomena ***virtual host*** o *vhost* a cada un dels servidors virtuals que hi ha en funcionament a banda del servidor principal o per defecte.

- Quan s'assignen servidors virtuals diferents a adreces IP diferents es parla de **servidors virtuals basats en IP** o *IP-Based vhosts*.
- Quan s'assignen múltiples seus virtuals a una mateixa adreça IP es parla de **servidors virtuals basats en nom** o *Name-Based vhosts*.

Per a cada seu virtual que es defineix cal utilitzar un bloc de configuració de la directiva *VirtualHost*:

```
<VirtualHost www.ioc.cat:80>
    ServerAdmin webmaster@www.ioc.cat
    DocumentRoot /var/www/www.ioc.cat
    ServerName ioc.cat
    ErrorLog logs/ioc.cat-error_log
    CustomLog logs/ioc.cat-access_log common
</VirtualHost>
```

Cal recordar que a part de les seus virtuals que es defineixin hi ha sempre una seu global o principal. Existeixen mecanismes per desactivar-la, però no els tractarem aquí.

Fem una anàlisi detallada de les principals opcions de configuració necessàries per definir una seu virtual:

- **VirtualHost**: aquesta directiva és la que fa el *bind*, el lligam amb l'adreça IP i port assignats a la seu virtual. Tot i que, per claredat, en l'exemple s'ha indicat un nom d'amfitrió (*host*) en lloc d'una adreça IP, Apache recomana usar sempre l'adreça IP.
- **ServerAdmin**: indica el nom de l'administrador de la seu virtual. De fet, n'indica el correu electrònic.
- **DocumentRoot**: defineix el directori de publicació de la seu web virtual. El directori que s'indica és una ruta absoluta del sistema físic de fitxers, no una ruta relativa del servidor web.
- **ServerName**: és el nom virtual amb el qual es reconeix aquest web, el nom que els clients han de referenciar per poder accedir al web.
- **errorLog** i **CustomLog**: aquestes dues directives especifiquen la ubicació dels fitxers de registre o *logs* de monitoratge de l'activitat d'aquesta seu web. Les rutes que s'hi indiquen són relatives i s'utilitza el directori de *logs* definit en la configuració global.

El problema dels fitxers de registre

En els exemples es pot observar que per a cada seu web es defineixen dos fitxers de registre (en poden ser tants com calgui). Si el servidor té en funcionament força seus

virtuals amb un trànsit de xarxa normal, pot succeir que de tants fitxers de *log* com té s'acabin esgotant els *file descriptors* del sistema.

El nombre de fitxers que pot tenir oberts un sistema és limitat. Si se supera, el sistema es bloqueja. És a dir, que realment no és difícil que això acabi provocant un problema.

Vegeu a l'apartat "Monitoratge del servei" com gestionar aquestes situacions.

L'exemple següent mostra un llistat de seus web virtuals d'un mateix servidor. Observeu el següent:

- La seu virtual www.ioc.cat està lligada a les adreces 11.0.0.3:80, 11.0.0.2:80, 11.0.0.1:80 i 10.0.0.1:80.
- La seu virtual www.inf.ioc.cat està lligada a l'adreça 10.0.0.2:80.
- L'adreça IP 10.0.0.3:80 conté diverses seus virtuals *Name-Based*. Concretament, www.virtual.cat i www.ioc-virtual.cat. La primera actua com a seu per defecte per a aquesta adreça IP.
- L'adreça IP 10.0.0.1:80 té també diverses seus virtuals *Name-Based*. Són www.ioc.cat i www.institut.cat. El servidor sempre utilitza la primera que s'ha definit en el fitxer de configuració com a seu per defecte de l'adreça IP.

```
root@server:/# apache2 -S
VirtualHost configuration:
11.0.0.3:80            www.ioc.cat (/etc/apache2/apache2.conf:1022)

10.0.0.2:80            www.inf.ioc.cat (/etc/apache2/apache2.conf:1050)

10.0.0.3:80            is a NameVirtualHost
         default server www.virtual.cat (/etc/apache2/apache2.conf:1067)
         port 80 namevhost www.virtual.cat (/etc/apache2/apache2.conf:1067)
         port 80 namevhost www.ioc-virtual.cat (/etc/apache2/apache2.conf:1075)

11.0.0.2:80            www.ioc.cat (/etc/apache2/apache2.conf:1022)

11.0.0.1:80            www.ioc.cat (/etc/apache2/apache2.conf:1022)

10.0.0.1:80            is a NameVirtualHost
         default server www.ioc.cat (/etc/apache2/apache2.conf:1022)
         port 80 namevhost www.ioc.cat (/etc/apache2/apache2.conf:1022)
         port 80 namevhost www.institut.cat (/etc/apache2/apache2.conf:1031)
ServerRoot: "/etc/apache2"
Main DocumentRoot: "/var/www/html"
Main ErrorLog: "/var/log/apache2/error.log"
Mutex default: dir="/var/run/apache2/" mechanism=default
Mutex watchdog-callback: using_defaults
PidFile: "/var/run/apache2/apache2.pid"
Define: DUMP_VHOSTS
Define: DUMP_RUN_CFG
User: name="www-data" id=33
Group: name="www-data" id=33
```

1.4.1 Seus virtuals basades en IP

El servidor web té la capacitat d'oferir serveis webs diferents a adreces IP diferents. De fet, pot oferir serveis diferents per a tantes combinacions IP:port com faci falta. Caldrà un bloc de configuració *VirtualHost* per a cada seu web. Si a cada una

d'aquestes diferents combinacions IP:port s'hi vol accedir amb un nom de seu web caldrà que la resolució DNS es faci apropiadament (globalment amb DNS o localment amb /etc/hosts).

Per definir servidors virtuals, *vhosts* en la terminologia Apache, basats en les adreces IP, cal usar la directiva:

```
<VirtualHost adreça_ip:port>
... configuració de la seu virtual ...
</VirtualHost>
```

adreça-IP: es recomana escriure l'adreça IP i no el nom de la seu web per indicar a quina adreça es lliga aquest *vhost*. També és vàlid escriure el nom de la seu, però això implica una doble resolució. Es pot usar el metacaràcter asterisc, (*), per indicar que s'escolta per totes les adreces IP, tot i que sembla un contrasentit, ja que precisament s'estan definint *IP-Based vhosts*. Usar l'asterisc pot tenir sentit si s'utilitza conjuntament amb ports diferents que permetin generar combinacions IP:port diferents.

port: indica el port associat al servidor virtual per l'adreça IP donada. També es pot usar el metacaràcter * per indicar qualsevol port. En aquest cas les diferents seus virtuals han de diferir d'adreça IP.

Per a cada seu virtual o **vhost** diferent que es vol implementar caldrà una directiva ***VirtualHost***.
La combinació **adreça-IP:port** permet establir l'associació de la seu virtual amb una combinació d'adreça IP més port. Es poden especificar múltiples associacions i es pot usar el metacaràcter *.

Es poden combinar múltiples expressions del tipus *adreça-IP:port* en cada sentència *VirtualHost*.

Alguns exemples de combinacions possibles són:

- **<VirtualHost 10.0.0.1:*>**: Seu virtual associada (*bind*) a qualsevol port de l'adreça 10.0.0.1.
- **<VirtualHost www.ioc.cat:*>**: Seu virtual associada a qualsevol port de l'adreça IP amb la qual es resolgui el nom www.ioc.cat.
- **<VirtualHost 10.0.0.2:80>**: Seu virtual associada exclusivament al port 80 de l'adreça IP 10.0.0.2.
- **<VirtualHost 10.0.0.2:443>**: Seu virtual associada al port 443 (el del protocol HTTPS) de l'adreça 10.0.0.2. Examinant aquest exemple i l'anterior es pot observar que es mostren seus virtuals diferents si se sol·licita l'adreça 10.0.0.2 via HTTP o via HTTPS.
- **<VirtualHost *:80>**: Seu virtual associada al port 80 de totes les adreces IP del servidor. És a dir, sigui quina sigui la IP, si és pel port 80 es mostrarà aquest *vhost*.

- **<VirtualHost *:443>**: El mateix que en l'exemple anterior, però en aquest cas associat exclusivament al port de l'HTTPS.
- **<VirtualHost *:*>**: Aquesta expressió no té sentit, ja que indica qualsevol port per a qualsevol IP. Bé, sí que té sentit, però no cal fer un amfitrió virtual per implementar aquest servei, es pot fer directament des del servei web principal.
- **<VirtualHost 10.0.0.3:80 10.0.0.3:8080 192.168.1.30:* 192.168.1.31:443>**: Estableix que aquesta seu web està associada als ports 80 i 8080 de l'adreça IP 10.0.0.3. També està lligada a qualsevol port de l'adreça IP 192.168.1.30, i finalment també està associada al port 443 de l'adreça IP 168.168.1.31.
- **<VirtualHost www.ioc.cat:80 www.ioc.cat:8080 www.xtec.cat:8080>**: Aquesta directiva lliga cada una de les adreces IP amb les quals es resolen els noms de seu web indicats i el seu port corresponent. Cal recordar que la documentació recomana usar les adreces IP en lloc dels noms d'amfitrió.

Exemple d'implementació (local) de seus virtuals basades en IP

Tot seguit implementarem tres seus virtuals "inventades" lligades a tres adreces falses en un servidor (per exemple el nostre mateix PC). Els passos a seguir són:

1. Crear les adreces IP falses. Per facilitar el monitoratge amb eines tipus Wireshark es faran les tres adreces al *loopback*.
2. Assignar noms d'amfitrió localment a cada adreça IP imitant noms de domini de seus web.
3. Crear i omplir de contingut els directoris de publicació de cada seu virtual.
4. Crear les entrades corresponents a cada *VirtualHost*.
5. Comprovar-ne el funcionament.

Creació les adreces IP falses al *loopback* i verificar-les:

```
# Creació les IP falses
root@server:/# ip address add 10.0.0.1/24 dev lo
root@server:/# ip address add 10.0.0.2/24 dev lo
root@server:/# ip address add 10.0.0.3/24 dev lo
```

Comprovem que s'han creat i que comuniquen:

```
root@server:/# ip address show lo
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
    default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet 10.0.0.1/24 scope global lo
       valid_lft forever preferred_lft forever
    inet 10.0.0.2/24 scope global secondary lo
       valid_lft forever preferred_lft forever
```

```
    inet 10.0.0.3/24 scope global secondary lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
root@server:/# ping 10.0.0.1 -c 2
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.059 ms

--- 10.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 0.019/0.039/0.059/0.020 ms
root@server:/# ping 10.0.0.2 -c 2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.020 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.061 ms

--- 10.0.0.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 21ms
rtt min/avg/max/mdev = 0.020/0.040/0.061/0.021 ms
root@server:/# ping 10.0.0.3 -c 2
PING 10.0.0.3 (10.0.0.3) 56(84) bytes of data.
64 bytes from 10.0.0.3: icmp_seq=1 ttl=64 time=0.021 ms
64 bytes from 10.0.0.3: icmp_seq=2 ttl=64 time=0.058 ms

--- 10.0.0.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 22ms
rtt min/avg/max/mdev = 0.021/0.039/0.058/0.019 ms
root@server:/#
```

Cal configurar la resolució de noms local per assignar noms de seu web (falsos) a cada una de les adreces IP creades. Els noms de seu que s'assignen són www.ioc.cat per a la primera adreça IP, www.virtual.cat per a la segona i www.secret.cat per a la tercera.

```
root@server:/# cat /etc/hosts
10.0.0.1  www.ioc.cat
10.0.0.2  www.virtual.org
10.0.0.3  www.secret.cat
root@server:/#
```

Evidentment, si es vol disposar de diverses seus virtuals és per publicar coses diferents a cada una. Cal crear els seus directoris de publicació i posar-hi contingut. Aquest contingut ha de ser **diferent** per poder observar fàcilment amb quina seu es contacta, quina seu mostra el servidor.

Els directoris de publicació tindran el mateix nom que la seu web i es trobaran dins del directori global WWW. El contingut de cada seu web pot ser la mateixa pàgina índex amb el títol modificat per mostrar el nom de la seu web.

```
# Creació dels tres directoris de publicació
root@server:/# mkdir /var/www/{www.ioc.cat,www.virtual.cat,www.secret.cat}

# Creació de la pàgina índex per a cada seu web. Per a la primera es pot fer:
root@server:/# cat /var/www/www.ioc.cat/index.html
<html>
    <head>
        <title>Seu virtual basada en IP</title>
    </head>
    <body>
        <h1>Seu virtual basada en IP</h1>
    </body>
</html>
```

```
root@server:/#
```

Un cop està tot a punt, cal fer la configuració apropiada en el servidor. S'han d'afegir les tres seus virtuals indicant la configuració de cada una. Un cop fet això caldrà reiniciar el servei (o recarregar la configuració). Aquest és l'aspecte del fitxer de configuració apache2.conf:

```
# Seu virtual ip-based "www.ioc.cat" port 80
<VirtualHost 10.0.0.1:80>
    ServerAdmin webmaster@host
    DocumentRoot /var/www/www.ioc.cat
    ServerName www.ioc.cat
</VirtualHost>

# Seu virtual ip-based "www.virtual.cat" qualsevol port
<VirtualHost 10.0.0.2:*>
    ServerAdmin webmaster@host
    DocumentRoot /var/www/www.virtual.cat
    ServerName www.virtual.org
</VirtualHost>

# Seu virtual ip-based "www.secret.cat"
<VirtualHost 10.0.0.3:443>
    ServerAdmin webmaster@host
    DocumentRoot /var/www/www.secret.cat
    ServerName www.secret.cat
    ... configuració SSL ...
</VirtualHost>
```

Per recarregar el servei sense aturar-lo cal fer:

```
root@server:/# service apache2 reload
```

Finalment, cal verificar el funcionament de les tres seus virtuals. Evidentment, el sistema més senzill és verificar des d'un navegador cada una de les seus web i observar que es mostra la pàgina inicial que s'ha definit per a cada seu. A continuació es mostra un altre mecanisme de verificació "en text", usant utilitats de comandes com Telnet per HTTP i Curl o OpenSSL per HTTPS:

```
# Verificar l'accés a la seu web virtual www.ioc.cat pel port 80
root@server:/# telnet 10.0.0.1 80
Trying 10.0.0.1...
Connected to 10.0.0.1.
Escape character is '^]'.
GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Tue, 15 Sep 2020 13:38:56 GMT
Server: Apache/2.4.38 (Debian)
Last-Modified: Tue, 15 Sep 2020 13:30:16 GMT
ETag: "97-5af5a26d14299"
Accept-Ranges: bytes
Content-Length: 151
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

<html>
    <head>
        <title>Seu virtual basada en IP</title>
    </head>
    <body>
        <h1>Seu virtual basada en IP</h1>
```

```
    </body>
</html>
Connection closed by foreign host.
root@server:/#
```

Es pot observar que el Telnet permet connectar al port 80 de l'adreça 10.0.0.1. La petició GET s'ha fet usant el protocol HTTP 1.0, de manera que no cal posar cap capçalera addicional per fer una petició de pàgina web.

En l'exemple següent es valida el funcionament de la segona seu virtual. Observeu que la petició GET s'ha fet utilitzant el protocol HTTP 1.1, que requereix obligatòriament la capçalera *Host: nomSeu*. Aquesta capçalera indica realment quina és la seu virtual a la qual es vol accedir:

```
root@server:/# telnet 10.0.0.2 80
Trying 10.0.0.2...
Connected to 10.0.0.2.
Escape character is '^]'.
GET / HTTP/1.1
host: www.virtual.cat

HTTP/1.1 200 OK
Date: Tue, 15 Sep 2020 13:38:59 GMT
Server: Apache/2.4.38 (Debian)
Last-Modified: Tue, 15 Sep 2020 13:30:46 GMT
ETag: "97-5af5a26d14299"
...
```

Finalment, cal verificar el funcionament de la tercera seu virtual, que s'ha configurat per usar connexions segures HTTPS via SSL. En l'exemple no s'han inclòs totes les directives necessàries ni la gestió dels certificats digitals per tal de poder generar una seu web segura. Tot això serà tractat més endavant. Les eines **OpenSSL** i **Curl** permeten comprovar-ne el funcionament en mode text:

```
root@server:/# openssl s_client -connect www.secret.cat:443 -state -debug
GET / HTTP/1.0
...
HTTP/1.1 200 OK
```

```
root@server:/# curl https://www.secret.cat -kv
...
HTTP/1.1 200 OK
```

1.4.2 Seus virtuals basades en nom

El servidor web pot oferir serveis webs diferents associats a una mateixa adreça IP. Això vol dir que una adreça IP pot tenir més d'una seu web associada. De fet, en pot tenir tantes com facin falta. Igual que passa amb les seus virtuals basades en nom també caldrà un bloc de configuració *VirtualHost* per a cada seu web a publicar. Si a cada una d'aquestes **seus virtuals basades en nom** s'hi vol accedir amb un nom de seu web caldrà que la resolució DNS es faci apropiadament (globalment amb DNS o localment amb /etc/hosts).

Per tant, cal que la petició HTTP tingui una capçalera *Host: nomSeu*, que determina quina de les seus associades es demana. Si la petició HTTP no conté aquesta capçalera, el servidor web es veu incapaç de determinar la seu virtual i contacta amb la seu per defecte associada a l'adreça IP:port (el primer dels *vhosts* definits per a cada ip:port és la seu per defecte).

El protocol **HTTP 1.0** no utilitza la capçalera *host*. Per tant, no permet diferenciar entre diverses seus virtuals d'una combinació IP:port. En aquest cas es contacta amb la seu per defecte.
El protocol **HTTP 1.1** requereix obligatòriament la capçalera **Host: nomSeu** per indicar a quina seu s'intenta accedir. Aquesta capçalera és la que determina a quin servei web s'accedirà. Aquest nom ha de coincidir amb un dels *ServerName* declarats.

Exemples d'implementació de seus virtuals basades en nom

Alguns exemples d'implementació són els següents:

- Diverses seus web basades en nom sobre una única adreça IP.

```
# Apache escolta al port 80
Listen 80

<VirtualHost *:80>
    DocumentRoot "/www/exemple1"
    ServerName www.exemple.com
    # Altres directives...
</VirtualHost>

<VirtualHost *:80>
    DocumentRoot "/www/exemple2"
    ServerName www.exemple.org
    # Altres directives...
</VirtualHost>
```

- Diverses seus web basades en nom sobre amb més d'una adreça IP.

```
Listen 80

# El servidor principal s'executa a 172.20.30.40
ServerName server.exemple.com
DocumentRoot "/www/mainserver"

<VirtualHost 172.20.30.50>
    DocumentRoot "/www/exemple1"
    ServerName www.exemple.com
    # Altres directives...
</VirtualHost>

<VirtualHost 172.20.30.50>
    DocumentRoot "/www/exemple2"
    ServerName www.exemple.org
    # Altres directives...
</VirtualHost>
```

- Diferents seus web en diferents ports.

```
Listen 80
Listen 8080

<VirtualHost 172.20.30.40:80>
    ServerName www.exemple.com
    DocumentRoot "/www/domini-80"
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
    ServerName www.exemple.com
    DocumentRoot "/www/domini-8080"
</VirtualHost>

<VirtualHost 172.20.30.40:80>
    ServerName www.exemple.org
    DocumentRoot "/www/domini-80"
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
    ServerName www.exemple.org
    DocumentRoot "/www/domini-8080"
</VirtualHost>
```

- Combinació de seus web i ports.

```
Listen 172.20.30.40:80
Listen 172.20.30.40:8080
Listen 172.20.30.50:80
Listen 172.20.30.50:8080

<VirtualHost 172.20.30.40:80>
    DocumentRoot "/www/exemple1-80"
    ServerName www.exemple.com
</VirtualHost>

<VirtualHost 172.20.30.40:8080>
    DocumentRoot "/www/exemple1-8080"
    ServerName www.exemple.com
</VirtualHost>

<VirtualHost 172.20.30.50:80>
    DocumentRoot "/www/exemple2-80"
    ServerName www.exemple.org
</VirtualHost>

<VirtualHost 172.20.30.50:8080>
    DocumentRoot "/www/exemple2-8080"
    ServerName www.exemple.org
</VirtualHost>
```

- Combinació de seus web basades en nom i en IP.

```
Listen 80

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/exemple1"
    ServerName www.exemple.com
</VirtualHost>

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/exemple2"
```

```
    ServerName www.exemple.org
</VirtualHost>

<VirtualHost 172.20.30.40>
    DocumentRoot "/www/exemple3"
    ServerName www.exemple.net
</VirtualHost>

# IP-based
<VirtualHost 172.20.30.50>
    DocumentRoot "/www/exemple4"
    ServerName www.exemple.edu
</VirtualHost>

<VirtualHost 172.20.30.60>
    DocumentRoot "/www/exemple5"
    ServerName www.exemple.gov
</VirtualHost>
```

Es poden trobar molts més exemples a la documentació oficial d'Apache.

1.5 Autenticació

Fins ara hem vist com crear i configurar diverses seus web accessibles per tothom que tingui accés al servidor. Hi ha ocasions en que es vol restringir l'accés a una part del web o a tot el web però només per a uns usuaris concrets. En aquest apartat es descriuen diverses formes de realitzar-ho.

El servei web incorpora mecanismes bàsics per verificar els usuaris que volen accedir a àrees restringides. Però a més a més la flexibilitat dels mòduls fa que es puguin afegir nous mecanismes que puguin sorgir més endavant tot i que no hàgin estat desenvolupats per Apache. Així, per validar l'accés a un directori amb material dels professors en un web d'una escola segurament n'hi ha prou amb el mecanisme bàsic de verificació d'usuaris i grups. En canvi, per accedir a un web ultrasecret d'una agència governamental potser cal incorporar mecanismes addicionals, basats per exemple en l'empremta òptica i el registre de veu.

Primerament cal analitzar els mecanismes de validació d'usuaris generals que permet el servidor web:

- **Autenticació bàsica amb fitxers**: el mecanisme més simple per implementar el control d'accés a recursos d'una seu web és utilitzar fitxers d'**usuaris** i **grups** propis del servidor. Apache proporciona eines per crear-los. L'avantatge principal d'aquest mètode és la facilitat d'administració. L'inconvenient és que comporta una gestió diferenciada dels usuaris del servei web i dels del sistema. De fet, això pot ser un inconvenient o un avantatge, si el que interessa és tenir-los segregats.
- **Autenticació mitjançant PAM**: en els sistemes GNU/Linux actuals l'autenticació dels usuaris es realitza via PAM (*Pluggable Authentication Module*). El PAM comprovarà el directori /etc/passwd, el LDAP, el Kerberos, les empremtes dactilars o el que calgui. Usar el lligam amb el mòdul del PAM

és un bon mecanisme per validar els usuaris del servei web igual que es validen els usuaris del sistema.

- **Autenticació mitjançant LDAP**: un dels mecanismes més populars actualment per a l'autenticació (i per a altres tasques) és el LDAP. Usar el mòdul del LDAP permet passar la validació dels usuaris a l'encarregat de gestionar l'autenticació LDAP dels usuaris del sistema. També es pot tenir en funcionament un servei LDAP específic per a les validacions del servei web.

Tot seguit es descriuen alguns conceptes clau relacionats amb el control d'accés al servidor:

- **Autenticació**: el procés d'autenticació és el que determina si un usuari és qui diu ser. En cap moment governa quins drets té, què pot fer i què no, simplement s'encarrega de comprovar que l'usuari és qui diu que és. Per implementar l'autenticació hi ha innumerables sistemes, des dels fitxers d'usuaris i contrasenyes fins a sofisticats mecanismes d'empremtes dactilars, òptiques, dades biomètriques o llapis USB (sense el llapis l'usuari no es pot identificar).
- **Autorització**: un cop s'ha identificat un usuari (és qui diu ser), què pot fer?, a quins recursos pot accedir?, a quins no? Això és l'autorització: determinar els drets d'utilització dels recursos.
- **Recurs amb accés restringit**: el control d'accés al servidor busca determinar quins recursos són accessibles per quins usuaris. Pot restringir l'accés a tota una seu web de manera que només els usuaris autoritzats puguin accedir als seus continguts. Sovint es restringeixen àrees concretes de la seu web, per exemple directoris que són accessibles només per un conjunt d'usuaris (els empleats, o els professors, en el web de l'escola). En aquest cas parlem de directoris amb accés restringit.
- **Reialme**: en una seu web hi poden haver diverses àrees restringides a perfils d'usuari diferents. Els reialmes permeten definir quines àrees restringides comparteixen el mateix grau d'accés. Tornem a l'exemple d'una seu web d'una escola on hi ha tot de continguts públics accessibles per tothom. El directori Notes és un recurs restringit on només hi poden accedir els alumnes de l'escola. Els directoris Programacions i Registres de Treball són accessibles només pels professors. Un professor que, per exemple, s'autentica per entrar a l'àrea Programacions introduïnt el seu identificador d'usuari i contrasenya, si vol entrar a l'àrea Registres de Treball s'hauria de tornar a identificar entrant de nou l'usuari i la contrasenya. Els reialmes permeten declarar que diversos llocs restringits tenen el mateix nivell d'accés, de manera que si un usuari s'ha autenticat en un està autenticat en tots els recursos que formen part del reialme.
- **Web amb inici de nom d'usuari/contrasenya**: un error molt típic és confondre l'autenticació a nivell de servidor amb l'autenticació a nivell de programari que realitzen les seus webs. Quan un usuari es valida en un

entorn web com per exemple Yahoo o Google, no està usant l'autenticació amb el servidor web. Està usant un usuari i una contrasenya de l'empresa web a la qual es connecta i la gestió d'aquesta sessió d'usuari per consultar el seu correu es realitza mitjançant la programació en les mateixes pàgines web que visita. Això **no** té res a veure amb el control d'accés al servidor que es tracta en aquest apartat.

Autenticar els usuaris és determinar de forma veraç si un usuari és qui diu ser. **Autoritzar** és indicar quins usuaris tenen dret a accedir a quins recursos. Les seus web i els directoris que limiten l'accés a un conjunt restringit d'usuaris s'anomenen **recursos restringits**. Els recursos restringits que implementen la mateixa política de seguretat es poden agrupar en **reialmes**.

1.5.1 Els mòduls de control d'accés

Mòduls

Es pot obtenir la llista de mòduls relacionats amb l'autenticació i el control d'accés consultant la pàgina corresponent de la documentació d'Apache.

Apache gestiona l'autenticació i el control d'accés al servidor mitjançant mòduls propis (a part que es poden incorporar mòduls externs). Cada mòdul consta d'un conjunt de directives que permeten configurar el funcionament de l'autenticació i control d'accés implementats. Aquests mòduls es poden classificar en tres categories segons la seva funcionalitat:

- **Tipus d'autenticació**: l'autenticació pot ser de tipus *basic* o *digest*. En aquests exemples s'utilitzarà autenticació bàsica. L'autenticació *digest* implica comunicacions xifrades. Aquests mòduls s'implementen amb la directiva ***AuthType***.

```
AuthType Basic
```

Podeu observar que els mòduls identifiquen en el seu nom la cadena **auth** d'*authentication*.

```
mod_auth_basic
mod_auth_digest
```

- **Proveïdor d'autenticació**: indica quin és el mecanisme usat per realitzar l'autenticació. Són els mòduls que permeten autenticar usant fitxers de contrasenyes o el mòdul PAM o el de LDAP... Es poden identificar els mòduls d'aquesta família perquè inclouen en el seu nom la cadena **authn** d'*authentication*.

```
mod_authn_file
mod_authn_alias
mod_authnz_ldap
...
```

- **Autorització**: els mòduls d'aquesta família proporcionen autorització a nivell d'usuaris, de grups, del LDAP o del que convingui. Aquests mòduls es determinen segons el valor que prengui la directiva ***Require***.

```
Require user valid-user
```

Podeu observar que els mòduls identifiquen en el seu nom la cadena **authz** d'*authorization*.

```
mod_authz_user
mod_authz_group
mod_authz_owner
mod_authz_ldap
...
```

1.5.2 Autenticació bàsica amb fitxers

El mecanisme més senzill per implementar l'autenticació en el servidor és l'autenticació bàsica amb fitxers d'usuaris i grups específics per al servidor web. Això es pot interpretar com un desavantatge perquè obliga a portar una gestió d'usuaris a més de la gestió d'usuaris del sistema. Però al mateix temps és un avantatge si el que volem és segregar aquets dos conjunts d'usuaris i administrar-los per separat.

Amb l'autenticació bàsica utilitzant fitxers es poden validar els usuaris utilitzant un **fitxer d'usuaris**, que conté els comptes d'usuaris i les seves contrasenyes.
També es poden validar grups d'usuaris amb un **fitxer de grups**, que indica quins usuaris formen part de cada grup.

El procés més simplificat per implementar la verificació d'usuaris i grups mitjançant fitxers de text pla amb contrasenyes requereix els passos següents:

1. Crear el fitxer d'usuaris en què s'indica la contrasenya corresponent a cada usuari.
2. Crear el fitxer de grups assignant a cada grup els usuaris que en formen part.
3. Identificar (o crear) el recurs que ha de tenir l'accés restringit.
4. Definir les directives apropiades per restringir l'accés al recurs als usuaris autoritzats.

L'exemple següent crearà un directori anomenat *privat* en la nostra web, al qual només hi podran accedir els usuaris autoritzats.

Primer cal crear el fitxer d'usuaris. Es tracta d'un fitxer de text pla en el qual s'emmagatzemen l'identificador i la contrasenya, que pot ser en text pla o xifrada,

de cada usuari. Per crear el fitxer i cada nou usuari s'utilitza l'ordre ***htpasswd***, proporcionada pel paquet del servidor. En el primer exemple s'utilitza l'opció *-c*, que crea el fitxer de nou.

```
root@server:/# htpasswd -c /var/www/passwd usuari
New password: usuari
Re-type new password: usuari
Adding password for user usuari
root@server:/# htpasswd /var/www/passwd usuari2
root@server:/# htpasswd /var/www/passwd usuari3
root@server:/# htpasswd /var/www/passwd usuari4
usuari:DeSaz54k9YRJU
usuari2:N0OF27Bcygc..
usuari3:okdHbmg.0G.Xo
...
```

A continuació cal posar en cada grup (de moment no n'hi ha cap) els usuaris que n'han de formar part. De fet, és tan senzill com crear un fitxer de text pla cada línia del qual consta del nom del grup, el delimitador dos punts (:) i la llista d'usuaris separats per espais.

```
root@server:/# vim /var/www/group
usuaris: usuari usuari2 usuari3 usuari4
```

Ara cal generar el **recurs restringit**, l'accés al qual només es permetrà als usuaris autoritzats. En aquest cas serà un directori anomenat *privat* a la nostra web.

```
root@server:/# mkdir /var/www/html/privat
root@server:/# vim /var/www/html/privat/index.html
... creació de la pàgina ....
```

Finalment s'assignen al directori local les directives apropiades per convertir-lo en un recurs d'accés restringit. Cal modificar el fitxer de configuració global apache2.conf (o el fitxer corresponent si es fa com a seu virtual) i definir un bloc de configuració usant la directiva ***Directory***. En aquesta directiva cal indicar la ruta absoluta corresponent al sistema de fitxers real del servidor (no es possible usar rutes relatives al servei web).

```
<Directori path-absolut-filesystem>
... opcions de configuració ...
</Directory>
```

Un exemple complet de configuració és el que es mostra a continuació, en el qual únicament es permet accedir al recurs a usuaris del grup anomenar *usuaris*:

```
<Directory /var/www/html/privat>
    AuthType Basic
    AuthName "Fitxers restringits"
    AuthBasicProvider file
    AuthUserFile /var/www/passwd
    AuthGroupFile /var/www/group
    Require group usuaris
</Directory>
```

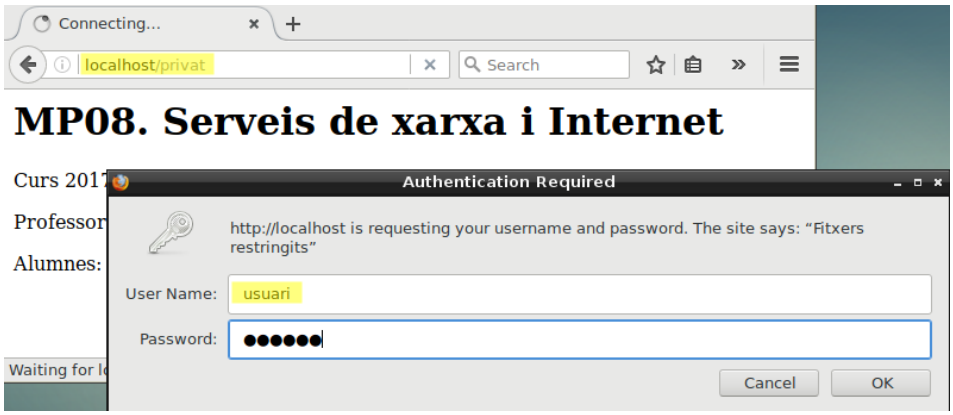
Vegeu les directives que s'utilitzen:

- ***AuthType***: indica que el tipus d'autenticació és bàsica (en lloc de *digest*).

- ***AythName***: declara el reialme al qual pertany el recurs restringit. Això permet que si hi ha altres recursos restringits associats a aquest reialme l'usuari que ja s'ha autenticat en un d'ells no ho hagi de fer en els altres. El nom del reialme el posa l'administrador web.
- ***AuthBasicProvider***: indica el mètode d'autenticació a usar. Pot prendre valors tipus *ldap*, *pam*, *dbm*, *bdb*, *file* i d'altres. El valor *file* significa que s'utilitzarà un fitxer d'usuaris i opcionalment un de grups.
- ***AuthUserFile***: indica quin és el fitxer que conté els comptes dels usuaris locals del servidor Apache. És el fitxer que s'ha creat en l'exemple anterior.
- ***AuthGroupFile***: indica quin és el fitxer de grups en el qual consta quins grups d'usuaris hi ha i quins usuaris pertanyen a cada grup.
- ***Require***: aquesta directiva és la que determina quina és la autorització a realitzar. En l'exemple es permet que qualsevol usuari del grup *usuaris* tingui accés al recurs.

Finalment, cal verificar que l'accés al directori local és concedit únicament als membres del grup profes. Evidentment el mecanisme més senzill és verificar des d'un navegador l'accés al recurs privat i observar que es demana l'autenticació. En la figura 1.3 es pot observar una petició d'autenticació d'usuari realitzada des d'un navegador Firefox.

Figura 1.3. Petició d'autenticació d'usuari



Exemples de mecanismes d'autorització

La directiva *Require* és la que defineix l'autorització d'accés al recurs, és a dir, qui pot accedir-hi. Aquests en són alguns exemples d'ús:

- *Require user valid-user*: permet l'accés a qualsevol usuari autenticat.
- *Require user usuari usuari2*: permet l'accés als usuaris indicats (usuari i usuari2).
- *Require group usuaris*: permet l'accés als usuaris que són membres d'algun dels grups indicats.

1.6 Comunicacions segures

El protocol HTTP pateix els mateixos problemes de seguretat que els seus companys dels inicis d'Internet (FTP, TFTP, SMTP...). Tota la informació viatja en text net i és fàcilment monitorable per altres. Quan un usuari es connecta a un web i indica l'usuari i la contrasenya, aquestes dades viatgen sense cap mena de protecció i qualsevol les pot capturar. Si el que es transmet són dades bancàries, llistes d'amistats íntimes o qualsevol tipus de dada privada, és desaconsellable fer-ho per HTTP.

El primer mecanisme de seguretat que es va implementar per a HTTP va ser el protocol SSL (Secure Socket Layer o capa de sòcol segur), desenvolupat per Netscape. L'SSL proporciona una capa entre la capa de transport TCP i la capa d'aplicació HTTP en què les dades viatgen xifrades. L'HTTPS solament és un esquema URI que indica la utilització d'HTML més algun mecanisme de transport xifrat, com SSL o TLS.

Quan s'utilitza HTTP amb un protocol xifrat com SSL o TCL s'anomena HTTPS (secure HTTP). Utilitza el port 443.

El protocol SSL es va enviar a l'IETF (Internet Engineering Task Force o Equip d'Enginyeria d'Internet, l'òrgan rector d'Internet) per a l'estandardització i després de diversos canvis va sorgir el protocol TLS (Transport Layer Security, Seguretat de capa de transport). El TLS proporciona les mateixes condicions de confidencialitat i autenticació en les transmissions HTTP que SSL.

L'HTTPS garanteix el trànsit de dades xifrat i el certificat del servidor. "En principi", autentica el web, però veurem que això depèn de si es confia o no amb el certificat usat.

Un dels avantatges de l'HTTPS és que permet la confidencialitat entre tots dos extrems de la comunicació encara que només sigui un dels extrems el que s'ha autenticat. Aquest model és molt pràctic quan, per exemple, un client anònim compra en un web autenticat. Quan es volen pagar els bitllets d'avió, interessa que les dades de la targeta de crèdit viatgin xifrades i que el receptor sigui la companyia aèria i no un web fals.

L'ús dels certificats no és exclusiu per autenticar el servidor. Si cal, els clients poden ser autenticats. Per exemple, un web pot requerir que els clients disposin del certificat que els atorga dret a accedir-hi (expedit, per exemple, per la mateixa entitat).

Els passos necessaris per implementar comunicacions segures que permeten a un navegador client (o a un client, sigui qui sigui) connectar-se via HTTPS a una seu web són:

- **Certificats digitals**: el servidor web ha de disposar d'una clau privada i d'un certificat digital.
- **Mòdul *mod_ssl***: cal tenir instal·lat el paquet de programari que proporciona les prestacions SSL al servidor i que la configuració activa en carregui els mòduls pertinents.

- **Configurar la seu web segura**: finalment cal establir les directives SSL apropiades a la seu web que es vol configurar per fer-la accessible via SSL.

L'objectiu de les explicacions següents és implementar connexions segures HTTPS a la seu web www.ioc.cat utilitzant SSL com a mecanisme de transport xifrat.

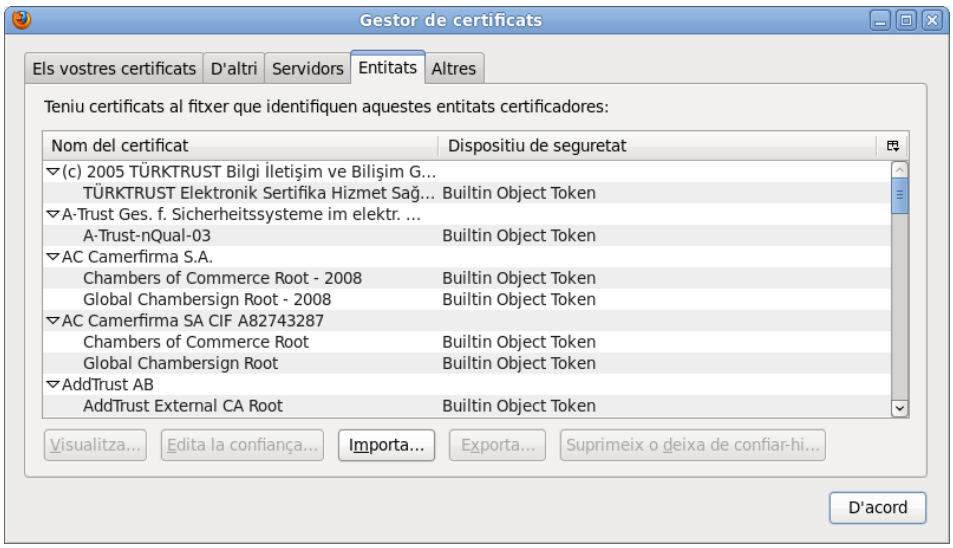
1.6.1 Els certificats del servidor

Suposarem que el servidor disposa ja d'una clau privada i d'un certificat, amb independència de com s'hagin obtingut. En concret, en el subdirectori certs del directori base del servei web hi ha:

- **server.crt**: el fitxer corresponent al certificat o clau pública del servidor. Aquest fitxer assegura als clients que es connecten a la seu web que el servidor és qui realment diu ser.
- **server.key**: és el fitxer amb la clau privada del servidor. Aquest fitxer s'ha codificat amb una *passphrase* o frase de pas de manera que cada vegada que s'inicialitzi el servidor web caldrà entrar aquesta frase.

La figura 1.4 mostra la pantalla típica de gestió de certificats de Firefox. En aquesta pantalla es poden llistar els certificats preinstal·lats en el navegador, observar-ne les seves propietats i també es poden afegir i eliminar certificats.

FIGURA 1.4. Pantalla de gestió de certificats de Firefox



Cal recordar que els navegadors clients validaran la confiança que els mereix el certificat contrastant el seu emissor amb la llista d'entitats certificadores que tenen carregada. Si l'emissor del certificat no hi és, caldrà fer passos per incorporar el certificat al navegador. Aquests passos poden ser:

- Admetre el certificat com a vàlid quan el navegador presenta "l'excepció de seguretat".
- Obtenir el certificat de l'entitat CA (Certification Authority o autoritat de certificació) que l'ha generat i incorporar l'entitat al llistat d'entitats en què el navegador confia.

Generar un certificat autosignat

A mode de recordatori ràpid, es pot generar una clau privada i un certificat autosignat fent:

```
# openssl req -new -x509 -nodes -out server.crt -keyout server.key
```

1.6.2 Configuració d'Apache per usar SSL

El servidor web podrà usar SSL si disposa dels mòduls que en proporcionen la capacitat. En cas de no tenir-los, cal buscar en els repositoris de programari habitual un paquet que proporcioni el mòdul apropiat, instal·lar-lo i examinar-ne el contingut. Usualment, tant el paquet com el mòdul que proporcionen les prestacions de trànsit segur SSL s'anomenen **mod_ssl**.

Per habilitar-lo a Apache:

```
root@server:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@server:~#
```

A la carpeta `/etc/apache2/mods-available` hi ha els diferents mòduls que es poden activar, entre els quals es troba el mòdul SSL. Es poden distingir dos fitxers, un que s'encarrega de la configuració i l'altre de la càrrega del mòdul:

```
root@server:/# ls /etc/apache2/mods-available/ssl.*
/etc/apache2/mods-available/ssl.conf  /etc/apache2/mods-available/ssl.load
root@server:/# cat /etc/apache2/mods-available/ssl.load
# Depends: setenvif mime socache_shmcb
LoadModule ssl_module /usr/lib/apache2/modules/mod_ssl.so
root@server:/# cat /etc/apache2/mods-available/ssl.conf
<IfModule mod_ssl.c>

  # Pseudo Random Number Generator (PRNG):
  # Configure one or more sources to seed the PRNG of the SSL library.
  # The seed data should be of good random quality.
  # WARNING! On some platforms /dev/random blocks if not enough entropy
  # is available. This means you then cannot use the /dev/random device
  # because it would lead to very long connection times (as long as
  # it requires to make more entropy available). But usually those
```

```
  # platforms additionally provide a /dev/urandom device which doesn't
  # block. So, if available, use this one instead. Read the mod_ssl User
  # Manual for more details.
  #
  SSLRandomSeed startup builtin
...
```

En executar la comanda *a2enmod*, hem fet que es crei un enllaç simbòlic des de la carpeta `/etc/apache2/mods-enabled` a la carpeta `/etc/apache2/mods-available` per a aquest mòdul. Aquest és el mecanisme que fa servir Apache per activar i desactivar els mòduls. En cas que es vulgui desactivar, cal fer servir la comanda *a2dismod*. També es poden fer servir sense paràmetres, i indicaran quins mòduls volem activar o desactivar dels que estan desactivats o activats respectivament:

```
root@server:/# a2enmod
Your choices are: access_compat actions alias allowmethods asis auth_basic
     auth_digest auth_form authn_anon authn_core authn_dbd authn_dbm authn_file
      authn_socache authnz_fcgi authnz_ldap authz_core authz_dbd authz_dbm
     authz_groupfile authz_host authz_owner authz_user autoindex brotli buffer
     cache cache_disk cache_socache cern_meta cgi cgid charset_lite data dav
     dav_fs dav_lock dbd deflate dialup dir dump_io echo env expires ext_filter
      file_cache filter headers heartbeat heartmonitor http2 ident imagemap
     include info lbmethod_bybusyness lbmethod_byrequests lbmethod_bytraffic
     lbmethod_heartbeat ldap log_debug log_forensic lua macro md mime
     mime_magic mpm_event mpm_prefork mpm_worker negotiation proxy proxy_ajp
     proxy_balancer proxy_connect proxy_express proxy_fcgi proxy_fdpass
     proxy_ftp proxy_hcheck proxy_html proxy_http proxy_http2 proxy_scgi
     proxy_uwsgi proxy_wstunnel ratelimit reflector remoteip reqtimeout request
      rewrite sed session session_cookie session_crypto session_dbd setenvif
     slotmem_plain slotmem_shm socache_dbm socache_memcache socache_shmcb
     speling ssl status substitute suexec unique_id userdir usertrack
     vhost_alias xml2enc
Which module(s) do you want to enable (wildcards ok)?
^C
root@server:/# a2dismod
Your choices are: access_compat alias auth_basic authn_core authn_file
     authz_core authz_host authz_user autoindex deflate dir env filter mime
     mpm_event negotiation reqtimeout setenvif status
Which module(s) do you want to disable (wildcards ok)?
^C
root@server:/#
```

1.6.3 Configuració de la seu web amb SSL

Cal aplicar a la nostra seu web les directives SSL apropiades per fer possible l'accés a aquesta seu web per HTTPS. El llistat de la directiva *VirtualHost* és:

```
<VirtualHost www.ioc.cat:443>
    ServerAdmin webmaster@ioc.cat
    DocumentRoot /var/www/www.ioc.cat
    SSLEngine On
    SSLProtocol all -SSLv3
    SSLCertificateKeyFile /var/www/certs/server.key
    SSLCertificateFile /var/www/certs/server.crt
</VirtualHost>
```

Vegeu les directives usades:

- **Port 443**: aquest és el port usual per a les connexions segures HTTP. Si la seu web només escolta per aquest port, només es podrà accedir al seu contingut per HTTPS. Si es volen seus diferents per al trànsit xifrat i per al no xifrat, n'hi ha prou de crear una altra seu virtual amb un altre port.
- ***SSLEngine On***: aquesta directiva indica que cal activar el trànsit SSL per a aquesta seu web.
- ***SSLProtocol all -SSLv3***: en aquesta directiva s'indiquen quins protocols es poden usar per generar el trànsit xifrat. Les opcions *all* i *-SSLv3* indiquen que s'accepten tots els protocols vàlids excepte el protocol SSLv3.
- ***SSLCertificateKeyFile* <clau privada del servidor>**: aquesta directiva indica el fitxer amb la clau privada del servidor.
- ***SSLCertificateFile* <certificat>**: indica el fitxer que conté el certificat del servidor. Aquest és el certificat que els navegadors clients veuran i del qual hauran de decidir si hi confien o no.
- ***SSLCACertificateFile* <certificat-CA>**: aquesta directiva és opcional i permet indicar quin és el fitxer que conté el certificat públic que ha emès l'entitat de certificació CA.

Un cop configurada apropiadament la seu virtual, cal reiniciar el servei. Des de qualsevol navegador s'ha de poder accedir a la seu usant HTTPS. Ara bé, es generarà una excepció de seguretat perquè el navegador desconeix la procedència del certificat. Si l'usuari accepta confiar en la seu web, el certificat s'incorporarà al navegador i accedirà de forma xifrada a la seu web.

No obstant, Apache porta ja una seu web predeterminada (en forma de plantilla) per activar la seu web amb SSL. Aquest fitxer és el `/etc/apache2/sites-available/default-ssl.conf`.

1.6.4 Verificació de les connexions SSL

Els problemes principals que es poden trobar els navegadors en connectar amb seus web amb certificats són els següents:

- Amb un certificat autosignat no cal definir cap CA. El navegador client mostrarà la típica pantalla d'excepció de seguretat i caldrà indicar que s'accepta el certificat de servidor per a la nostra entitat. És un certificat emès per la mateixa entitat.
- Amb un certificat generat per una CA local cal incorporar manualment el certificat al navegador. Un cop fet això el navegador serà capaç de validar el certificat del servidor amb la CA que l'ha expedit (*issuer*).

A més dels navegadors, existeixen eines d'entorn de text per verificar connexions SSL, de la mateixa manera que s'utilitza *telnet host 80* per verificar connexions

HTTP. D'una banda, es pot usar el mateix **OpenSSL** i, de l'altra, es pot instal·lar la utilitat **Curl**, que permet fer un ampli seguiment del diàleg SSL.

```
root@server:/# openssl s_client -connect www.ioc.org:cat
root@server:/# curl https://www.ioc.cat -kv
```

1.7 Monitoratge del servei

El servei web incorpora diverses eines per monitorar el funcionament del servei, algunes de configurades per defecte i d'altres que s'hi poden afegir. Les dues utilitats tractades en aquest apartat són *server-status* i *server-info*. Cal afegir el codi corresponent per tal d'indicar a quines seus web es vol fer el monitoratge per tal d'activar-los. A més a més, cal assegurar que aquests mòduls estan habilitats, ja que depenent de la versió d'Apache i distribució de Linux no sempre és així.

```
root@server:/etc/apache2# a2enmod info
Enabling module info.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@server:/etc/apache2# a2enmod status
Module status already enabled
root@server:/etc/apache2#
```

El fragment de codi següent mostra la configuració que permet monitorar les seus corresponents al nom de host *server* (la seu principal o per defecte) i la seu www.ioc.cat.

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from www.ioc.cat server
</Location>
<Location /server-info>
    SetHandler server-info
    Order deny,allow
    Deny from all
    Allow from www.ioc.cat server
</Location>
```

Per accedir als continguts del monitoratge simplement cal indicar des d'un navegador client els URL apropiats:

```
www.ioc.cat/server-info
www.ioc.cat/server-status
```

1.7.1 Utilitat de server-status

La informació bàsica que es mostra en consultar el *server-status* de la seu web www.ioc.cat és la següent:

```
Apache Server Status for www.ioc.cat (via 10.0.2.15)

Server Version: Apache/2.4.38 (Debian)
Server MPM: event
Server Built: 2019-10-15T19:53:42

Current Time: Monday, 21-Sep-2020 19:50:49 CEST
Restart Time: Monday, 21-Sep-2020 19:49:40 CEST
Parent Server Config. Generation: 1
Parent Server MPM Generation: 0
Server uptime: 1 minute 9 seconds
Server load: 0.00 0.03 0.06
Total accesses: 2 - Total Traffic: 14 kB - Total Duration: 15
CPU Usage: u0 s0 cu0 cs0
.029 requests/sec - 207 B/second - 7.0 kB/request - 7.5 ms/request
1 requests currently being processed, 49 idle workers

Slot  PID Stopping  Connections   Threads Async connections
total accepting busy  idle  writing keep-alive  closing
0 7866  no  1 yes 1 24  0 0 0
1 7867  no  0 yes 0 25  0 0 0
Sum 2 0 1   1 49  0 0 0

_____________W_____________________________.............
..........................................................
......................

Scoreboard Key:
"_" Waiting for Connection, "S" Starting up, "R" Reading Request,
"W" Sending Reply, "K" Keepalive (read), "D" DNS Lookup,
"C" Closing connection, "L" Logging, "G" Gracefully finishing,
"I" Idle cleanup of worker, "." Open slot with no current process
Srv PID Acc M CPU   SS  Req Dur Conn  Child Slot  Client  Protocol  VHost
    Request
0-0 7866  0/1/1 _   0.00  12  15  15  0.0 0.01  0.01  server  http/1.1  server.
    ioc.cat:80 GET /server-info HTTP/1.1
0-0 7866  0/1/1 _   0.00  12  0 0 0.0 0.00  0.00  server  http/1.1  server.ioc.
    cat:80 GET /favicon.ico HTTP/1.1
0-0 7866  0/0/0 W   0.00  0 0 0 0.0 0.00  0.00  10.0.2.15 http/1.1  server.ioc.
    cat:80 GET /server-status HTTP/1.1
Srv Child Server number - generation
PID OS process ID
Acc Number of accesses this connection / this child / this slot
M Mode of operation
CPU CPU usage, number of seconds
SS  Seconds since beginning of most recent request
Req Milliseconds required to process most recent request
Dur Sum of milliseconds required to process all requests
Conn  Kilobytes transferred this connection
Child Megabytes transferred this child
Slot  Total megabytes transferred this slot
Apache/2.4.38 (Debian) Server at www.ioc.cat Port 80
```

Els elements més destacats de la informació d'estatus són:

- Versió del servidor i data de compilació.
- Últims cops que s'ha engegat el servei i temps que fa que està actiu.
- Ús de CPU, detalls del trànsit i estadístiques sobre les peticions realitzades.

1.7.2 Utilitat de server-info

El servei de monitoratge *server-info* proporciona informació detallada de la configuració del dimoni del servidor, els valors obtinguts de processar cada un dels fitxers de configuració, els mòduls carregats i les configuracions de cada mòdul.

- **server settings**: descriu la versió del servidor i les opcions amb què s'ha compilat. En particular, podeu observar els valors que descriuen l'arrel de l'estructura de fitxers del servidor i el fitxer de configuració a usar.

```
Server Root: /etc/apache2
Config File: /etc/apache2/apache2.conf
```

- **configuration files**: descriu detalladament tots els valors de configuració, obtinguts dels diferents fitxers de configuració. Mostra el número de línia, la directiva i el valor que pren. Es pot observar que s'analitzen tots els fitxers actualment carregats en la configuració, el fitxer global apache2.conf i els inclosos en el directori `/etc/apache2`.

```
Configuration:
    In file: /etc/apache2/apache2.conf
      87: PidFile /var/run/apache2/apache2.pid
      92: Timeout 300
      98: KeepAlive On
     105: MaxKeepAliveRequests 100
     111: KeepAliveTimeout 5
     115: User www-data
     116: Group www-data
     126: HostnameLookups Off
     134: ErrorLog /var/log/apache2/error.log
     143: LogLevel warn
    In file: /etc/apache2/mods-enabled/alias.conf
      14: Alias /icons/ "/usr/share/apache2/icons/"
      16: <Directory "/usr/share/apache2/icons">
      17:   Options FollowSymlinks
      18:   AllowOverride None
      19:   Require all granted
        : </Directory>
    In file: /etc/apache2/mods-enabled/autoindex.conf
      ...
```

- **Llistat dels mòduls**: fa un llistat de tots els mòduls carregats actualment en la memòria. Es poden observar quins mòduls estan carregats estàticament i quins dinàmicament.

```
Server Module List
    core.c
    event.c
    http_core.c
...
```

- **Informació detallada de cada mòdul**: segurament aquesta és una de les opcions més interessants, perquè permet observar les directives proporcionades per un mòdul i els valors que té assignats. D'aquesta manera, es pot validar fàcilment si el mòdul adopta els valors apropiats o si s'ha comès alguna errada en la configuració.

1.8 Registres del servei

Els fitxers de *logs* enregistren tots els successos que es produeixen en el servei web i que s'ha declarat que s'han d'enregistrar. El seu funcionament és idèntic al dels *logs* del sistema. El servidor web Apache utilitza un fitxer que enregistra els errors que es produeixen i un altre que enregistra els accessos al servidor web. Aquests fitxers enregistren el servei web global o per defecte. Per a cada seu web virtual o *virtual host* es poden definir els fitxers de *log* que es considerin oportuns. El contingut que s'enregistra per a cada succés és configurable, de manera que l'administrador pot personalitzar al seu gust quina informació s'emmagatzema i en quin format.

> Els fitxers de ***log*** enregistren els successos del servidor web. S'anoten aquells successos que s'han **declarat** per ser enregistrats i en un **format** de missatge configurable.
> El servei web principal utilitza un fitxer d'**errors** i un de **accés** en la configuració predeterminada. Cada una de les diferents seus virtuals defineix els seus propis fitxers de *log*.

El següent és un recull de les directives relacionades amb la gestió de *logs* que hi ha al fitxer de configuració global apache2.conf:

```
1 ErrorLog ${APACHE_LOG_DIR}/error.log
2 LogLevel warn
3 LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\""
      vhost_combined
4 LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\""
      combined
5 LogFormat "%h %l %u %t \"%r\" %>s %O" common
6 LogFormat "%{Referer}i -> %U" referer
7 LogFormat "%{User-agent}i" agent
```

Els conceptes principals que descriuen aquestes directives són:

- ***ErrorLog***: defineix quin és el fitxer en què s'han d'enregistrar els successos d'error.
- ***LogLevel***: defineix quin és el nivell (*verbosity*) dels successos que s'han d'enregistrar. Aquest poden ser: emerg, alert, crit, error, warn, notice, info, debug, trace1... trace8. Estan ordenats de menys informació a més, doncs cal tenir en compte d'ajustar bé aquest paràmetre si no volem tenir fitxers

de *log* molt grans i que alenteixin el servei web, sobretot en entorns de producció (un cop ja testejats).

- ***CustomLog***: defineix el fitxer en què s'han de desar els accessos al servei web. Aquest fitxer contindrà el registre de totes les sol·licituds i respostes gestionades pel servidor. El format dels missatges de *log* que s'enregistraran per defecte és *combined*. Aquesta directiva es pot usar per definir tants fitxers de *log* com es cregui pertinent, cal una directiva per a cada fitxer a generar.
- **LogFormat**: cada una d'aquestes directives defineix un format de missatge de *log*. En l'exemple es defineixen cinc formats: *vhost_combined*, *combined*, *common*, *referer* i *agent*. Amb aquesta directiva l'administrador pot personalitzar el format que han de tenir els missatges de *log* al seu gust.
- **Directori de logs**: no es mostra explícitament en cap de les directives, però es pot veure que està definit en la variable d'entorn APACHE_LOG_DIR definida al fitxer `/etc/apache2/envvars`.

```
root@server:/# cat /etc/apache2/envvars | grep APACHE_LOG_DIR
export APACHE_LOG_DIR=/var/log/apache2$SUFFIX
root@server:/#
```

Aquesta, al seu torn, també està formada per una altra variable d'entorn (SUFFIX), que s'afegeix en el cas que s'hagi de tenir múltiples instàncies del servidor Apache:

```
# for supporting multiple apache2 instances
if [ "${APACHE_CONFDIR##/etc/apache2-}" != "${APACHE_CONFDIR}" ] ; then
        SUFFIX="-${APACHE_CONFDIR##/etc/apache2-}"
else
        SUFFIX=
fi
```

- **Seus virtuals**: cada una de les seus virtuals defineix els seus propis fitxers de *log*. És usual assignar a aquests fitxers el nom de la seu virtual, tal com es pot veure en l'exemple següent. Si una seu virtual no declara fitxers de *log* propis amb les directives *ErrorLog* i *CustomLog*, s'utilitzen els fitxers globals.

```
<VirtualHost www.ioc.cat:80>
    ServerAdmin webmaster@ioc.cat
    DocumentRoot /var/www/www.ioc.cat
    ServerName www.ioc.cat
    ErrorLog logs/www.ioc-error_log
    CustomLog logs/www.ioc-access_log common
</VirtualHost>
```

Així, seguint les indicacions estàndard, el directori de *logs* contindrà una parella de fitxers anomenats **error_log** i **access_log** per a cada seu virtual i per a la seu global. A més, altres mòduls, com per exemple SSL, poden generar els seus propis fitxers de *log*. També es mantenen històrics dels registres, de manera que es poden trobar múltiples versions del mateix fitxer corresponents a períodes de temps diferents (tasca duta a terme per *log-rotate*).

2. Instal·lació i administració de serveis de transferència de fitxers

Seguretat de la xarxa

En els orígens d'Internet, els usuaris eren part implicada i no els va passar pel cap que hi poguessin haver usuaris amb ànim d'"atacar" altres sistemes.

L'**FTP** (File Transfer Protocol o **protocol de transferència de fitxers**) és el protocol que proporciona el servei de transferència de fitxers més usat. Es basa en una arquitectura client/servidor i utilitza el protocol de transport TCP. Permet la transferència de fitxers de qualsevol tipus entre dos equips. El servidor actua a mode de repositori de fitxers i el client s'hi connecta per baixar (*download*) o pujar (*upload*) fitxers.

El **protocol FTP** (File Transfer Protocol o protocol de transferència de fitxers) és un protocol TCP que permet la transferència de fitxers en ambdós sentits entre un client i un servidor. Aquesta transferència és independent de la plataforma usada i del tipus de fitxers manejats.

La necessitat d'un mecanisme de transferència de fitxers sorgeix des de bon principi a Internet i el 1980 ja apareix la primera especificació de l'FTP. L'octubre de 1985 es publica l'RFC 959, que és la base del protocol actual. A aquest RFC se n'hi han afegit posteriorment d'altres per incorporar seguretat, internacionalització... Tractant-se d'un protocol tan "vell", no és estrany que sigui insegur. De fet, la majoria de protocols originaris d'Internet ho són (HTTP, SMTP...).

Una de les virtuts del model FTP és que permet transferir fitxers (també modificar-los, esborrar-los, afegir-los...) independentment de la plataforma i del sistema on resideixen. És a dir, l'FTP amaga els detalls d'implementació. Un client (que funcioni amb sistema operatiu Unix, per exemple) baixa un fitxer de text d'un servidor FTP sense saber el tipus de màquina ni el tipus de sistema de fitxers del servidor (Windows, per exemple).

Els objectius del servei FTP són els següents:

- Compartir fitxers tant de dades com de programes.
- Permetre prosseguir les transferències de fitxers un cop interrompudes (reprendre les baixades).
- Transferir dades de manera eficient i fiable.

La debilitat principal del protocol FTP és la falta de seguretat. Tot el flux de dades viatja en text pla, sense encriptar, fins i tot els noms d'usuari i les contrasenyes. Això ha obligat a adoptar noves estratègies per proporcionar confidencialitat a les transmissions.

2.1 Servei de transferència de fitxers

El servei FTP es basa en l'arquitectura client/servidor, de manera que caldrà descriure'l en cada un d'ells. En aquest apartat es descriuen els diferents modes d'operació d'un servidor FTP, les seves funcionalitats i els tipus d'accés permesos. Els servidors poden oferir els seus serveis a tota la comunitat d'Internet o a un entorn restringit, com per exemple una xarxa local. En el primer cas es parla d'un servidor d'accés públic global, en el segon cas, d'un servidor local o corporatiu. El servei pot ser ofert a qualsevol usuari, incloent usuaris anònims, o es pot restringir a usuaris i grups determinats.

El servidor FTP es pot classificar segons sigui:

- D'accés **públic /** d'accés **corporatiu**
- D'accés amb usuari **identificat /** d'accés amb usuari **anònim**
- De mode de transferència **ASCII /** de mode de transferència **binari**

De forma inusual, el servei FTP utilitza dos ports del sistema. A través del port 21 es realitza la interpretació de les instruccions. El port 20 és destinat a la transferència de dades, tot i que es pot utilitzar un altre port dinàmic en el seu lloc. El mode real de funcionament de la transmissió de dades i els ports implicats depenen del mode de funcionament, que pot ser actiu o passiu.

El mode de funcionament de la transferència FTP pot ser actiu o passiu. La funcionalitat del servei es classifica en:

- Mode intèrpret del protocol.
- Mode de transferència de dades.

2.1.1 Tipus de clients i servidors

El **servidor FTP** és una màquina que executa un programari determinat que proporciona el servei FTP a clients FTP. Usualment, en entorns GNU/Linux aquest programa és un dimoni, anomenat usualment *ftpd* o similar. En funció del tipus d'usuaris que permet connectar i de l'àmbit d'accés que permet, el podem classificar de maneres diferents.

Segons el tipus de clients que accepta, podem classificar els servidors FTP de la manera següent:

- **Usuari identificat**. El servidor requereix un nom d'usuari i una contrasenya vàlids per accedir al servei. Els comptes d'usuari poden ser gestionats directament per l'aplicació del servidor o se'n pot delegar l'autenticació al sistema operatiu.
- **Accés anònim**. Un servidor que permet accessos anònims permet que qualsevol usuari pugui accedir al repositori de fitxers. Usualment cal indicar com a nom d'usuari "anonymous" i com a contrasenya s'accepta qualsevol text, però per convenció s'escriu el correu electrònic de l'usuari.

Alguns servidors FTP permeten l'accés anònim acceptant qualsevol nom d'usuari i qualsevol contrasenya o fins i tot sense contrasenya.

Segons l'àmbit del servei que proporciona, podem classificar els servidors FTP de la manera següent:

- **Servidor públic**. Molts servidors FTP a Internet ofereixen servei d'accés anònim a mode de repositoris de programari perquè els usuaris el puguin utilitzar. N'hi ha que actuen com a rèpliques (miralls, *mirrors*) d'altres repositoris per tal d'apropar les baixades a l'usuari. Aquest servei és usualment només de lectura (pel client) i en sistemes GNU/Linux s'ubica sovint en els directoris /var/ftp o /var/ftp/pub.
- **Servidor corporatiu**. No cal oferir per força els serveis FTP a Internet; l'administrador de xarxa pot configurar el servidor FTP per oferir els serveis als equips que cregui oportuns. Dins d'una xarxa corporativa es pot disposar d'un o més servidors FTP que permeten l'accés als usuaris de la xarxa (tant a usuaris identificats com a usuaris anònims de la xarxa corporativa).

Exemple de mirall

Si us voleu baixar un GNU/Linux Live, en lloc de contactar el servidor de Fedora o Ubuntu, ho podeu fer en un mirall de RedIRIS, que és més proper.

Evidentment, el servidor FTP pot combinar els models anteriors i proporcionar accés tant a usuaris identificats com a usuaris anònims, i pot diferenciar els recursos que ofereix en funció de si són usuaris interns de la xarxa corporativa o d'Internet.

El **client FTP** és el programari que s'utilitza per establir una connexió amb el servidor per tal de poder baixar o pujar fitxers al servidor. El client pot llistar, modificar i afegir fitxers al servidor, a més de realitzar altres accions, sempre que hi estigui autoritzat. L'aplicació client pot ser basada en text o d'entorn gràfic, però en qualsevol cas ha de poder establir connexió amb el servidor.

URL

Acrònim d'*Uniform Resource Locator* (en català, localitzador uniforme de recursos). Sovint és usat com a sinònim d'URI (*Uniform Resource Identifier*, identificador uniforme de recursos), tot i que no és el mateix.

La connexió FTP es pot indicar mitjançant un URL del tipus ftp://servidor/fitxer, on *fitxer* pot ser una trajectòria. De fet, l'URL pot ser més detallat:

```
1 ftp://usuari:contrasenya@servidor:port/fitxer
```

En aquest format s'indica l'usuari i la contrasenya, el servidor, el port d'accés i el fitxer al qual es vol accedir. En determinats sistemes operatius es pot usar aquesta sintaxi en la línia d'ordres per indicar un fitxer remot, igual que es faria per indicar un fitxer local.

El **servidor** proporciona un repositori de fitxers i una aplicació que permet que els **clients** s'hi connectin i facin ús dels fitxers (baixar-los o pujar-ne). L'URL per accedir a un fitxer per FTP es pot expressar així: **ftp://usuari:contrasenya@servidor:port/fitxer**.

2.1.2 Funcionament del servei FTP

El port 21 és el port del canal de control. El port 20 és el del canal de dades.

L'FTP és un protocol d'aplicació basat en TCP/IP que utilitza TCP com a capa de transport. El servidor escolta connexions entrants de clients pel port 21 i inicia una sessió si l'autenticació s'estableix correctament. El servidor pot funcionar com un servei per si mateix (*stand-alone*) o pot estar configurat per funcionar dins d'un superservei de xarxa com, per exemple, inetd o xinetd. Si funciona en mode de servei propi és el servidor qui escolta les connexions entrants i les atén. Si s'executa dins del superservei de xarxa, aquest és qui detecta les connexions entrants i activa el dimoni de l'FTP perquè les atengui. Un cop ateses, el dimoni de l'FTP acaba i torna a ser el superservidor de xarxa el que es queda escoltant.

Tipus de client

- Identificat: té accés al sistema de fitxers complet.
- Anònim: és engabiat en un punt de l'arbre de fitxers.

Serveis autònom i xinetd

En el servei autònom (*stand-alone*) el servidor escolta per si mateix les connexions entrants. En el servei xinetd o inetd, el servidor està dins d'un superservei de xarxa. inetd és un superdimoni de xarxa que escolta connexions entrants de diferents protocols i executa el dimoni del servei corresponent en rebre una connexió. xinetd n'és la versió millorada.

L'accés als recursos del servidor varia usualment en funció de si el client és anònim o està identificat. Els clients identificats poden navegar per l'estructura de fitxers segons els seus permisos. Els clients anònims usualment estan engabiats (*chroot*) en una part de l'arbre de fitxers i no en poden sortir. Usualment, el servidor fa correspondre el directori de publicació (/var/ftp o /srv/ftp en sistemes GNU/Linux) amb el directori arrel (/) d'accés del client. El client pot descendir a partir d'aquest punt, però no pot anar a directoris superiors.

Chroot

En sistemes GNU/Linux, *chroot* és una utilitat o una tècnica consistent a "engabiar" serveis en una part del disc com si fos el disc sencer, de manera que es fa correspondre un directori particular (on hi ha el servei) a una arrel de disc virtual. Els usuaris del servei creuen que naveguen per tot el disc, però en realitat estan "engabiats" en una estructura virtual.

ASCII

Acrònim d'American Standard Code for Information Interchange. En català, codi estàndard americà per a l'intercanvi d'informació.

Els modes en què es transfereixen els fitxers entre el client i el servidor poden ser múltiples. Els dos més importants són aquests:

- **ASCII**. El fitxer es transmet caràcter a caràcter. Els caràcters han de correspondre als caràcters del codi bàsic ASCII. Si el fitxer conté caràcters ASCII no vàlids, la transferència fallarà. Per tant, es tracta d'un mode vàlid únicament per transferir text net. El receptor farà les conversions de caràcter necessàries per desar les dades en el format que requereixi.

Format del salt de línia

Penseu en el típic problema del salt de línia que varia segons el sistema operatiu. El servidor, per exemple, envia text usant el caràcter LF com a salt de línia (usa Unix) i el receptor els desa usant la combinació de caràcters CR+LF (que és el format usat pel seu sistema operatiu diferent d'Unix).

- **Binari** (*binary*). Quan el mode de transferència és binari, el fitxer s'envia bit a bit sense interpretació de cap mena. És el mode que cal usar per transmetre programes, imatges, vídeo, so, dades binàries...

2.1.3 Especificació del protocol FTP

El protocol FTP és un protocol de capa d'aplicació basat en TCP com a capa de transport. Utilitza el port 21 per al canal de control i el port 20 per al canal de dades. És un dels pocs protocols actuals que encara utilitzen més d'un port per a la comunicació. El port 21 s'utilitza com a canal de comunicació entre client i servidor. És per on es transmeten les ordres, però no els fitxers. Aquests es transmeten per una connexió diferent, per un canal diferent, que en principi utilitza el port 20 del servidor.

Tant en el client com en el servidor hi ha dues entitats clarament diferenciades:

- **Intèrpret del protocol**: és l'encarregat de l'intercanvi d'ordres i respostes entre client i servidor. Utilitza el canal de control establert entre el port de sortida del client i el port 21, on escolta el servidor. És l'encarregat d'interpretar les ordres de l'aplicació client convertint-les en instruccions FTP, executar-les en el servidor i retornar les respostes al client. No s'encarrega de la transferència de fitxers.
- **Transferència de dades**: és la part encarregada d'intercanviar els fitxers i directoris entre client i servidor. En el funcionament bàsic, el canal de dades s'estableix entre un nou port del client (port dinàmic i específic per a la transmissió del fitxer) i el port 20 (*ftp-data*) del servidor.

Modes de connexió FTP

- Actiu: el servidor usa el port 20. Correspon a l'ordre PORT del protocol.
- Passiu: el servidor usa un port dinàmic (no el 20). Correspon a l'ordre PASV del protocol.

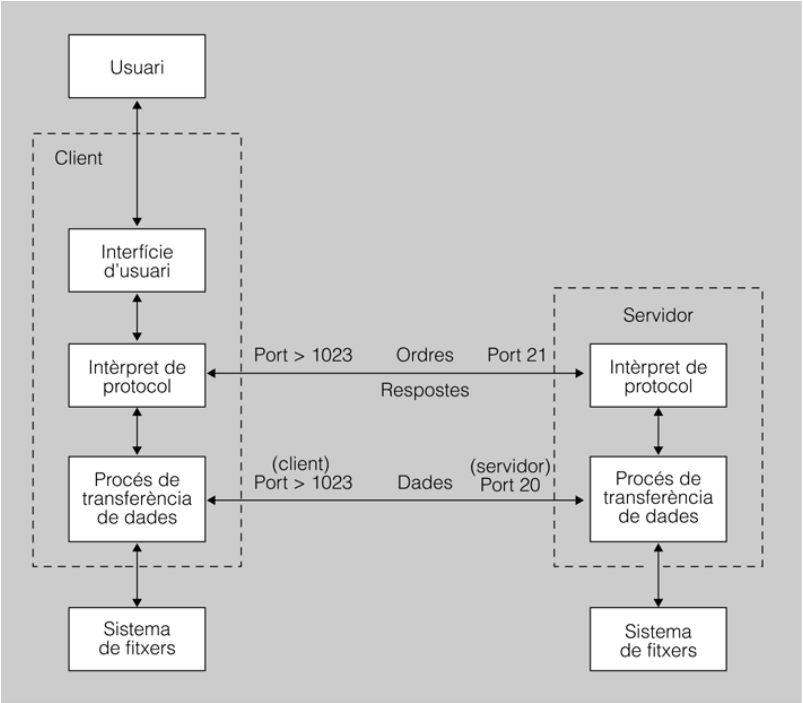
La connexió TCP del canal de dades entre client i servidor es pot establir de dues maneres diferents:

- **Mode actiu**: generalment és el mode per defecte. Abans de fer una sol·licitud al servidor que impliqui transferir dades pel canal de dades, el client indica al servidor el port dinàmic que utilitzarà. Per tant, el canal de dades s'estableix entre aquest port dinàmic del client i el port 20 del servidor. Servidor i client estableixen una nova connexió TCP per aquest canal.
- **Mode passiu**: el client fa una sol·licitud de mode passiu al servidor. Aquest respon enviant el seu port dinàmic, per on s'establirà el canal de dades (en lloc del port 20). Llavors el client inicia una nova connexió TCP entre un port dinàmic nou seu i el port dinàmic del servidor. Aquest és el canal de dades.

En resum, podríem dir que el client es posa en contacte amb el servidor connectant-se al port 21. Mitjançant aquest canal de comunicació, client i servidor governen tota la sessió FTP. Les transferències de fitxers es realitzen per una altra connexió

que es pot crear i destruir al llarg de la sessió (per exemple, per a cada baixada o segons els períodes d'inactivitat). En la figura 2.1 es pot veure l'esquema de connexió i de ports entre un client i un servidor FTP.

FIGURA 2.1. Model funcional del protocol FTP



Ordres i respostes FTP

En el document RFC 959 hi ha la llista d'ordres i la taula de codis/missatges de resposta del protocol FTP. Aquest document és l'especificació de l'estàndard FTP.

El protocol FTP descriu diferents categories d'ordres que el client pot realitzar (no les confongueu amb les ordres concretes de l'aplicació client). Aquestes s'agrupen en tres grups:

1. **Ordres de control d'accés**: les que gestionen l'accés al servei FTP. Per exemple, inici i finalització de la sessió, validació de l'usuari i la contrasenya, canvis de directori i de sistemes de fitxers...
2. **Ordres de paràmetres de transferència**: gestionen les opcions relacionades amb la transferència de fitxers com, per exemple, el mode de transferència binari o ASCII, els ports, el tipus de mode passiu o actiu...
3. **Ordres de servei FTP**: són les ordres d'allò que es vol fer en una sessió FTP com, per exemple, baixar un fitxer, pujar-lo, modificar-ne el nom...

Per a cada ordre del client, el servidor emet una resposta pel canal de control en què indica l'estatus de l'execució de l'ordre rebuda. Per exemple, el client envia el seu nom d'usuari i contrasenya, el servidor els valida i retorna una resposta positiva. Si l'ordre implica la transferència de dades, aquesta es realitza pel canal de dades.

Poseu observar en els exemples de sessió FTP d'aquest mòdul els codis de resposta que es donen a les diverses ordres que realitza el client. Els trobareu a l'apartat "2.6 Modes d'accés al servidor".

El protocol FTP defineix un conjunt de respostes FTP consistents en un codi numèric de tres xifres i un text descriptiu de la resposta, com per exemple "250 Directory successfully changed". Tot el diàleg client/servidor té forma d'ordres i respostes.

2.2 Instal·lació i configuració del servidor

Hi ha moltes aplicacions FTP en el mercat, tant per a clients com per a servidors. Al mateix temps, hi ha versions de text i gràfiques per a cada cas. Hi ha moltes aplicacions que són de font pública i que es poden baixar gratuïtament.

Aplicacions FTP recomanables

Més que fer una enumeració de les aplicacions FTP actuals, us suggerim que busqueu a Internet quines aplicacions estan "de moda" i esbrineu les característiques que les fan especials.

La majoria de sistemes GNU/Linux proporcionen l'aplicació client **ftp** o **lftp** (*lftp* és una versió més nova i simple). També disposen d'una aplicació servidor, entre d'altres, anomenada **vsftpd** (*very secure FTP daemon* o dimoni FTP molt segur).

Una de les eines més usades per fer baixades FTP de repositoris públics són els navegadors web. Els navegadors web permeten utilitzar el protocol FTP per realitzar baixades, però no proporcionen totes les prestacions que pot arribar a tenir un client FTP específic. Per accedir a un servidor FTP amb un navegador, n'hi ha prou d'indicar l'URL del protocol i el servidor al qual es vol accedir (per exemple: ftp://ftp.rediris.es).

Així, doncs, quan parlem d'instal·lar el servei FTP fem referència al procés d'instal·lació i configuració del programari del servidor. Això es fa de manera molt similar a la d'altres serveis de xarxa (com els serveis DHCP, DNS, HTTP...): es tracta d'instal·lar els paquets o *tarballs* (fitxers .tar) de l'aplicació servidor i fer-ne la configuració apropiada.

Per fer això cal plantejar-se els passos següents:

- Quin programari proporciona aquest servei? Quines característiques té? Com es pot adquirir?
- Obtenir l'aplicació que proporciona el servei FTP.
- Observar l'estat de la xarxa actual. Està ja el servei en funcionament? Existeix ja un servidor FTP instal·lat i actiu?
- Instal·lar l'aplicació servidor.
- Comprovar que la instal·lació s'ha efectuat correctament.
- Configurar el servei en el servidor i comprovar que els clients hi poden accedir.
- Comprovar que el servei funciona correctament.

Cerca d'FTP a Internet

Usualment, l'administrador s'informa mitjançant el seu cercador preferit, per exemple Google, i de webs com la Viquipèdia. Proveu a buscar "FTP" o "FTP server" en aquests llocs web.

Usualment l'administrador acaba utilitzant l'aplicació servidor que li proporciona el mateix sistema operatiu que està utilitzant. Si utilitzeu Windows, l'empresa Microsoft ofereix una aplicació pròpia, però també en podeu trobar d'altres a Internet. Igualment, si utilitzeu GNU/Linux, segurament la mateixa distribució ja proporciona un servidor FTP o bé n'existeix algun de clàssic provinent d'Unix. De totes maneres en podeu obtenir d'altres a Internet.

L'eina que s'utilitzarà en aquesta unitat per oferir el servei FTP és l'aplicació **vsftpd** o *very secure FTP daemon*.

2.2.1 Instal·lació de l'aplicació servidor

Els usuaris de GNU/Linux poden buscar fàcilment per Internet paquets de servidor FTP usant eines com *yum* o *apt-get* i els repositoris de paquets apropiats segons quina sigui la distribució que utilitzin. A més, sempre poden usar els cercadors web per localitzar tot allò que els faci falta.

Un cop instal·lat el programari caldrà identificar què s'ha instal·lat. Quins paquets i què contenien. A vegades no s'instal·laran paquets sinó fitxers *tarball*, el contingut dels quals també caldrà saber examinar. És important saber identificar quins dels components instal·lats corresponen a fitxers executables, quins a fitxers de configuració i quins a fitxers de documentació.

Tot servei instal·lat s'ha de configurar apropiadament i posar en marxa. Per tant, caldrà saber gestionar l'estat del servei (engegar, aturar, reiniciar...) i definir l'estat que ha de tenir en els diferents *runlevels* (nivells d'execució) del sistema.

En definitiva, el procediment d'instal·lar inclourà usualment:

- Buscar el programari del servei (sigui en format de paquets .deb, .rpm o .tar) i descarregar-lo utilitzant l'eina apropiada segons quina sigui la distribució que s'utilitzi.
- Examinar el sistema per identificar quin programari, quins paquets, hi ha instal·lats relacionats amb el servei.
- Identificar els components del servei. Quins són els fitxers executables, quins els de configuració i quins els de documentació.
- Consultar i establir l'estat del servei (engegar i aturar) i saber establir l'estat per defecte per a cada *runlevel*.

2.3 Creació d'usuaris i grups

L'accés a un servidor FTP es pot considerar des de diferents angles. Des del punt de vista de la configuració de xarxes (ports, tallafocs, *TCP wrappers*...) es poden filtrar els amfitrions (*hosts*) que poden accedir al servidor FTP. Des del punt de vista dels usuaris, poden ser usuaris anònims, locals i virtuals.

Els **usuaris** es poden classificar en:

- Usuaris anònims
- Usuaris locals del sistema
- Usuaris propis del servei FTP, anomenats usuaris *virtuals*

L'accés anònim al servidor permet que qualsevol client pugui accedir a l'àrea pública del servidor FTP com a usuari "anonymous". Aquest dret d'accés de lectura es pot concedir o restringir. També se li pot proporcionar a l'usuari anònim el dret a publicar documentació dins de directoris amb els permisos apropiats.

Podeu trobar més informació sobre els usuaris anònims en l'apartat "Configuració de l'accés anònim".

L'accés al servei com a usuaris locals del sistema permet accedir al servei FTP als usuaris que ho són també del sistema on s'executa el servei. És per això que s'anomenen *usuaris locals*, perquè són locals a l'amfitrió on s'executa el servei. Aquest accés es pot permetre o restringir segons es desitgi.

Finalment hi ha l'accés d'usuaris virtuals. Es tracta d'usuaris identificats (no anònims) que no són (o no necessàriament) usuaris del sistema. Són usuaris propis del servei FTP i caldrà portar-ne la gestió pròpia amb les típics fitxers d'usuaris i contrasenyes.

2.3.1 Usuaris locals

Permetre l'accés al servei FTP d'usuaris locals és ben simple: simplement cal activar la directiva *local_enable*. Fet això, els usuaris locals tindran accés de descàrrega. Per permetre'ls pujar documents caldrà configurar més directives.

Els usuaris del sistema accedeixen al seu directori d'inici (*home*), que és el directori de publicació, i en general tenen dret a navegar per tot el sistema de fitxers de manera similar a com ho farien en una sessió d'usuari del sistema. Es pot engabiar (*chroot*) els usuaris en el seu directori d'inici, de manera que aquest passarà a ser l'arrel del sistema de fitxers al qual poden accedir via FTP i, per tant, no en podran sortir.

Els usuaris **locals** accedeixen al seu directori per defecte de publicació. Poden moure's únicament pel subarbre del seu *home* o per tot el sistema de fitxers segons si estan **engabiats** o no.

2.3.2 Usuaris virtuals

Generalment, tot servidor FTP permet un tercer tipus d'usuaris que no són ni els anònims ni els usuaris del sistema. Es tracta d'usuaris propis del servei FTP. L'avantatge d'aquest model és la separació entre els usuaris del servei i els usuaris del sistema. Això permet l'accés identificat a serveis FTP sense necessitat de crear comptes d'usuari en el sistema. També permet un grau més gran de portabilitat, ja que el servei es pot traslladar a un altre entorn (amb uns altres usuaris del sistema) i continuar mantenint els usuaris propis del servei. L'inconvenient d'aquest model és que generalment implica dur l'administració d'una gestió d'usuaris paral·lela a la del sistema i crear els fitxers o una base de dades d'usuaris FTP amb els noms i contrasenyes de cada un d'ells.

Els servidors FTP generalment permeten administrar **usuaris específics** del servei FTP (ni locals ni anònims). Això té l'avantatge de la independència respecte dels usuaris del sistema i presenta l'inconvenient de requerir una administració pròpia d'aquests usuaris.
En el servidor vsftpd aquests usuaris s'anomenen ***virtual users***.

Els usuaris virtuals són comptes d'usuaris que no existeixen realment com a usuaris del sistema. Això proporciona un grau major de seguretat, ja que un compte d'usuari virtual compromès únicament pot explotar les debilitats del servei FTP. En canvi, un compte del sistema compromès pot intentar explotar debilitats de tot el sistema. Una de les finalitats principals de la possibilitat de crear usuaris virtuals és permetre l'accés a contingut que ha de ser accessible a usuaris no validats en el sistema, però que no es vol fer públic per a tothom.

El procés que cal seguir per generar usuaris virtuals en el servidor vsftpd és el següent:

1. Crear la base de dades d'usuaris virtuals.
2. Crear/editar el fitxer de configuració PAM per usar la base de dades creada anteriorment.
3. Generar el directori de publicació de l'usuari virtual tot creant aquest usuari.
4. Generar el fitxer de configuració apropiat per permetre l'ús d'usuaris virtuals.

2.4 Configuració de l'accés anònim

El client que accedeix a un servidor FTP ho pot fer com a usuari identificat (del sistema o del servidor) o amb accés anònim. Tradicionalment, els usuaris anònims

tenen accés a un directori de publicació des d'on poden realitzar descàrregues però no pujar documents al servidor (tot i que es pot configurar per permetre-ho). Aquests usuaris s'identifiquen normalment amb el nom "anonymous" i com a contrasenya és costum que introdueixin el seu correu electrònic (vàlid o inventat).

L'**accés d'usuaris anònims** es fa usualment amb el nom d'usuari **anonymous** i una adreça de **correu electrònic** com a contrasenya. Sovint, els comptes anònims s'usen únicament per descarregar fitxers del directori de publicació, però també se'ls pot permetre publicar documents.

Quan els usuaris anònims poden publicar documents cal determinar en nom de qui ho fan, és a dir, quin és l'usuari, el grup i els permisos que s'assignen en el sistema de fitxers als documents pujats per aquests usuaris. També es pot configurar si se'ls permet crear directoris dins de l'arbre de publicació o no.

En general els elements que cal configurar en un servidor FTP relacionats amb els usuaris anònims són:

- Determinar si es permet l'accés als usuaris anònims.
- Determinar si se'ls concedeix el dret a pujar documents.
- Concedir-los o no el dret a crear directoris en el servidor (en l'àrea de publicació).
- Determinar l'usuari, el grup i la màscara amb els quals pujar els documents (si se'ls permet fer-ho).
- Engabiar (*chroot*) o no l'usuari en el seu accés al sistema de fitxers.
- Establir si cal demanar contrasenya als usuaris anònims.
- Generar una llista de contrasenyes (correus electrònics) no acceptades pel sistema.
- En el sistema de fitxers del servidor, establir els permisos a fitxers i directoris des dels quals es permeti descarregar documents i als quals es puguin pujar documents.

En fer-se la instal·lació del servidor FTP, usualment es crea de manera automatitzada un usuari anomenat **ftp**, que és l'usuari del sistema que representarà els usuaris anònims. Aquest usuari no té dret a realitzar sessions interactives amb el sistema, no té *shell* (**nologin**). Usualment el directori d'inici d'aquest usuari és el directori de publicació del servei FTP. Així, en el cas del servei vsftpd, aquest usuari ftp té com a directori d'inici **/srv/ftp** (tot i que pot variar en segons quines distribucions). És costum que dins d'aquest directori existeixi un directori anomenat **pub**, en el qual hi ha tota la documentació d'accés públic (les descàrregues permeses a tothom).

El codi següent mostra la línia que defineix l'usuari ftp en el fitxer d'usuaris del sistema:

```
root@server:/# grep "ftp" /etc/passwd
ftp:x:116:127:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
root@server:/#
```

L'usuari del sistema **ftp** és l'usuari usat en les connexions anònimes. Aquest usuari té el directori d'inici **/srv/ftp**.

2.5 Limitacions d'accés

Directives

El llistat de totes les directives es pot consultar amb l'ordre:

man 5 vsftpdconf.

Existeixen diverses directives que permeten establir múltiples aspectes del funcionament del servei FTP. Aquest apartat se centra en les directives que afecten als aspectes següents:

- Rendiment
- Mode de transferència
- Seguretat
- Mode del servei: autònom o xinetd
- Gestió dels *logs*
- Bàners globals i missatges de directori

2.5.1 Rendiment

El rendiment o *performance* permet configurar aspectes com el nombre màxim de connexions que pot atendre el servidor concurrentment, el nombre de connexions que es permet a un mateix client, l'ample de banda màxim permès en les connexions locals i en les anònimes... Aquestes són algunes de les directives relacionades amb el rendiment del servei:

- ***accept_timeout*** (60): estableix el *timeout* (o temps d'espera) en segons que té un client per establir connexions segures. Entre parèntesi se n'indica el valor per defecte
- ***anon_max_rate*** (0): estableix la taxa màxima de transferència que es permet als clients anònims. S'estableix en *bytes* per segon i per defecte val zero, que significa il·limitat.
- ***connect_timeout*** (60): estableix el temps màxim que té un client per respondre a una connexió de tipus PORT.

- ***data_connection_timeout*** (300): estableix el nombre màxim de segons en què una sessió de transferència de dades pot estar inactiva. Si se supera aquest límit sense progressar en la transferència la sessió es tanca.
- ***delay_failed_login*** (1): indica el nombre de segons de pausa abans d'indicar un error d'inici de sessió.
- ***delay_successful_login*** (0): indica el nombre de segons de pausa abans de permetre una connexió correcta.
- ***idle_session_timeout*** (300): estableix el nombre màxim de segons que una sessió pot estar inactiva. Passat aquest temps la sessió es tanca.
- ***local_max_rate*** (0): especifica la taxa màxima de transferència (en *bytes* per segon) que pot usar un usuari local. Per defecte val zero, que significa il·limitada.
- ***max_clients*** (0): estableix el nombre màxim de clients que es poden connectar concurrentment. Els clients següents rebran un missatge d'error informant que no es permeten més connexions. Per defecte pren el valor zero, que significa connexions il·limitades.
- ***max_login_fails*** (3): indica el nombre màxim d'intents d'inici de sessió fallits. Quan s'arriba a aquest número es tanca la sessió.
- ***max_per_ip*** (0): indica el nombre màxim de clients que es poden connectar simultàniament des de la mateixa adreça IP. Això permet establir límits al nombre de connexions d'un sol client basant-se en la seva adreça IP.
- ***one_process_model*** (YES): activa el model de funcionament *one process per connection*, que proporciona més velocitat de processament de les peticions client.

2.5.2 Mode d'accés

Els servidors FTP permeten treballar en mode ASCII i binari. Actualment, el servei es configura en mode binari per defecte. El mode ASCII pot presentar algunes vulnerabilitats de seguretat en fer un ús més intensiu dels recursos. Aquestes són algunes de les directives que controlen els modes d'accés:

- ***ascii_upload_enable*** (YES): estableix si es permet o no la transferència de dades en mode ASCII a més del mode binari. Activant aquesta directiva es permet pujar documents al servidor en mode ASCII.
- ***ascii_download_enable*** (YES): és la directiva anàloga a l'anterior per permetre o no les descàrregues en mode ASCII.
- ***async_abor_enable*** (NO): si s'activa, permet a clients FTP antics cancel·lar descàrregues a mig fer. D'altra manera, aquests clients es bloquejaven.

2.5.3 Seguretat

Hi ha diverses qüestions relacionades amb la seguretat d'un servei de xarxa a més de les connexions anònimes i els usuaris locals o virtuals. Algunes són aspectes com el rang de ports a usar, les connexions PASV i PORT, el permís per fer llistats, la configuració PAM a usar, els *TCP wrappers*...

Algunes de les principals directives que tracten aspectes relacionats amb la seguretat són:

- ***pasv_min_port*** (0): indica el mínim port permès per a connexions tipus PASV. Si s'utilitza conjuntament amb la opció *pasv_max_port* permet definir un rang (mínim-màxim) amb els valors de port dinàmic que es poden utilitzar en connexions passives.
- ***pasv_max_port*** (0): indica el port màxim que es pot usar en connexions de tipus PASV. En aquestes connexions s'utilitza un port dinàmic que sempre serà igual o inferior a l'especificat aquí.
- ***pasv_enable*** (YES): si es defineix a *NO*, no es permetran les connexions PASV, de tipus passiu. Per defecte pren el valor *YES*.
- ***port_enable*** (YES): si s'estableix a *NO* no es permetrà la transferència de dades en mode actiu, és a dir, usant el mètode PORT. Per defecte pren el valor *YES*.
- ***tcp_wrappers*** (NO): aquesta opció permet activar la seguretat de les connexions de xarxa usant *TCP wrappers*. Permet establir regles de connexió en funció dels noms d'amfitrions o de les seves adreces.
- ***ls_recurse_enable*** (NO): per defecte es desactiva la propietat de poder fer llistats recursius amb l'ordre *ls -R*. Això generalment es fa per evitar riscos de seguretat, ja que aquesta ordre implica un gran comsum de recursos que pot fer caure el servidor.
- ***local_umask*** (077): estableix el valor per defecte de la màscara de permisos per a aquells elements que publiquin els usuaris locals.
- ***connect_from_port_20*** (NO): indica si les connexions de transferència de dades de tipus PORT han d'usar obligatòriament el port 20 o no.
- ***dirlist_enable*** (YES): especifica si es concedeix el permís de fer llistats. Si s'assigna el valor *NO*, els clients no podran llistar els directoris.
- ***download_enable*** (YES). indica si es permet o no la descàrrega de continguts. Evidentment, el cas general és permetre la descàrrega, de manera que per defecte s'estableix a *YES*.
- ***hide_ids*** (NO): aquesta opció permet ocultar la informació d'usuari i grup en els llistats dels directoris de manera que es mostri sempre el nom *ftp*.

- ***hide_file*** (none): aquesta opció permet indicar patrons de noms de fitxer que seran exclosos dels llistats dels directoris. És a dir, els noms de fitxers i de directoris que coincideixin amb els patrons especificats no seran vistos pels clients.
- ***pam_service_name*** (vsftpd): indica el nom del servei PAM que s'utilitza. Si usualment la configuració PAM utilitza el directori /etc/pam.d i aquesta directiva s'estableix a vsftpd, significa que espera trobar un fitxer amb aquest nom dins del directori.
- ***ftp_username*** (ftp): especifica el nom de l'usuari real del sistema que s'utilitzarà quan es realitzin connexions anònimes. És a dir, els clients anònims són mapejats a aquest usuari.

2.5.4 Mode del servei: autònom o xinetd

El servidor vsftpd pot funcionar en mode autònom (*stand-alone*) o dins del superservei de xarxa xinetd. A més, es poden executar diverses instàncies del servidor per atendre diferents seus virtuals o per atendre amb més eficiència les peticions. Els següents són exemples de directives del mode autònom:

```
# Standalone mode
listen=YES

This tells vsftpd to run in standalone mode. Do NOT try and run vsftpd from
an inetd with this option set – it won't work, you may well get 500 OOPS:
could not bind listening socket.

One further note on standalone mode, regarding virtual IPs. This is very
easy – just run one copy of vsftpd per virtual IP (remembering to give each
a separate config file on the command line).
Distinguish which vsftpd is for which virtual IP with a setting like this
in the vsftpd.conf:

listen_address=192.168.1.2
```

- ***listen*** (YES): és la directiva que indica al servidor que ha de funcionar en mode autònom.
- ***listen_address*** (adreça IP): indica per quina adreça IP en concret escolta el servidor. Es poden engegar diversos servidors, cada un amb el seu propi fitxer de configuració i atenent a una adreça IP concreta. Així es poden implementar seus virtuals.

El fragment de codi següent mostra un exemple de fitxer de configuració del servei vsftpd dins del superdimoni de xarxa xinetd. Usualment, els serveis es configuren amb un fitxer amb el mateix nom que el servei dins del directori apropiat de xinetd, usualment /etc/xinetd.d.

```
# vsftpd is the secure FTP server.
service ftp
{
```

```
        disable                 = no
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/local/sbin/vsftpd
        per_source              = 5
        instances               = 200
        no_access               = 192.168.1.3
        log_on_success          += PID HOST DURATION
        log_on_failure          += HOST
}
```

- ***disable*** (no): indica que el servei ha de ser escoltat per el xinetd. Si pren el valor *YES*, el servei queda inhabilitat.
- ***socket_type*** (stream): indica que utilitza TCP.
- ***wait*** (no): indica si el servei és amb un sol *thread* o *multithread*. És a dir, si s'executa una sola instància del servei per atendre totes les peticions o si xinetd pot llançar múltiples instàncies. El valor *no* permet ser multithread.
- ***user*** (root): el servei s'executa en nom de l'usuari *root*.
- ***server*** (/usr/local/sbin/vsftpd): especifica el nom de l'executable del servidor.
- ***per_source*** (5): estableix un màxim de 5 connexions simultànies des d'un mateix client.
- ***instances*** (200): permet un màxim de 200 connexions concurrents, un màxim de 200 clients simultànis.
- ***no_access*** (192.168.1.3): denega l'accés al servidor a clients amb aquesta adreça IP origen.
- ***log_on_success*** (+= PID HOST DURATION): especifica el format dels *logs* quan es realitzen connexions satisfactòries. Es desa l'adreça IP, el PID i el temps invertit en la connexió.
- ***log_on_failure*** (+= HOST): especifica el format dels missatges de *log* per a les connexions fallides. Simplement s'enregistre l'adreça del client.

2.5.5 Logs

Els aspectes principals a descriure en les directives de *logs* són indicar si cal generar els *logs* de les connexions clients o no i establir el format que han de tenir els missatges. El següent és un llistat de les principals directives implicades:

```
# Activate logging of uploads/downloads.
xferlog_enable=YES

# You may override where the log file goes if you like. The default is shown
# below.
xferlog_file=/var/log/vsftpd.log
```

```
7
8  # If you want, you can have your log file in standard ftpd xferlog format.
9  # Note that the default log file location is /var/log/xferlog in this case.
10 xferlog_std_format=YES
```

- ***xferlog_enable*** (YES/NO): indica si cal generar *logs* de les connexions que es realitzen.
- ***xferlog_file*** (/var/log/vsftpd.log): indica la ubicació i el nom del fitxer on es desaran els *logs*.
- ***xferlog_std_format*** (YES/NO): especifica que el format que han de tenir els missatges de *log* ha de ser l'estàndard.

2.5.6 Bàners i missatges

En el servidor FTP es poden especificar diversos bàners o missatges de capçalera, que es mostren segons convingui. Per exemple, és habitual mostrar un bàner de benvinguda als usuaris que inicien sessió en el servidor.

```
1  # You may fully customise the login banner string:
2  # This  string option allows you to override the greeting banner displayed by
3  # vsftpd when a connection first comes in.
4  ftpd_banner=Welcome to blah FTP service.
5
6  # This  option is the name of a file containing text to display when someone
7  # connects to the server. If set, it overrides the banner string provided by
       the
8  # ftpd_banner option.
9  banner_file=
```

Un element diferent dels bàners són els missatges (*messages*). En els directoris de publicació es poden posar fitxers anomenats *.message* que poden contenir, per exemple, una descripció dels continguts del directori. El servidor FTP mostra automàticament el contingut d'aquests fitxers en forma de capçalera quan l'usuari accedeix al directori.

```
1  # Messages, banners
2
3  # Activate directory messages - messages given to remote users when they
4  # go into a certain directory.
5  dirmessage_enable=YES
6
7  # This option is the name of the file we look for when a new directory is
8  # entered. The contents are displayed to the remote user. This option is only
9  # relevant if the option dirmessage_enable is enabled.Default: .message
10 message_file=
```

2.6 Modes d'accés al servidor

Verificar el funcionament del servidor és tan senzill com tractar de connectar des d'un client FTP al servidor i fer diverses sol·licituds en una mateixa sessió. Des de l'entorn de text, el mecanisme més simple per verificar el funcionament sempre és:

- Usar un client FTP text i realitzar les comprovacions.
- Tractar de simular un diàleg FTP usant una sessió Telnet. Com que el protocol FTP té la particularitat que utilitza dos ports, això serà una mica més difícil. Caldrà un *telnet* per al diàleg de control i un altre per a cada transferència de dades.

A part de realitzar una sessió FTP de prova per verificar el funcionament del servei, un administrador també ha de saber:

- Observar l'estat dels ports amb Nmap.
- Observar l'estat del servei.
- Monitorar el trànsit FTP amb utilitats tipus *ss*, *netstat* i IPTraf.
- Monitorar el trànsit amb Wireshark.

2.6.1 Sessió FTP

En la secció "Annexos" del web d'aquest mòdul trobareu una captura del trànsit de xarxa corresponent a una comunicació (diàleg) FTP.

En aquest apartat es mostra un exemple de sessió FTP usant el client text FTP i un exemple de sessió simulant el diàleg amb diverses connexions TCP amb *telnet*. En una mateixa sessió es poden realitzar diverses ordres de consulta, descàrrega i pujada de fitxers. També es combina la transferència de dades en mode actiu i passiu.

Exemple de diàleg FTP

El diàleg client-servidor té forma d'ordres i respostes, tal com es pot veure a continuació, on es realitza una connexió FTP i s'executen diverses ordres. S'utilitza el client FTP per defecte i un servidor vsftpd instal·lat al *localhost*. En aquesta sessió es realitza transferència de dades tant en mode **passiu** com en mode **actiu**.

```
1 Ordre      ports    comanda/resposta
2 [root@portatil ~]# ftp localhost
3     c49962 – s21    ... establiment de la connexió TCP...
4     s21 – c4992     Connected to localhost (127.0.0.1).
5     s21 – c4992     220 (vsFTPd 2.0.5)
```

```
6
7  Name (localhost): anonymous
8      c4992 – s21     ... enviar el nom d'usuari al servidor...
9                      USER anonymous <CRLF>
10     s21 – c4992     331 Please specify the password.
11
12 Password:
13     c49962 – s21    ... enviar la contrasenya al servidor...
14                     PASS usuari@fpoberta.cat
15     s21 – c4992     230  Login successful.
16     s21 – c4992     Remote system type is UNIX.
17     s21 – c4992     Using binary mode to transfer files.
18
19 ftp> cd pub
20     c49962 – s21    ... canviar al directori pub...
21                     CWD pub
22     s21 – c4992     250 Directory successfully changed.
23
24 ftp> dir
25     c49962 – s21     ... indicar el port dinàmic client ...
26                      PORT 127,0,0,1,137,108 <CRLF>
27     s21 – c49962     200 Port command succesfull
28     c49962 – s21     ... llistar el directori ...
29                      LIST <CRLF>
30 *   s20 – c35180     ... establiment connexió canal dades...
31 *   c35180 – s20     ... connexió TCP client/servidor...
32     s21 – c49962     150 Here comes the directory listing
33 *   s20 – c35180     total 2
34                      –rw–r––r–– 1 0 0 110 May 31 12:15 llistat
35                      –rw–r––r–– 1 0 0 362047 Feb 23 16:12 services
36                      ... tancament de la connexió de dades...
37     s21 – c4992      226 Directory send OK.
38
39 ftp> get carta.txt
40     c49962 – s21     ... descarregar el fitxer carta.txt...
41     c49962 – s21     ... demanar el mode passiu al server...
42                      PASV
43     s21 – c49962     227 Entering Passive Mode (127,0,0,1,142,120).
44     c49962 – s21     RETR carta.txt
45     s21 – c49962     150 Opening BINARY mode data connection
46                      for carta.txt (110 bytes)
47 *   c435181 – s36472 ... establiment connexió canal dades...
48 *   s36472 – c435181 ... connexió TCP client/servidor...
49 *                    ... transferència del contingut del fitxer...
50 *                    ... tancament de la connexió de dades...
51     s21 – c4992      226 File send OK
52
53 ftp> quit
54     c49962 – s21     QUIT <CRLF>
55     s21 – c49962     221 Goodbye.
```

El seguiment de la sessió és el següent:

- El client FTP es connecta al servidor via *localhost* i s'identifica com a usuari *anonymous* amb la contrasenya usuari@fpoberta.cat. Observeu que el servidor respon amb "230 Login successful": indica que el sistema operatiu és Unix i que el mode de transferència és binari.
- El client canvia al directori pub amb l'ordre d'usuari *cd pub*. Aquesta ordre correspon a l'ordre *CWD pub* del protocol FTP.
- El client demana fer un llistat amb l'ordre *dir*. Internament, el client aplica aquesta ordre fent dues ordres al servidor: PORT i LIST. Amb l'ordre *PORT 127,0,0,1,137,108*, el client indica al servidor quin és el seu port dinàmic. La

transferència del llistat es realitzarà del port 20 del servidor al port dinàmic del client. El mode de transferència és **actiu**.

- Tot seguit l'usuari client demana descarregar un fitxer amb l'ordre *GET carta.txt*. El client FTP transforma aquesta petició GET en dues ordres al servidor: PASV i RETR. La primera demana al servidor que operi en mode **passiu** i el servidor ho fa responent "227 Entering Passive Mode (127,0,0,1,142,120)". És a dir, el servidor informa de quin és el port (dinàmic) que utilitzarà per a la transferència de dades en lloc del port per defecte 20. Així, serà el client qui haurà de prendre la iniciativa d'iniciar la connexió de dades. L'ordre RETR fa que el servidor envii al client el fitxer sol·licitat. La comunicació, per tant, és en mode **passiu** del port dinàmic del servidor al port dinàmic del client.
- Finalment el client tanca la sessió amb l'ordre *quit*.

Mode actiu i mode passiu

Per activar el mode actiu s'utilitza l'ordre PORT A, B, C, D, PH, PL.

El client indica el seu port de dades especificant la pròpia adreça IP: A.B.C.D i el port. El port s'indica en dos octets PL i PH tals que el número del port correspon a l'expressió *port = PH 256 + PL*.

Per activar el mode passiu s'utilitza l'ordre PASV.

El client sol·licita al servidor treballar en mode passiu. El servidor comunica quin és el seu port de dades (que usarà en lloc del port 20). Emet una resposta amb la informació A, B, C, D, PH, PL, que indica la IP del servidor (A.B.C.D) i el port del servidor en el format *port = PH 256 + PL* (el valor del port s'obté dels octets PL i PH aplicant l'expressió indicada).

Exemple de diàleg FTP usant Telnet

La majoria de protocols TCP inicials d'Internet es poden simular usant Telnet al port pertinent i simulant manualment les ordres del protocol. En el cas del protocol FTP això és molt difícil, ja que requereix dues connexions, una de control i una de dades.

Si el servidor treballa en mode actiu és ell el que estableix la comunicació de dades del seu port 20 a un port dinàmic del client. Per poder recrear aquest diàleg caldria una aplicació que escoltés en el port del client (un Telnet escoltant per aquest port).

Si el servidor treballa en mode passiu és el client el que genera la nova connexió de dades al port especificat pel servidor. Això sí que es pot recrear amb un altre Telnet.

Calen dos Telnets per simular una transferència FTP en mode passiu:

- Un pel port 21, per realitzar el control de la comunicació (del client al servidor).
- L'altre (també del client al servidor) per realitzar la transferència d'un port dinàmic del client a un port dinàmic del servidor.

Vegem-ne un exemple:

Sessió 1: canal de control

```
[user@host ~]# telnet localhost 21
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 (vsFTPd 2.0.5)

USER pere
331 Please specify the password.

PASS pere
230 Login successful.

PWD
257 "/home/pere"

LIST
425 Use PORT or PASV first.

PORT 127,0,0,1,231,175
200 PORT command successful. Consider using PASV.

LIST
150 Here comes the directory listing.
426 Failure writing network stream.

PASV
227 Entering Passive Mode (127,0,0,1,202,67)
*** iniciar sessió 2 en un altre terminal ***

LIST
150 Here comes the directory listing.
226 Directory send OK.

PASV
227 Entering Passive Mode (127,0,0,1,165,50)
*** iniciar sessió 3 en un altre terminal ***

RETR pere.info
150 Opening BINARY mode data connection for pere.info (70 bytes).
226 File send OK.

QUIT
221 Goodbye.
Connection closed by foreign host.
```

Sessió 2: llistat

```
[user@host ~]# telnet localhost 51779
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
-rw-r--r--    1 500      501          1695 Jun 01 17:19 nou.odt
-rw-rw-r--    1 500      501            70 Jun 01 17:14 pere.info
```

```
Connection closed by foreign host.
```

Sessió 3: '*get*' del fitxer

```
[user@host ~]# telnet localhost 42290
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
per crear un usuari com per exemple pere:
# useradd pere
# passwd pere
Connection closed by foreign host.
```

Per saber quines són les comandes possibles del protocol FTP cal consultar l'RFC del protocol.

El seguiment del diàleg és el següent:

- Fer un Telnet al port 21 del servidor, identificar-se amb les ordres del protocol USER i PASS. Esbrinar quin és el directori per defecte amb l'ordre PWD.
- En intentar fer un llistat del directori amb l'ordre LIST, el servidor contesta que primer cal executar l'ordre PORT o PASV. És a dir, que el client indiqui quin és el seu port de dades o que li mani al servidor treballar en mode passiu (perquè sigui el servidor qui indiqui el seu port dinàmic).
- El client indica el seu port de dades amb l'ordre *PORT 127,0,0,1,231,175*. Aquesta ordre indica la IP del client (127.0.0.1) i el port de dades a usar, el 59311 (231 * 256 + 175 = 59.311). En aquest punt el servidor, intenta una nova connexió TCP entre el seu port 20 i el port indicat pel client. Com que el client no atén aquesta connexió, es produeix un error i el servidor no realitza la transferència.
- En el tercer bloc del llistat Telnet, el client executa l'ordre PASV, que provoca que el servidor respongui indicant el port dinàmic de dades que utilitzarà en lloc del port 20. El servidor respon "227 Entering Passive Mode (127,0,0,1,202,67)". Indica la seva IP (127,0,0,1) i el port que utilitzarà: 202 * 256 + 67 = 51.779.
- En un altre terminal del client es pot iniciar una segona sessió Telnet a aquest port, tal com es pot veure a "Sessió 2: llistat": *telnet localhost 51779*. En aquesta connexió es rebrà la transferència del canal de dades de l'ordre que es realitzi a continuació (si és de transferència).
- En el canal de control es realitza l'ordre LIST i automàticament en la sessió 2 apareix el contingut del llistat i es tanca la connexió per part del servidor.
- Tot seguit es realitza el mateix procés per descarregar el fitxer pere.info. El client realitza l'ordre PASV en el canal de control perquè el servidor indiqui el port de dades que utilitzarà.
- Establir una tercera sessió fent un *telnet* al servidor al port que ha indicat (65 * 256 + 50 = 42.290). En fer l'ordre *RETR pere.info* (equivalent al GET del fitxer), el contingut del fitxer es transfereix a la sessió 3. El llistat que es mostra després de l'establiment de connexió correspon al contingut del fitxer.

- Per acabar, es tanca la connexió del canal de control amb l'ordre QUIT (equivalent al BYE).

2.7 Comunicacions segures

Una característica fonamental del protocol FTP és que quan es va dissenyar no es va tenir en compte la seguretat, més enllà de l'usuari i la contrasenya (que és pròpia del sistema operatiu). Com en altres protocols, la informació que porten els paquets viatja per la xarxa com a text pla, sense encriptar. És a dir, que qualsevol persona amb uns mínims coneixements de xarxes, pot capturar aquests paquets i espiar les dades (usuaris, contrasenyes, informació sensible...). Si, a més, fem servir un medi sense fils, encara s'és més vulnerable.

Com d'altres protocols, s'han afegit extensions per tal de xifrar les comunicacions. En aquest cas hi ha dues maneres de fer-ho: a través del protocol FTPS i el SFTP.

2.7.1 El protocol FTPS

El protocol **FTPS** (FTP sobre SSL/TLS, *FTP over SSL/TLS*) utilitza el mateix mecanisme que altres protocols (HTTPS, etc.). Utilitza la capa de seguretat SSL/TLS per fer que les comunicacions siguin xifrades i proporcionar la confidencialitat i autenticació necessàries per a les comunicacions FTP. Utilitza els mateixos ports que el protocol FTP (20 i 21).

El programari *vsftpd* ja porta preparats uns certificats per tal de poder xifrar les comunicacions, tot i que es poden substituir per d'altres. Per tal d'activar el protocol FTPS cal descomentar les següents línies i reiniciar el servei:

```
ssl_enable=yes
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

Vegem amb més detalls les directives:

- ***ssl_enable***: habilita SSL per tal d'activar el protocol FTPS.
- ***rsa_cert_file***: conté el certificat públic.
- ***rsa_private_key_file***: conté la clau privada.

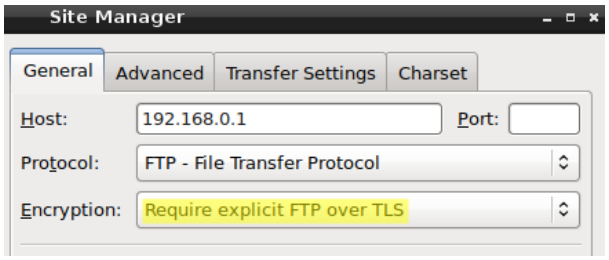
El client normal d'FTP que porten la majoria de distribucions no suporta SSL. Cal instal·lar algun client de text que ho suporti, com *ftp-ssl* o *lftp*.

```
usuari@client:~$ ftp 10.0.2.15
Connected to 10.0.2.15.
220 (vsFTPd 3.0.3)
```

```
Name (10.0.2.15:usuari): usuari
530 Non-anonymous sessions must use encryption.
Login failed.
421 Service not available, remote server has closed connection
ftp> quit
usuari@client:~$ ftp-ssl 10.0.2.15
Connected to 10.0.2.15.
220 (vsFTPd 3.0.3)
Name (10.0.2.9:usuari): usuari
234 Proceed with negotiation.
[SSL Cipher ECDHE-RSA-AES256-GCM-SHA384]
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Amb els clients gràfics també cal indicar que s'usa SSL, tal com es pot veure en la figura 2.2.

Figura 2.2. Connexió a un servidor amb suport FTPS



2.7.2 El protocol SFTP

El protocol **SFTP** és una implementació diferent del protocol FTP. De fet, és una extensió del protocol SSH (*Secure SHell*), que és qui realment ofereix el xifrat. En aquest cas, el protocol utilitza el port 22, que és el d'SSH. Aquest protocol **no** és compatible amb l'FTPS.

No s'ha de confondre amb el protocol *Simple File Transfer Protocol* (protocol simple per a la transferència de fitxers), ja que coincideixen les sigles SFTP. Aquest era una versió lleugera del protocol FTP que utilitzava el port 115 (RFC 913) i que avui en dia no s'utilitza, ja que en el seu lloc es fa servir el TFTP (Trivial FTP).

Per poder utilitzar el protocol SFTP cal instal·lar un altre programari, com per exemple la *suite openSSH*, que incorpora un propi servidor FTP.

```
root@server:/# apt install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  openssh-sftp-server
Suggested packages:
  molly-guard monkeysphere rssh ssh-askpass ufw
The following NEW packages will be installed:
  openssh-server openssh-sftp-server
0 upgraded, 2 newly installed, 0 to remove and 182 not upgraded.
```

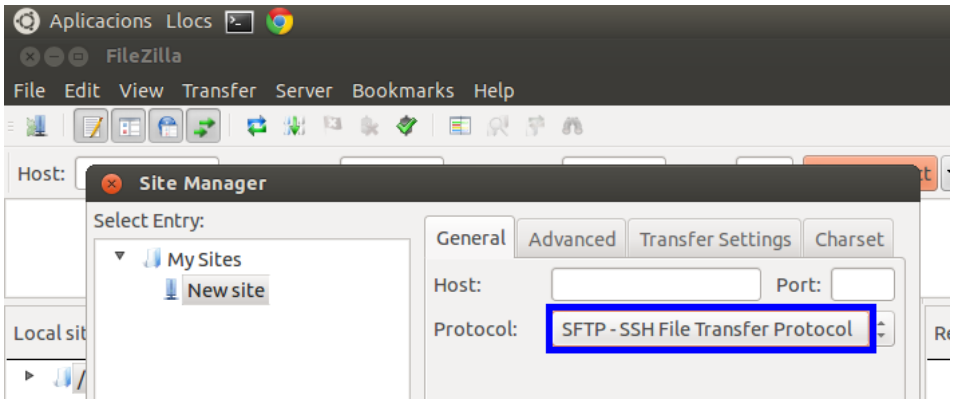
```
Need to get 397 kB of archives.
After this operation, 1609 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

I des d'un client ja ens podem connectar al servidor amb la comanda *sftp*:

```
usuari@client:~$ sftp 10.0.2.15
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ECDSA key fingerprint is SHA256:2cqYKzks+fFtaQutgx8jLfvy8X08lEzxgPdkXYg2DKw.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.15' (ECDSA) to the list of known hosts.
usuari@10.0.2.15's password:
Connected to 10.0.2.15.
sftp>
```

En el cas de voler-se connectar amb un client gràfic com, per exemple, FileZilla, també cal indicar que s'usa SFTP, tal com es pot veure en la figura 2.3:

Figura 2.3. Connexió a un servidor amb suport SFTP



2.8 Clients gràfics i de text

Abans de la popularització del World Wide Web, el servei FTP era profusament usat per a l'intercanvi d'informació i la transferència de fitxers, generalment en mode text o consola, ja que aquest era el mode més comú de treball. Més endavant van anar apareixent eines gràfiques que permetien gestionar còmodament, amb pocs clics, les descàrregues. Avui en dia les eines gràfiques han quedat superades pels omnipresents navegadors, la interfície amb la qual la majoria d'usuaris dialoguen amb el món.

2.8.1 Clients de text

Alguns dels clients més populars per treballar en mode text són:

- **ftp**: el client FTP, utilitat incorporada en tots els sistemes GNU/Linux del món.

- **wget**: una de les eines actualment més populars per a la descàrrega de continguts d'FTP i HTTP (entre d'altres).
- **SFTP**: eina del paquet ofimàtic SSH que permet la connexió FTP segura.

Client FTP

Un client FTP es pot connectar molt fàcilment des del mode d'ordres de la majoria de sistemes operatius. Únicament s'ha d'executar l'ordre *ftp <servidor>*, i automàticament s'inicia una connexió entre el client i el servidor indicat. Usualment, l'aplicació client permet un ús interactiu i es poden obrir i tancar sessions i treballar en diferents servidors al gust del client.

Vegeu un exemple de sessió client:

```
$ ftp ftp.uoc.net            # inicia una sessió al servidor ftp.uoc.net
> quit                       # finalitza la sessió al servidor ftp.uoc.net

$ ftp                        # engega l'aplicació client
> open ftp.rediris.es        # connecta al servidor FTP de RedIRIS
> get carta.txt              # descarrega el fitxer carta.txt
> get file1.txt /tmp/nou.txt # desa file1.txt a /tmp amb el nom nou.txt
> quit                       # finalitza la sessió al servidor atual
> open servidor_nou          # inicia una connexió a un altre servidor
> put /tmp/nou.txt           # puja el fitxer al directori actiu amb el nom nou
    .txt
> cd dir1                    # canvia el directori de destinació a dir1
> put carta.txt              # puja el fitxer carta.txt al directori actual (
    dir1)
```

El client disposa de les ordres FTP que el servidor implementi. No necessàriament s'implementen totes les ordres descrites en el protocol. A més, l'usuari pot disposar de més ordres si el client i el servidor permeten extensions del protocol (utilitats addicionals).

Les ordres més usuals són ***get***, ***mget***, ***put*** i ***mput*** per baixar i pujar fitxers; ***cd*** i ***ls*** per canviar i llistar directoris; ***ascii*** i ***binary*** per indicar el mode de transferència; ***!ordre*** per executar una ordre de sistema operatiu en el servidor, i ***help*** per mostrar la llista d'ordres.

A continuació es mostra la llista d'ordres que implementa el servidor al qual ens hem connectat:

```
ftp> help
Commands may be abbreviated. Commands are:
! debug mdir  sendport  site
$       dir     mget    put     size
account disconnect mkdir pwd    status
append  exit    mls     quit    struct
ascii   form    mode    quote   system
bell    get     modtime recv    sunique
binary  glob    mput    reget   tenex
bye     hash    newer   rstatus tick
case    help    nmap    rhelp   trace
cd      idle    nlist   rename  type
cdup    image   ntrans  reset   user
chmod   lcd     open    restart umask
close   ls      prompt  rmdir   verbose
cr      macdef  passive runique ?
```

```
delete  mdelete proxy   send
```

Aquest és un altre exemple de sessió de text amb un client FTP:

```
[pere@host ~]# ftp localhost
Name (localhost:root): pere
Password:

ftp> pwd
257 "/home/pere"
ftp> ls
-rw-rw-r-- 1 500 501 70 Jun 01 17:14 pere.info

ftp> cd /tmp
ftp> ls
-rw-rw-rw- 1 0 0 1695 Jun 01 17:15 fitxa.odt

ftp> !pwd
/root

ftp> get fitxa.odt
local: fitxa.odt remote: fitxa.odt

ftp> cd ~
250 Directory successfully changed.

ftp> pwd
257 "/home/pere"

ftp> !pwd
/root
ftp> get pere.info
local: pere.info remote: pere.info

ftp> put fitxa.odt nou.odt
local: fitxa.odt remote: nou.odt

ftp> put pere.info /tmp/pere.txt
local: pere.info remote: /tmp/pere.txt

ftp> cd pub
ftp> put pere.info
local: pere.info remote: pere.info
553 Could not create file.

ftp> bye
221 Goodbye.
```

El seguiment del diàleg mostra les accions següents:

Ús del servei FTP

L'ús del servei FTP exigeix una bona gimnàstica mental, ja que l'usuari ha de ser conscient en tot moment de quin és el seu sistema de fitxers local i quin és el sistema remot. La majoria d'ordres tenen una versió per al sistema de fitxers local (*lcd* o "local cd", per exemple) i una altra per al sistema remot (*cd*).

- Connectar al servidor com a usuari identificat: pere. El directori actiu en el servidor FTP és el dirctori d'inici de l'usuari (en aquest exemple /home/pere), com mostra l'ordre ***pwd***.
- Canviar el directori actiu en el servidor amb l'ordre ***cd***. Observar que en el client el directori actiu és un altre, de fet en el client sembla que som l'usurari *root* en el seu dirctori d'inici. L'ordre "*!pwd*" executa en el *shell* del client l'ordre que se li mani.
- La instrucció *get fitxa.odt* descarrega el fitxer d'aquest nom del servidor i el desa en el directori actiu del client amb el mateix nom.

- Es torna a canviar de directori actiu en el servidor (es retorna al dirctori d'inici) i es descarrega el fitxer pere.info. Es desa novament amb el mateix nom en el directori actiu en el client.
- L'ordre *put fitxa.odt nou.odt* permet desar el fitxer fitxa.odt que hi ha en el directori actiu del client amb un nom nou en el directori actiu del servidor.
- Es pot pujar un fitxer indicant tant la ubicació origen en el client com la ubicació de destinació en el servidor. L'ordre *put pere.info /tmp/pere.txt* desa el fitxer pere.info que hi ha en el directori actiu del client a /tmp en el servidor amb el nom pere.txt.
- L'últim exemple mostra que no es pot pujar un fitxer a una ubicació en què no es disposa de permisos per fer-ho.

Per obtenir informació d'una ordre en concret del client FTP es pot consultar la pàgina del manual del client, o també es pot usar l'ordre *help* dins de l'intèrpret client.

```
[user@host ~]$ ftp
ftp> help get
get         receive file
ftp> help mget
mget        get multiple files
```

Client Wget

La utilitat principal de Wget és descarregar un fitxer o un conjunt de fitxers d'un servidor FTP (també d'altres protocols) de forma desatesa, és a dir, sense fer-ho interactivament fitxer a fitxer. Si no indiquem l'usuari i la contrasenya amb els quals ens volem connectar, es realitza una connexió anònima.

```
root@server:/# wget ftp://ftp.rediris.es/welcome.msg
--2020-09-26 14:41:11--  ftp://ftp.rediris.es/welcome.msg
           => 'welcome.msg'
Resolving ftp.rediris.es (ftp.rediris.es)... 130.206.13.2, 2001:720:418:cafd::2
Connecting to ftp.rediris.es (ftp.rediris.es)|130.206.13.2|:21... connected.
Logging in as anonymous ... Logged in!
==> SYST ... done.    ==> PWD ... done.
==> TYPE I ... done.  ==> CWD not needed.
==> SIZE welcome.msg ... 93
==> PASV ... done.    ==> RETR welcome.msg ... done.
Length: 93 (unauthoritative)

welcome.msg
     100%[=============================================================>]      93
       --.-KB/s    in 0s

2020-09-26 14:41:12 (3.42 MB/s) - 'welcome.msg' saved [93]

root@server:/#
```

El llistat següent mostra, d'entre totes les opcions de wget, les referides al servei FTP:

```
root@server:/# wget -h
GNU Wget 1.10.1, un baixador de xarxa no interactiu.
```

```
Forma d'ús: wget [OPCIÓ]... [URL]...
... output suprimit ...

Opcions de FTP:
       --ftp-user=USUARI       estableix l'usuari d'FTP a USUARI.
       --ftp-password=PASS     estableix la contrasenya d'FTP a PASS.
       --no-remove-listing     no suprimir els fitxers «.»listing.
       --no-glob               inhabilita l'ús de comodins de fitxers per a FTP
           .
       --no-passive-ftp        inhabilita el mode de transferència passiu.
       --preserve-permissions  manté els permisos del fitxer remot
       --retr-symlinks         en mode de recursió, baixa els fitxers
                               apuntats per enllaços simbòlics que no siguen
                               directoris
... output suprimit ...
```

Client SFTP

La utilitat SFTP és molt potent i és utilitzada per navegadors de fitxers (per exemple, Nautilus) per poder mostrar sistemes de fitxers remots connectats per FTP. Aquest és el seu format:

```
root@server:/# sftp -h
usage: sftp [-46aCfpqrv] [-B buffer_size] [-b batchfile] [-c cipher]
          [-D sftp_server_path] [-F ssh_config] [-i identity_file] [-l limit]
          [-o ssh_option] [-P port] [-R num_requests] [-S program]
          [-s subsystem | sftp_server] destination
```

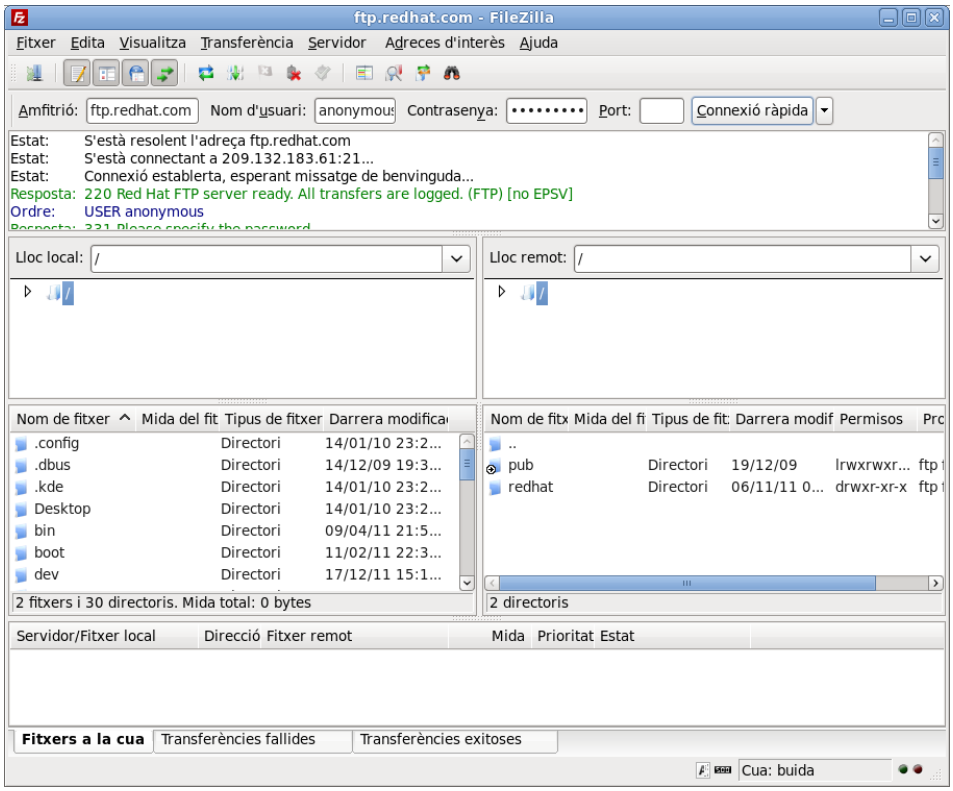
Vegeu un exemple de sessió client usant SFTP per realitzar una transferència segura de fitxers:

```
root@server:/# sftp pere@localhost
Connecting to localhost...
pere@localhost's password:

sftp> pwd
Remote working directory: /home/pere

sftp> ls
nou.odt    pere.info

sftp> put pere.info /tmp/pere.bak
Uploading pere.info to /tmp/pere.bak
pere.info                                 100%   70     0.1KB/s   00:00

sftp> get /tmp/pere.bak
Fetching /tmp/pere.bak to pere.bak
/tmp/pere.bak                             100%   70     0.1KB/s   00:00
```

2.8.2 Clients gràfics

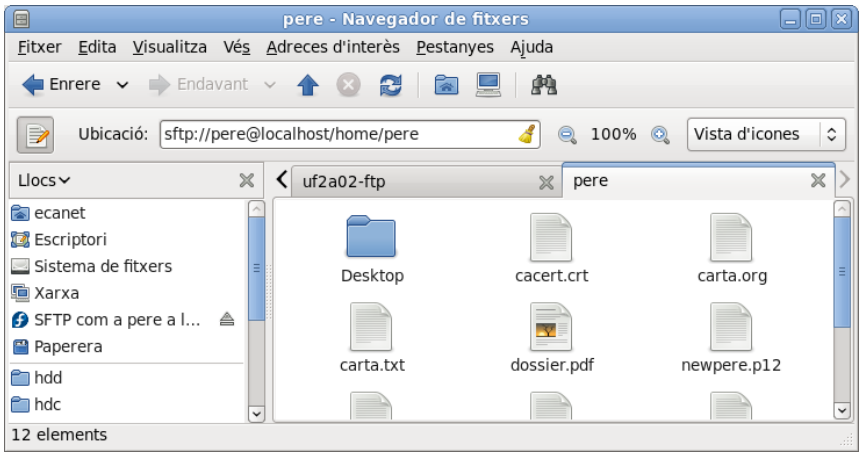
Actualment hi ha una gran varietat de clients gràfics que tenen més o menys èxit. Un dels més populars és FileZilla. En la figura 2.4 podem observar el client FileZilla connectat a la seu ftp.redhat.com.

FIGURA 2.4. Pantalla del client gràfic FileZilla



En el sistema GNU/Linux usualment hi ha algun tipus d'aplicació per navegar pel sistema de fitxers, com Konqueror o Nautilus, depenent de l'escriptori instal·lat. Aquestes eines permeten no només navegar per sistemes de fitxers, sinó també connectar a recursos externs per FTP o SFTP. La figura 2.5 mostra un exemple del funcionament:

FIGURA 2.5. Pantalla del client Nautilus connectat a un recurs per SFTP



2.8.3 El navegador com a client

Avui dia una de les eines imprescindibles per als usuaris d'Internet és el navegador (normalment considerat navegador web). Cada usuari utilitza el que prefereix. Alguns dels més destacats actualment són Firefox i Chrome. Els navegadors

permeten accedir a molts tipus de continguts diferents, entre els quals s'inclouen recursos FTP.

Simplement cal introduir l'URL indicant l'esquema (*schema*) adequat:

```
ftp://<url>
```

La figura 2.6 mostra un exemple de mirall (*mirror*) de descàrregues per FTP per a diferents distribucions de Linux. Els usuaris poden descarregar-se sistemes operatius usant un client navegador qualsevol, com Firefox o Chrome.

FIGURA 2.6. Pàgina de descàrregues per FTP de diferents distribucions de Linux