

Administració de servidors

Remo Suppi Boldrito

PID_00239610

Índex

Introducció	5
Objectius	6
1. Administració de servidors	7
1.1. <i>Active Directory Domain Controller</i> amb Samba4	7
1.1.1. Configuració del Samba Active Directory Domain Controller (AD-DC)	8
1.2. Servei de correu electrònic (<i>mail</i>)	12
1.2.1. <i>Mail Transport Agent</i> (MTA) Bàsic	12
1.2.2. External SMTP	13
1.2.3. <i>Internet Message Access Protocol</i> (IMAP)	15
1.2.4. Aspectes complementaris	16
1.2.5. Instal·lació de producció d'un servidor de correu vinculat externament	18
1.3. Grups de discussió	27
1.4. <i>World Wide Web</i> (httpd)	28
1.4.1. Servidors virtuals	30
1.4.2. Apache + PHP + Mysql + PhpMyAdmin	34
1.4.3. Altres servidors httpd	35
1.4.4. Test de validació i prestacions d'Apache2	36
1.5. Servidor de WebDAV	41
1.6. Proxies	44
1.6.1. Apache com a <i>reverse proxy</i> i amb balanç de càrrega ..	46
1.6.2. Apache com a <i>Forward Proxy</i> i <i>Proxy cache</i>	48
1.6.3. Servei de <i>proxy</i> : Squid	49
1.6.4. <i>Proxy SOCKS</i>	52
1.7. Seguretat a Apache	55
1.8. Servidor de wiki	60
1.8.1. Instal·lació ràpida	60
1.8.2. Instal·lació de servidor	61
1.9. Gestió de còpies de seguretat (<i>backups</i>)	62
1.9.1. Programes habituals de còpies de seguretat	63
1.9.2. <i>rdiff-backup</i> i <i>rdiff-backups-fs</i>	66
1.10. <i>Public Key Infrastructure</i> (PKI)	67
1.11. Open Computer and Software Inventory Next Generation (OCS)	73
1.12. GLPi	75
1.13. Supervisor (Process Control System)	77

1.14. OwnCloud. File Sync & Share Server 79

Activitats 82

Bibliografia 83

Introducció

La interconnexió entre màquines i les comunicacions d'alta velocitat han permès que els recursos que s'utilitzin no estiguin al mateix lloc geogràfic de l'usuari. UNIX (i per descomptat, GNU/Linux) és probablement el màxim exponent d'aquesta filosofia, ja que des del seu inici ha fomentat l'intercanvi de recursos i la independència de dispositius. Aquesta filosofia s'ha plasmat en una cosa comuna avui dia com són els serveis. Un servei és un recurs (que pot ser universal o no) que permet, sota certes condicions, obtenir informació, compartir dades o simplement processar la informació a distància. El nostre objectiu és analitzar els serveis que permeten el funcionament d'una xarxa (a més dels serveis ja esmentats en el mòdul de xarxa d'Administració GNU/Linux). Generalment, dins d'aquesta xarxa hi haurà una màquina (o més, segons les configuracions) que farà possible l'intercanvi d'informació entre les altres. Aquestes màquines es denominen *servidors* i contenen un conjunt de programes que permeten que la informació estigui centralitzada i sigui fàcilment accessible. Aquests serveis permeten la reducció de costos i amplien la disponibilitat de la informació, però s'ha de tenir en compte que un servei centralitzat presenta inconvenients, ja que pot quedar fora de servei i deixar sense atenció tots els usuaris. En aquest mòdul, es veuran els principals serveis que permeten que una màquina GNU/Linux jugui un paper molt important en una infraestructura tecnològica, tant a centralitzar i distribuir dades com a ser punt d'informació, accés o comunicació. D'altra banda, i amb l'avenç de les arquitectures (programari) orientades a serveis (SOA - *Service Oriented Architecture*), i les tecnologies de desenvolupament d'aplicacions que s'han estandarditzat en aquest paradigma de disseny de sistemes distribuïts, GNU/Linux s'ha transformat en la infraestructura per excel·lència que dóna suport a la creació de sistemes d'informació altament escalables. Aquest tipus d'arquitectura (SOA) s'ha transformat en una part essencial del desenvolupament de programari distribuït, ja que permet la creació de sistemes distribuïts eficients, que aprofiten tota la infraestructura subjacent, i estableix una interfície ben definida a l'exposició i crida de serveis web (de manera comuna però no exclusivament), cosa que facilita la interacció entre els sistemes propis i externs.

Serveis replicats

Una arquitectura de servidors ha de tenir els serveis replicats (*mirrors*) per a solucionar els inconvenients que comporta.

Objectius

En els materials didàctics d'aquest mòdul, trobareu els continguts i les eines procedimentals per aconseguir els objectius següents:

- 1.** Presentar els aspectes més rellevants dels conceptes involucrats, tant en un nivell teòric com pràctic, en l'estructura de servidors/serveis en un sistema GNU/Linux.
- 2.** Analitzar els conceptes relatius a serveis i servidors específics d'un sistema GNU/Linux.
- 3.** Experimentar amb la configuració i adaptar la instal·lació de serveis a un entorn determinat.
- 4.** Analitzar i participar en discussions sobre les possibilitats actuals i futures de nous serveis i els obstacles que hi ha bàsicament en aspectes de seguretat en els diferents entorns de treball GNU/Linux (servidor, escriptori multimèdia, escriptori ofimàtica, encaminador o *router*, etc.).

1. Administració de servidors

Els serveis es poden classificar en dos tipus: de vinculació ordinador-ordinador o de relació home-ordinador. En el primer cas, es tracta de serveis requerits per altres ordinadors, mentre que, en el segon, són serveis requerits pels usuaris (encara que hi ha serveis que poden actuar en ambdues categories). Dins del primer tipus es troben els serveis d'*Active Directory (Domain Controller)* o els serveis d'emmagatzematge intermedi (*proxies*). Dins de la segona categoria es preveuen serveis d'intercanvi d'informació en nivell d'usuari, com el correu electrònic (MTA, IMAP, POP), *news*, *World Wide Web* o *wiki*. Per a mostrar les possibilitats de GNU/Linux Debian, es descriurà cadascun d'aquests serveis amb una configuració mínima i operativa, però sense descuidar aspectes de seguretat i estabilitat.

1.1. *Active Directory Domain Controller* amb Samba4

Un dels aspectes més importants en la integració de sistemes és la gestió d'usuaris i identificació centralitzada, així com les autoritats d'autenticació i els permisos. En molt llocs basats en Windows, aquesta tasca és portada a terme per un *Active Directory* (AD, servei de directori en una xarxa distribuïda d'ordinadors (de vegades, també anomenat PDC per *Primary Domain Controller*), i és important disposar d'eines de la banda d'*Open Source* que permetin gestionar aquest tipus d'entorns. Samba versió 4 és una nova distribució d'aquest popular programari que permet la integració amb sistemes Windows actuant com a servidor d'arxius i/o impressores i a més, en aquesta última versió i de manera estable, pot actuar com a controlador d'un domini Windows acceptant clients, gestionant usuaris i directoris de manera centralitzada i totalment compatible [s40].

Dels experts d'*Active Directory* (que generalment són consultors especialitzats i amb un alt cost), sabem que AD és una unió (que pot ser molt complicada d'entendre per als administradors que no estiguin dedicats al món Windows) de diferents serveis i tecnologies com ara DNS, Kerberos, LDAP i CIFS. Alguns hi inclouen DHCP també, però com veurem no és necessari en la nostra instal·lació. Estaríem en un error si penséssim que configurant cadascun d'aquests serveis tindríem el resultat del conjunt, però Samba4 presenta una capa d'abstracció, estable i eficient, que els integra per a oferir un producte complex però que es pot configurar i administrar seguint un conjunt de passos sense grans dificultats (encara que no s'ha de pensar que és simple). L'element crític (base dels majors problemes i errors) de l'AD és el DNS (en realitat, Windows utilitzarà l'AD com a DNS), ja que aquest el farà servir per a "agregar extrafi-

cialment” a la llista de noms habituals en un DNS els servidors AD perquè els clients puguin trobar-los i tenir-hi accés.

En relació amb Kerberos i LDAP, els administradors de GNU/Linux saben de la seva potencialitat i complexitat, però en el cas d'AD estan integrats en el paquet i si bé els atorga una certa estandardització del sistema no són integrables amb servidors o altres clients, només s'utilitzen per als seus objectius i particularment quan fem servir Samba4, serà aquest qui els configurarà i gestionarà en el nostre nom amb petites modificacions per part de l'administrador. La versió actual de Samba (V4) no difereix de les anteriors quant a compartició d'arxius i impressores (fins i tot, s'ha simplificat la seva gestió/administració), i a més, amb la implementació d'AD permetrà que aquells administradors que utilitzin Windows puguin continuar utilitzant les seves eines de gestió i administració de domini només apuntant al servidor Samba. Tota la configuració de Samba passarà ara per l'arxiu `smb.conf` (normalment en `/etc/samba/smb.conf`) amb definicions simplificades però permetent la gestió complexa d'un domini Windows mitjançant les eines (també complexes) de Windows com per exemple RSAT (*Remote Server Administration Tools*) i, òbviament, a través del sistema GNU/Linux i la CLI de Samba4 es tindrà accés a tota la configuració i administració de l'AD (sense necessitat d'utilitzar Windows en cap cas) [s40, s43]

A partir de la versió 8 (Jessie) Debian incorpora els paquets de Samba4 (versió 4.2) i per això no és necessari compilar-ho (només s'hauran d'instal·lar els paquets i configurar-los). En la wiki oficial de Samba* es descriuen tots els passos per compilar-ho des de les fonts en cas que no es disposin dels paquets compilats o es desitgi incloure una nova versió.

*https://wiki.samba.org/index.php/build_Samba_from_source

1.1.1. Configuració del Samba Active Directory Domain Controller (AD-DC)

Per fer una prova es considerarà una màquina virtual amb dues interfícies, una amb NAT cap a l'exterior i connexió a Internet, i una altra interfície connectada a una xarxa privada que gestionarà l'AD-DC per a totes les màquines de la xarxa privada (en cas que es desitgi que el servidor sigui per a una xarxa pública el supòsit és el mateix i només s'ha de canviar la interfície en la configuració inicial).

En aquest cas la configuració de `/etc/hosts` contindrà:

```
27.0.0.1      localhost
172.16.1.1   srv.nteum.org  srv
```

I la configuració de `/etc/network/interfaces`:

```
auto lo
iface lo inet loopback
allow-hotplug eth0 eth1
```



```
iface eth0 inet dhcp
iface eth1 inet static
    address 172.16.1.1
    netmask 255.255.255.0
```

A continuació, s'hauran de carregar tots els paquets necessaris a Debian executant:

```
apt-get install acl attr autoconf bison build-essential debhelper
dnsutils docbook-xml docbook-xsl flex gdb krb5-user libacl1-dev
libaio-dev libattr1-dev libblkid-dev libbsd-dev libcap-dev libcups2-dev
libgnutls28-dev libjson-perl libldap2-dev libncurses5-dev libpam0g-dev
libparse-yapp-perl libpopt-dev libreadline-dev perl perl-modules
pkg-config python-all-dev python-dev python-dnspython python-crypto
```

Durant la configuració ens demanarà diferents dades (el valor per defecte l'extraurà de la configuració) i premerem *acceptar* (o el canviarem). Entre ells tenim: *Default Kerberos version 5 realm*: NTEUM.ORG (cal respectar les majúscules), *Kerberos servers for your realm*: srv.nteum.org i *Administrative server for your Kerberos realm*: srv.nteum.org.

A continuació, s'instal·laran els paquets de Samba necessaris:

```
apt-get install smbclient samba winbind
```

L'arxiu de configuració inicial s'ha de tornar a anomenar amb:

```
cd /etc/samba; mv smb.conf smb.conf.org
```

I, a continuació, hem de proporcionar el domini, indicant-li la interfície (a més de *lo*) on es vol que respongui (veure detalls dels paràmetres a [Sam]):

```
samba-tool domain provision --use-rfc2307 --interactive --option="interfaces=lo eth1"
--option="bind interfaces only=yes"
```

A continuació, el programa indicarà el Realm (domini Kerberos que s'ha introduït abans, NTEUM.ORG en el nostre cas) i el nom del domini NTEUM (si es desitja, es pot canviar); després, el rol que volem donar-li [DC], el DNS que s'utilitzarà [SAMBA_INTERNAL] i el servidor de noms que actuarà de *forwarder* per a les peticions externes on s'haurà d'indicar l'extern del nostre proveïdor o un de públic (per exemple, el 8.8.8.8). Finalment, per al domini demanarà una clau que ha de complir uns criteris de seguretat entre majúscules, minúscules, dígit, longitud (per exemple, CaVeMeCat2016).

Reiniciem la màquina i podrem provar el seu funcionament amb*:

```
smbclient -L localhost -U% (o també amb el nom de la màquina -L srv)
```

*Verificarem que els serveis *samba-ac-dc* i *winbind* estan en funcionament o, si no ho estan, els posarem en funcionament des de */etc/init.d*.

```

Domain=[NIEUM] OS=[Windows 6.1] Server=[Samba 4.2.10-Debian]
  Sharename      Type           Comment
  -----
  netlogon       Disk
  sysvol         Disk
  IPC$           IPC           IPC Service (Samba 4.2.10-Debian)
Domain=[NIEUM] OS= [Windows 6.1] Server=[Samba 4.2.10-Debian]
  Server         Comment
  -----
  Workgroup      Master
  -----
  WORKGROUP     SRV

```

Amb l'execució d'aquesta ordre es pot verificar si Samba proveeix els recursos compartits per defecte *netlogon* i *sysvol*.

També es pot provar l'execució de:

```
smbclient //localhost/netlogon -U Administrator -c 'ls'
```

Enter Administrator's password: <introduir el passwd donat durant l'aprovisionament >

```

Domain=[NTEUM] OS= [Windows 6.1] Server=[Samba 4.2.10-Debian]
.          D 0 Tue Jun 21 17:47:49 2016
..         D 0 Tue Jun 21 17:47:53 2016
          3840152 blocks of size 1024. 1970736 blocks available

```

Un aspecte essencial per al correcte funcionament de l'*Active Directory* és que el DNS es trobi ben configurat, ja que, sense les entrades correctes, Kerberos no validarà i en AC-DC no funcionarà. Per fer això, s'haurà de modificar el */etc/resolv.conf* amb (perquè apunti a la mateixa màquina i al domini):

```

domain nteum.org
nameserver 172.16.1.1

```

I, per comprovar que tot és correcte, podem executar les ordres següents i verificar que rebem la resposta que es mostra a continuació:

```
host -t SRV _ldap._tcp.nteum.org.
```

```
_ldap._tcp.nteum.org has SRV record 0 100 389 srv.nteum.org.
```

```
host -t SRV _kerberos._udp.nteum.org.
```

```
_kerberos._udp.nteum.org has SRV record 0 100 88 srv.nteum.org.
```

```
host -t A srv.nteum.org
```

```
srv.nteum.org has address 172.16.1.1
```

Finalment, podem verificar que Kerberos funciona correctament obtenint un tiquet:

```
kinit administrator@NTEUM.ORG
```

Password for administrator@NTEUM.ORG:

Warning: Your password will expire in 41 days on Tue 02 Aug 2016 17:47:52 BST

```
klist
```

```
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: administrator@NTEUM.ORG
Valid starting      Expires              Service principal
21/06/16 19:16:51   22/06/16 05:16:51   krbtgt/NTEUM.ORG@NTEUM.ORG
        renew until 22/06/16 19:16:46
```

A continuació, es crearà un usuari en el domini perquè es pugui connectar des de Windows (i caldrà introduir el *passwd* i la seva repetició –amb els criteris assenyalats anteriorment–):

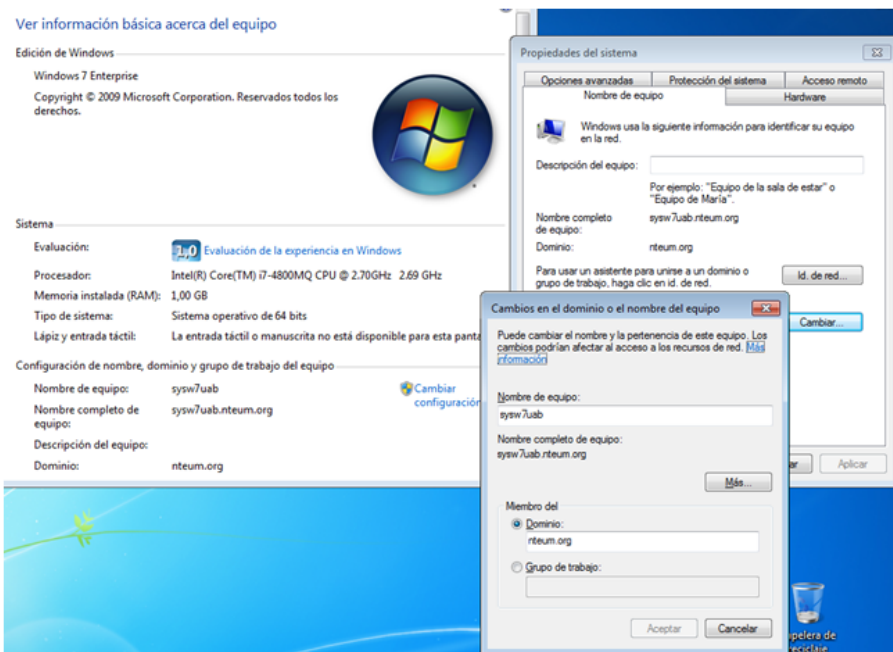
samba-tool user add debian	
samba-tool user list	Per visualitzar els usuaris
samba-tool user delete debian	Per esborrar un usuari
samba-tool user setpassword debian	Per canviar el <i>passwd</i>
samba-tool user enable disable debian	Per habilitar/deshabilitar un compte
samba-tool group list	Per visualitzar els grups
samba-tool group listmembers "Domain Users"	Per visualitzar els usuaris d'un grup
samba-tool group add delete NteumMas	Per afegir/esborrar un grup
samba-tool group addmembers removemembers NteumMas debian	Per afegir/treure membres d'un grup

Finalment ens quedaria posar una màquina (pot ser una de virtual amb Windows) a la mateixa xarxa privada i, accedint-hi com a administrador, modificar la configuració del DNS al servidor AD-DC (172.16.1.1 en el nostre cas). Després modifiquem la configuració d'aquesta màquina perquè es connecti al domini fent: *Panel de Control-> Sistema y Seguridad-> Sistema-> Configuración del nombre del equipo, dominio y grupo de trabajo*. Aquí seleccionem *Cambiar la configuración* i introduïm el domini **nteum.org**. A continuació, ens demanarà l'usuari i la clau de l'administrador que serà *Administrator* i el *passwd* introduït durant la configuració. Si desitgem treure la màquina del domini, haurem d'entrar amb l'usuari local (administrador) de la màquina i repetir el procés invers seleccionant un grup de treball i traient-la del domini.

La figura 1 mostra la finestra de configuració indicada anteriorment sobre Windows 7.

Reiniciem la màquina Windows i ja es podrà connectar a l'usuari del domini (definit prèviament –usuari *debian* en el nostre cas–). És molt probable que, en detectar el domini, sol·liciti que es premi Crtl-Alt-Del per introduir l'usuari. Si

Figura 1



estem executant Virtualbox sobre una màquina Windows, ens donarà problemes, ja que s'activarà aquesta seqüència al *host* i no a la màquina virtual. Per evitar-ho, hem de prémer la tecla *Host-Key-Combination-VBox* (generalment tecla *Crlt-Dreta*) + *Del* i, com a usuari, *NTEUM\debian* (encara que normalment ja indica el domini, per la qual cosa serà tan sols *debian*). Si volem tornar a l'usuari local, farem el *NOM_DE_LA MÀQUINA\usuari_local* (que ens servirà per treure la màquina del domini sempre que vulguem i revertir la connexió al domini deixant la màquina amb els usuaris locals).

Per a administrar el lloc AD, es poden utilitzar les eines de Windows (RSAT), les quals es poden obtenir (sense càrrec) des del lloc del fabricant i amb la guia indicada en [s41] i exemples de configuració en [s42]. També és possible utilitzar l'eina Swat (<https://wiki.samba.org/index.php/swat2>, última versió de novembre del 2012), però la seva instal·lació pot presentar alguns inconvenients (sobretot si Samba4 s'ha instal·lat des de les fonts). Finalment, en l'adreça <https://wiki.samba.org/index.php/samba4/InitScript> podrem trobar l'*script* per a iniciar i apagar el servidor AD de manera automàtica (si bé a Debian 8.x aquesta acció no és necessària ja que el paquet d'instal·lació configura aquests dins de */etc/init.d*).

1.2. Servei de correu electrònic (*mail*)

1.2.1. *Mail Transport Agent*(MTA) Bàsic

Un MTA (*Mail Transport Agent*) s'encarrega d'enviar i rebre els correus des d'un servidor de correu electrònic cap a Internet i des d'Internet, que implementa el protocol SMTP (*Simple Mail Transfer Protocol*). Totes les distribucions incorpo-

ren diferents MTA i, p. ex., els de Debian es poden consultar en el seu paquet virtual *mail-transport-agent* (<https://packages.debian.org/jessie/mail-transport-agent>). Una de les que s'utilitzen habitualment és *exim*, ja que és més fàcil de configurar que altres paquets MTA, com són *postfix* o *sendmail* (aquest últim és un dels precursors). *Exim* presenta característiques avançades com rebutjar connexions de llocs de *spam* coneguts, posseeix defenses contra correus brossa (*junk mails*) o bombardeig de correu (*mail bombing*), i és extremadament eficient en el processament de grans quantitats de correus.

La seva instal·lació és mitjançant `apt-get install exim4-daemon-heavy` (en aquest cas, s'optarà per la instal·lació de la versió *heavy*, que és la més completa i suporta llista d'accés, ACL, i característiques avançades, i en instal·lacions més senzilles es pot optar per *exim4-daemon-light*). La seva configuració es fa mitjançant `dpkg-reconfigure exim4-config`, on una resposta típica a les preguntes efectuades és:

- *General type of mail configuration: Internet site*; el correu és enviat i rebut utilitzat per SMTP.
- *System mail name*: `remix.world` (el nostre domini)
- *IP-addresses to listen on for incoming SMTP connections*: (deixar en blanc)
- *Other destinations for which mail is accepted*: `remix.world`
- *Domains to relay mail for*: (deixar en blanc)
- *Machines to relay mail for*: (deixar en blanc)
- *Keep number of DNS-queries minimal (Dial-on-Demand)?*: No
- *Delivery method for local mail: Maildir format in home directory*
- *Split configuration into small files?*: No

El servidor ja estarà configurat i es pot provar amb

```
echo test message | mail -s "test" adminp@SySDW.nteum.org
```

(canviant l'adreça, per descomptat) i verificar que el correu ha arribat a l'usuari *adminp* (els errors es poden trobar en `/var/log/exim4/mainlog`). La configuració serà emmagatzemada en l'arxiu `/etc/exim4/update-exim4.conf.conf`. Per a configurar autenticació per TLS, ACL i Spam Scaning, podeu consultar la web <https://wiki.debian.org/Exim>. Com hem seleccionat Maildir, en el directori *home* podem llegir els correus amb un client que suporti format Maildir i executant, p. ex., `mutt -f $HOME/Maildir` o fent `export MAILDIR=Maildir` i executant `mutt` veurem el correu enviat prèviament a l'usuari *adminp*.

1.2.2. External SMTP

Quan instal·lem un nou sistema com a servidors o estacions de treball, un aspecte rellevant és el servidor de correu i podem instal·lar grans paquets com els ja esmentats Postfix, Exim o Zimbra (en la seva versió Community <http://www.zimbra.com/>), fent que els correus cap a dominis externs utilitzin els serveis externs d'SMTP (per exemple, els de Google). Per a màquines

virtuals, estacions de treball o portàtils és una mica més complicat ja que generalment tenen IP privades o en xarxes internes, per la qual cosa és necessari tenir un servidor que faci de receptor dels correus externs al nostre domini, és a dir, un servidor que faci les funcions de *smarthost*, per exemple el Google Apps SMTP. Per a detalls de la seva configuració, es pot seguir la documentació d'<https://wiki.debian.org/gmailandexim4>. D'acord amb la informació de Google (<https://support.google.com/a/answer/2956491?hl=es>) i per a comptes gratuïts, el nombre màxim de destinataris permès per domini i dia és de 100 missatges i Gmail reescriurà l'adreça del remitent. Per a la seva configuració, executarem `dpkg-reconfigure exim4-config` i seleccionarem:

- *mail sent by smarthost; received via SMTP or fetchmail.*
- *System mail name: localhost*
- *IP-addresses to listen on for incoming SMTP connections: 127.0.0.1*
- *Other destinations for which mail is accepted: (deixar en blanc)*
- *Machines to relay mail for: (deixar en blanc)*
- *IP address or host name of the outgoing smarthost: smtp.gmail.com::587*
- *Hide local mail name in outgoing mail?: No*
- *Keep number of DNS-queries minimal (Dial-on-Demand)?: No*
- *Delivery method for local mail: mbox format in /var/mail*
- *Split configuration into small files?: Yes*

Aquesta és la configuració més adequada si no es té un IP visible externament. L'enviament sobre el port 587 de Gmail utilitza STARTTLS per a assegurar la protecció del passwd i per a indicar l'usuari i passwd d'accés a Gmail (utilitzeu un compte només per a aquest objectiu, no el compte habitual de Gmail), s'ha d'editar el fitxer `/etc/exim4/passwd.client` i agregar la següent línia.

```
*.google.com:SMTPAccountName@gmail.com:yOuRpaSsw0RD
```

Després, executar (per a evitar que altres usuaris de la màquina puguin llegir el seu contingut):

```
chown root:Debian-exim /etc/exim4/passwd.client
chmod 640 /etc/exim4/passwd.client
```

Gmail reescriurà l'adreça del remitent automàticament, però si no ho fes, o enviem un *smarthost* que no ho fa, hauríem de configurar `/etc/email-addresses` amb totes les combinacions d'adreces possibles per a utilitzar (una per línia) i l'adreça que es reescriurà (per exemple, `nteum@remix.world: Smtppaccount-name@gmail.com`). Després, s'haurà d'executar:

```
update-exim4.conf
invoke-rc.d exim4 restart
exim4 -qff
```

Amb això s'actualitza i recarrega la configuració, i es força a enviar tots els correus que estan pendents. Com vam mostrar amb anterioritat, en el fitxer `/var/log/exim4/mainlog` tindrem els errors si n'hi ha. Si existeixen errors

d'autenticació sobre Gmail, hem de verificar amb `host smtp.gmail.com` quins són els `hosts` que retorna i si aquests concorden amb el patró inclòs en `/etc/exim4/passwd.client`. Si és diferent, canvieu-lo perquè coincideixi.

Cal tenir en compte, finalment, que amb les últimes actualitzacions de seguretat de Gmail la màquina/app des de la qual estem enviant el correu serà considerada no segura (ja que intenta accedir amb usuari i contrasenya) per la qual cosa rebrem una alerta i no el correu. Per a canviar això hem d'accedir a l'administració del nostre compte (tal com s'explica a l'adreça web <https://support.google.com/accounts/answer/6010255>) i en l'apartat de *Sig-in & Security* hem d'anar al darrer apartat *Allow less secure apps* i posar-lo com ON. Amb aquesta configuració ja podrem rebre i enviar correus.

1.2.3. Internet Message Access Protocol (IMAP)

Aquest servei suportat pel *daemon* `imapd` (els actuals suporten el protocol IMAP4rev1) permet accedir a un arxiu de correu electrònic (*mail file*) que es troba en una màquina remota. El servei `imapd` es presta mitjançant els ports 143 (`imap2`), 220 (`imap3`) o 993 (`imaps`) quan suporta encriptació per SSL. Si s'utilitza `inetd`, aquest servidor s'engega mitjançant una línia en el fitxer `/etc/inetd.conf` com:

```
imap2 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/imapd
imap3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/imapd
```

En aquest exemple, es crida el *wrapper* `tcpd` que funciona amb `hosts.allow` i `hosts.deny` per a incrementar la seguretat. Les aplicacions més populars són `courier-imap`, `cyrus-imapd`, `dovecot-imapd`, entre d'altres. Per a provar que el servidor `imap` funciona, es podria utilitzar un client, per exemple Thunderbird/Icedove (Debian), Evolution, Squirrelmail, o qualsevol altre client que suporti IMAP, crear un compte per a un usuari local, configurar-lo de manera adequada perquè es connecti sobre la màquina local i verificar el funcionament de `imap`.

Com a prova de concepte, podem fer la instal·lació de `apt-get install dovecot-imapd` que amb les opcions per defecte permet una connexió encriptada per SSL i sobre bústies *mailbox* (o si volem, sobre bústies *maildir* hauríem de canviar la configuració de `/etc/dovecot/conf.d/10-mail.conf`). Dovecot és un servidor molt potent, per la qual cosa permet una gran quantitat d'opcions i configuracions (consulteu <http://wiki2.dovecot.org/>). Les proves es poden completar configurant Evolution o IceDove perquè es connecti al nostre servidor/usuari i llegir els correus del servidor prèviament configurat. És important notar que alguns clients de correu/servidors d'Imap només suporten el format *mailBox* i no *maildir*, i per aquest motiu s'ha de tenir en compte quan s'utilitzin els clients/servidors d'Imap. En les activitats que hem fet fins ara, tant `exim4` com `dovecot-imapd` suporten tots dos formats i s'han de configurar durant la instal·lació.

1.2.4. Aspectes complementaris

Suposem que com a usuaris, tenim quatre comptes de correu en servidors diferents i volem que tots els missatges que arriben a aquests comptes es recullin en un només, al qual puguem accedir externament, i que hi hagi també un filtre de correu brossa (*antispam*). Primer s'han d'instal·lar `exim4 + imapd` i comprovar que funcionen.

Per a recollir els missatges de diferents comptes, s'utilitzarà `Fetchmail`, (que s'instal·la amb `apt-get install fetchmail`). A continuació, s'ha de crear el fitxer `.fetchmailrc` en el nostre `$HOME` (també es pot utilitzar l'eina `fetchmailconf`), que haurà de ser una cosa semblant a:

```
set postmaster "adminp"
set bouncemail
set no spambounce
set flush
poll pop.domain.com proto pop3
  user 'nteum' there with password 'MyPaSSwOrD' is 'nteum' here
poll mail.domain2.com
  user 'adminp' there with password 'MyPaSSwOrD' is 'adminp' here
  user 'nteum' there with password 'MyPaSSwOrD' is 'nteum' here
```

L'acció `set` indica a `Fetchmail` que aquesta línia conté una opció global (enviament d'errors, eliminació dels missatges dels servidors, etc.). A continuació, s'especifiquen dos servidors de correu: un perquè comprovi si hi ha correu amb el protocol POP3 i un altre perquè provi a fer servir diversos protocols amb la finalitat de trobar-ne un que funcioni. Es comprova el correu de dos usuaris amb la segona opció de servidor, però tot el correu que es trobi s'envia a l'*spool* de correu de `adminp`. Això permet comprovar diverses bústies de diferents servidors, com si es tractés d'una única bústia. La informació específica de cada usuari comença amb l'acció `user`. El `Fetchmail` es pot posar en el cron (per exemple, en `/var/spool/cron/crontabs/fetchmail` agregant `1 * * * * /usr/bin/fetchmail -s`) perquè s'executi automàticament o executar-lo en mode *daemon* (poseu `set daemon 60` en `.fetchmailrc` i executeu-lo una vegada, per exemple, en autostart de Gnome/KDE o en el `.bashrc` i s'executarà cada 60 segons).

Per a treure el correu brossa, es farà servir `SpamAssassin`.

Aquesta configuració s'executarà mitjançant `Procmail`, que és una eina molt potent en la gestió del correu (permet repartir el correu, filtrar-lo, reenviar-lo de manera automàtica, etc.). Un cop instal·lat (`apt-get install procmail`), s'ha de crear un fitxer anomenat `.procmailrc` en el home de cada usuari, que cridarà l'`SpamAssassin`:

```
# Poseu yes per a missatges de funcionament o depuració
VERBOSE=no
# Considerem que els missatges estan en "~/Maildir", canviar si és un altre
PATH=/usr/bin:/bin:/usr/local/bin:
MAILDIR=$HOME/Maildir
```

Podeu instal·lar `SpamAssassin` mitjançant `apt-get install spamassassin`.


```

DEFAULT=$MAILDIR/

# Directori per a emmagatzemar els fitxers
PMDIR=$HOME/.procmail
# Comentar si no volem log de Procmail
LOGFILE=$PMDIR/log
# filtre de Smap
INCLUDERC=$PMDIR/spam.rc

```

L'arxiu `~/ .procmail/spam.rc` conté:

```

# si l'SpamAssassin no és en el PATH, agregueu a la variable PATH el directori
:Ofw: spamassassin.lock
| spamassassin -a

# Les tres línies següents mouran el correu spam a un directori anomenat
# "spam-folder". Si es vol guardar el correu en la safata d'entrada,
# per a després filtrar-lo amb el client, comenteu les tres línies.

:O:
* ^X-Spam-Status: Yes
spam-folder

```

L'arxiu `~/ .spamassassin/user_prefs` conté algunes configuracions útils per a SpamAssassin (consulteu la bibliografia).

```

# user preferences file. Vegeu man Mail::SpamAssassin::Conf

# Llindar per a reconèixer un Spam.
# Default 5, però amb 4 funciona una mica millor
required_hits 4

# Llocs dels quals considerem que mai arribarà Spam
whitelist_from root@debian.org
whitelist_from *@uoc.edu

# Llocs dels quals sempre arriba SPAM (separats per comes)
blacklist_from viagra@dominio.com

# les direccions en Whitelist i Blacklist són patrons globals com:
# "amigo@lugar.com", "*@isp.net", o "*.domain.com".

# Inserteu la paraula SPAM en el subject (facilita fer filtres).
# Si no es desitja comentar la línia.
subject_tag [SPAM]

```

Això generarà un `tag X-Spam-Status: Yes` en la capçalera del missatge si es creu que el missatge és *spam*. Després, s'haurà de filtrar i posar a una altra carpeta o esborrar-lo directament. Es pot fer servir el `procmail` per a filtrar missatges de dominis, usuaris, etc. Finalment, es pot instal·lar un client de correu i configurar els filtres perquè seleccionin tots els correus amb `X-Spam-Status: Yes` i els esborri o els envii a un directori. Després, verificarem els falsos positius (correus identificats com a brossa però que no ho són). Un aspecte complementari d'aquesta instal·lació és que si es desitja tenir un servidor de correu a través de correu web (*webmail*), és a dir, poder consultar els correus del servidor mitjançant un navegador sense haver d'instal·lar un client ni configurar-lo (com consultar un compte de Gmail o Hotmail),

Enllaç d'interès

Per a més informació sobre `procmail` i el filtrat de missatges, consulteu: <http://www.debian-administration.org/articles/242>.

és possible instal·lar Squirrelmail (`apt-get install squirrelmail`) per a donar aquest servei.

Enllaç d'interès

Hi ha altres possibilitats com instal·lar MailDrop en lloc de Procmail, Postfix en lloc d'Exim, o incloure Clamav/Amavisd com a antivirus (Amavisd permet vincular Postfix amb SpamAssassin i Clamav). Per a saber més sobre aquest tema, podeu visitar la següent pàgina web: <http://www.debian-administration.org/articles/364>.

Enllaç d'interès

Sobre Squirrelmail en Debian, consulteu: <http://www.debian-administration.org/articles/200>.

1.2.5. Instal·lació de producció d'un servidor de correu vinculat externament

En aquest apartat es descriurà com instal·lar un servidor de correu de gran capacitat sobre una distribució Debian. L'objectiu és que permeti una gran quantitat de transaccions i serveis externs de correu i també la possibilitat de reexpedir-los cap a altres MTA. També es vol que pugui rebre i emmagatzemar correus per als usuaris definits del sistema o usuaris virtuals des de i cap a Internet, amb IMAP per a l'accés remot, i utilitzant connexions xifrades per protegir la privadesa de la informació i controlant l'Spam que es pugui rebre o reexpedir. Per assolir aquest objectiu és necessària la instal·lació d'un servidor de correu d'altres prestacions (per exemple, Postfix) associat a IMAP (per exemple, Dovecot), un servidor web (per exemple, Apache), un client de *webmail* (per exemple, SquirrelMail o RoundCube) i un conjunt d'utilitats com SpamAssassin, xifrat (SSL/TLS) o antivirus (p.i Amavis/ClamAV). I tot això funcionant conjuntament. El diagrama de flux seria alguna cosa similar al que segueix:

Usuaris virtuals

Els usuaris virtuals són els usuaris del servei, però no del sistema operatiu.

- 1) Un correu arriba per SMTP al port 25 i el rep `Postfix`, que realitza unes comprovacions (l·listes negres / grises / etc.).
- 2) Després passa per `AMaVis`, que l'envia a `SpamAssassin` i després a `ClamAV`.
- 3) Posteriorment, `Postfix` realitza l'expansió d'àlies i pren algunes decisions.
- 4) Finalment, `Dovecot` el posa a disposició dels usuaris per IMAP, ja sigui a través d'un client (`Icedove`, `Evolution`) o a través de *webmail* (`Apache` + `RoundCube`), que hi accedeixen remotament.

Cas 1

La primera opció és instal·lar `iRedMail` en la seva versió *opensource edition*, que no presenta dificultats (configuració per a Debian [Ired]). Per fer això, s'ha de descarregar el paquet des d'<http://www.iredmail.org/download.html>, que tindrà el format *iRedMail-x.y.z.tar.bz2* on *x.i.z* i serà la versió corresponent. En primer lloc, hem de verificar que disposem d'un domini configurat correctament (en el nostre cas *srv.nteum.org*):

iRedMail

`iRedMail` és una unió i integració dels programes esmentats anteriorment juntament amb una interfície administrativa via web.

```
hostname -f  
srv.nteum.org
```

I que en el `/etc/hosts` existeix una línia amb el FQDN:

```
172.16.1.1 srv.nteum.org srv
```

Per a aquestes proves, s'utilitza una màquina virtual amb `eth0` a NAT i connectada a Internet, i amb `eth1` configurada estàticament amb IP 172.16.1.1. Una vegada realitzades aquestes comprovacions, anirem al directori on s'ha descarregat el paquet, descomprimirem l'arxiu `bz2` i executarem l'instal·lador:

```
tar xjf iRedMail-x.y.z.tar.bz2  
cd iRedMail-x.y.z/  
bash iRedMail.sh
```

Uns instants després, l'instal·lador preguntarà:

- 1) El directori d'emmagatzematge dels correus: `/var/vmail`
- 2) La base de dades per emmagatzemar els comptes dels usuaris: per exemple, MySQL (però pot ser MariaDB o OpenLdap)
- 3) El domini del correu: `mail.nteum.org` (no pot coincidir amb el FQDN de la màquina, en el nostre cas `srv.nteum.org`)
- 4) Paraula clau per a l'administrador (`username: postmaster@mail.nteum.org`)
- 5) La selecció de components opcionals: `iRedAdmin` (aplicació web per administrar els comptes), `RoundCube` (*webmail*), `SOGo` (*webmail*, calendari i llibreta d'adreces), `failban2`.
- 6) La verificació final i la instal·lació: [Y]

failban2

`failban2` és una aplicació per bloquejar usuaris que han intentat entrar diverses vegades a un compte per suplantar la seva identitat.

Després de la instal·lació demanarà que es reiniciï la màquina. Totes les dades importants (URL, `passwd` i arxius de configuració) quedaran recollides a l'arxiu `iRedMail.tips`, dins del directori on s'ha iniciat la instal·lació.

A continuació, es poden realitzar les proves de funcionament en les següents URLs:

- 1) `https://srv.nteum.org/iredadmin/`, amb l'usuari i `passwd` del punt 4 anterior, per a la interfície d'administració.
- 2) `http://srv.nteum.org/mail/` o `https://srv.nteum.org/mail/` (si es prefereix, s'hi pot accedir per SSL/TLS) per al client *webmail*.

Amb això es podran enviar correus als comptes creats o al mateix *postmaster* o als usuaris locals. Per enviar-los cap a màquines des de fora, hem d'utilitzar un *relay host*, que pot ser el del nostre proveïdor d'Internet, configurant-lo en */etc/postfix/main.cf*, agregant una línia com *relayhost = nom.domini.ISP* i reiniciant el servei. Per rebre correus en el nostre domini, s'han de configurar els registres MX en el Servidor de DNS que tingui el registre del nostre domini [Ired-DNS], i també podem configurar els clients externs per IMAP [Ired-IMAP]. A [Ired-Relay] amplien com resoldre els problemes de *relayhost* quan estiguin autenticats.

Cas 2

Si es desitja tenir control sobre els paquets i realitzar una instal·lació similar de les mateixes característiques, però tenint el control sobre la instal·lació i els paràmetres/configuració (opció per a entorns amb característiques particulars i administradors avançats), es pot fer el següent:

1) `apt-get install postfix sasl2-bin`. Seleccionem "No configuratió" i fem el següent:

```
cp /usr/lib/postfix/main.cf /etc/postfix/main.cf
```

Editem *vi /etc/postfix/main.cf* i el modifiquem*:

```
mail_owner = postfix                línia 59
myhostname = mail.nteum.org          línia 76
mydomain = nteum.org                 línia 83
myorigin = $mydomain                 línia 104
inet_interfaces = all                 línia 118
mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
local_recipient_maps = unix:passwd.byname $alias_maps    línia 166
mynetworks = 127.0.0.0/8, 172.16.1.0/24    línia 265
alias_maps = hash:/etc/aliases         línia 388
alias_database = hash:/etc/aliases     línia 399
home_mailbox = Maildir/                línia 421
smtpd_banner = $myhostname ESMTP       línia 559
sendmail_path = /usr/sbin/postfix      línia 632
newaliases_path = /usr/bin/newaliases   línia 637
mailq_path = /usr/bin/mailq            línia 642
setgid_group = postdrop                línia 648
#html_directory =                     línia 652
#manpage_directory =                  línia 656
#sample_directory =                   línia 661
#readme_directory =                   línia 665
```

I afegim al final:

```
message_size_limit = 10485760
mailbox_size_limit = 1073741824
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
smtpd_recipient_restrictions = permit_mynetworks, permit_auth_destination,
                                permit_sasl_authenticated, reject
```

Desinstal·lació d'iRedMail

Si desitgem desinstal·lar iRedMail, existeix (a Internet i en els fòrums del propi iRedMail) un *script* (*clean_iredmail.sh*), però que no es troba actualitzat i que deixa diversos rastres dels paquets instal·lats que es poden acabar de desinstal·lar posteriorment (a més de desinstal·lar els paquets, hem d'eliminar els directoris creats a */opt* i les entrades del *crontab*).

*Cal anar amb compte perquè algunes característiques estan definides, però amb altres valors –es mostra la línia que cal modificar, però pot variar en funció de la versió–.

Finalment, executem `newaliases` i, després, `systemctl restart postfix`.

```
apt-get install dovecot-core dovecot-imapd
```

Editem

```
vi /etc/dovecot/dovecot.conf
    listen = *                               línea 30

vi /etc/dovecot/conf.d/10-auth.conf
    disable_plaintext_auth = no             línea 10
    auth_mechanisms = plain login          línea 100

vi /etc/dovecot/conf.d/10-mail.conf
    mail_location = maildir:~/Maildir      línea 24

vi /etc/dovecot/conf.d/10-master.conf
    # Postfix smtp-auth                     línea 95
    unix_listener /var/spool/postfix/private/auth {
        mode = 0666
        user = postfix
        group = postfix
    }
```

Finalment, reiniciem el servei `systemctl restart dovecot`.

2) Amb això ja es pot utilitzar un client IMAP, per exemple Icedove (per a instal·lar-ho, `apt-get install icedove`), i configurar un compte IMAP contra el servidor i un usuari definit en ell.

3) Per instal·lar RoundCube, prèviament haurem d'instal·lar una base de dades, per exemple MySQL, amb `apt-get install mysql-server-5.5` (ens demanarà el *passwd* per a l'usuari *root* de la base de dades). Per comprovar que funciona es pot executar:

```
mysql --defaults-file=/etc/mysql/debian.cnf
```

o també `mysql -p` (i introduir el *passwd*). Després executem el client `mysql`. Veurem el prompt `mysql>` i podrem executar:

```
select host, user, password from mysql.user;  mostra informació
show databases;                               mostra les DB
quit                                           per sortir
```

4) Per a Roundcube i atès que no es troba en el repositori oficial (per diverses raons), però sí a *backports*, és necessari inserir a `/etc/apt/sources.list` la següent línia:

```
deb http://http.debian.net/debian jessie-backports main
```

Fem un `apt-get update` i després

```
apt-get install roundcube roundcube-plugins,
```

seleccionant *mysql* com a gestor de bases de dades i introduint el *passwd* per als seus usuaris.

A continuació, s'ha de modificar */etc/apache2/sites-available/000-default.conf* escrivint les següents línies (per incloure la configuració dins d'Apache) abans del tag *</VirtualHost>*:

```
Include /etc/roundcube/apache.conf
Alias /var/lib/roundcube/
```

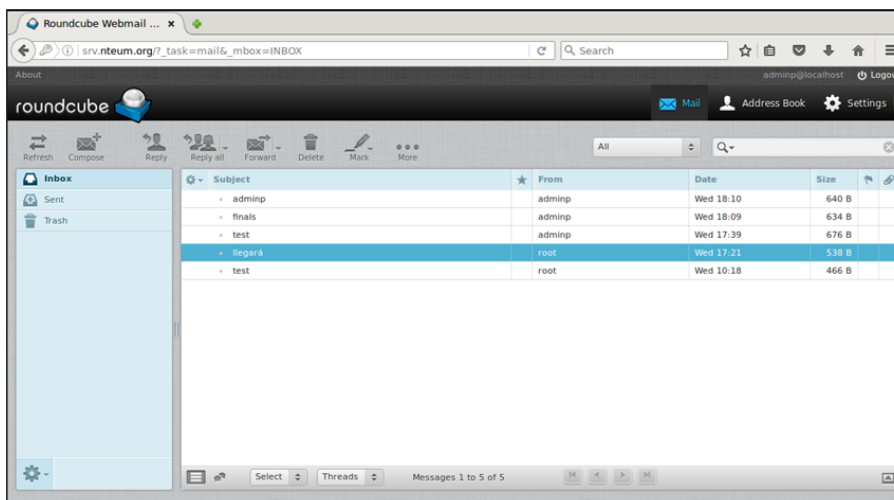
Finalment, reiniciem el servidor amb `systemctl restart apache2` i ens connectem a l'URL (<http://srv.nteum.org>) on podrem accedir amb algun dels usuaris definits en la màquina.

Es pot observar que, a més a més de sol·licitar l'usuari i el *passwd*, també demana el servidor. Atès que és una única instal·lació, es pot modificar l'arxiu */etc/roundcube/config.inc.php* per incloure '*localhost*' en la línia 35 (aproximadament):

```
$config['default_host'] = 'localhost';
```

La figura 2 mostra una pantalla del client webmail implementat per RoundCube.

Figura 2



5) Si es desitja mantenir la privadesa de la informació a través d'una connexió https, és necessari crear primer un certificat*

```
openssl req -newkey rsa:4096 -nodes -sha512 -x509 -days 3650
-nodes -out /etc/ssl/certs/mail.pem -keyout /etc/ssl/private/mail.key
```

*En aquest cas optarem per l'opció més senzilla, però n'hi ha d'altres de disponibles. Es pot consultar [StartSSL].

i respondre les preguntes per al certificat, tenint en compte que cal introduir com a *CommonName* el FQDN del servidor (*srv.nteum.org*). A continuació, per protegir la clau privada fem:

```
chmod 600 /etc/ssl/private/mail.key
```

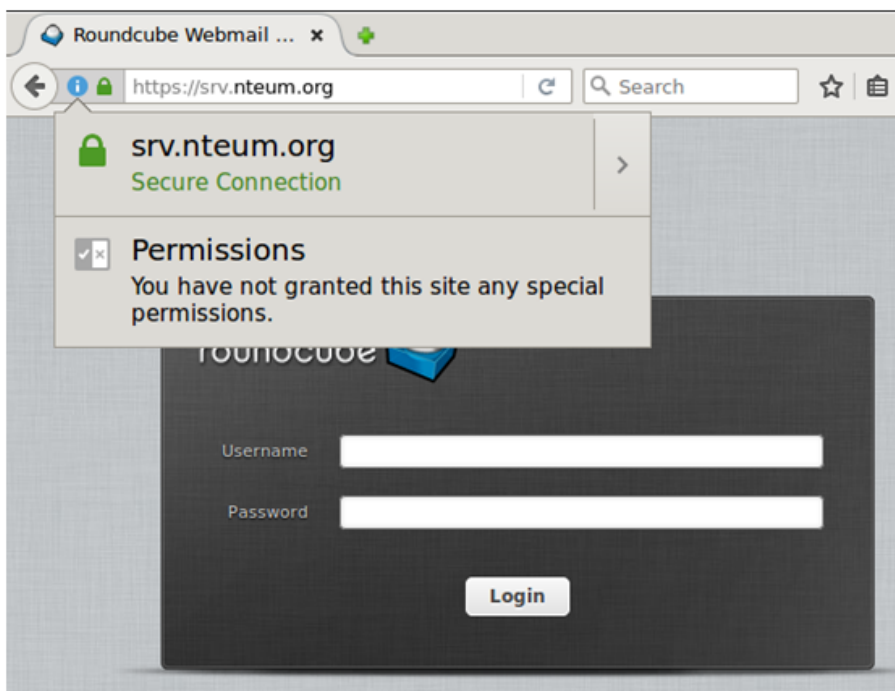
6) Configurem Apache modificant */etc/apache2/sites-available/default-ssl.conf*, en concret, les següents línies:

```
SSLCertificateFile /etc/ssl/certs/mail.pem
```

```
SSLCertificateKeyFile /etc/ssl/private/mail.key
```

Després executem `a2enmod ssl` per habilitar el mòdul de SSL, `a2ensite default-ssl` per habilitar el lloc web i `service apache2 reload` per reiniciar el servei verificant que ens podem connectar a `https://srv.nteum.org/` (després d'acceptar l'excepció, ja que el certificat està signat per nosaltres mateixos), i obtenim la pàgina d'inici de RoundCube per connexió amb SSL, tal com es pot apreciar a la figura 3.

Figura 3



Certificat signat gratuït

Recordeu que, per evitar autosignar el certificat i sempre que es tingui un domini vàlid d'Internet, es pot obtenir un certificat signat gratuït des de StartSSL. Per fer això es poden seguir els passos indicats a [StartSSL].

7) Si es desitja habilitar la privadesa per IMAP (utilitzant IMAPS), s'haurà de modificar Postfix i Dovecot per xifrar la comunicació amb TLS. Per fer això s'utilitzaran els certificats generats en el punt 5 i s'haurà de modificar

a) vi */etc/postfix/main.cf*, i afegir al final:

```
smtpd_use_tls = yes
```

```
smtpd_tls_cert_file = /etc/ssl/certs/mail.pem
smtpd_tls_key_file = /etc/ssl/private/mail.key
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
```

b) vi /etc/postfix/master.cf, i treure el comentari a les línies 28, 29 i 30:

```
smtps inet n - - - - smtpd
-o syslog_name=postfix/smtps
-o smtptd_tls_wrappermode=yes
```

c) vi /etc/dovecot/conf.d/10-ssl.conf

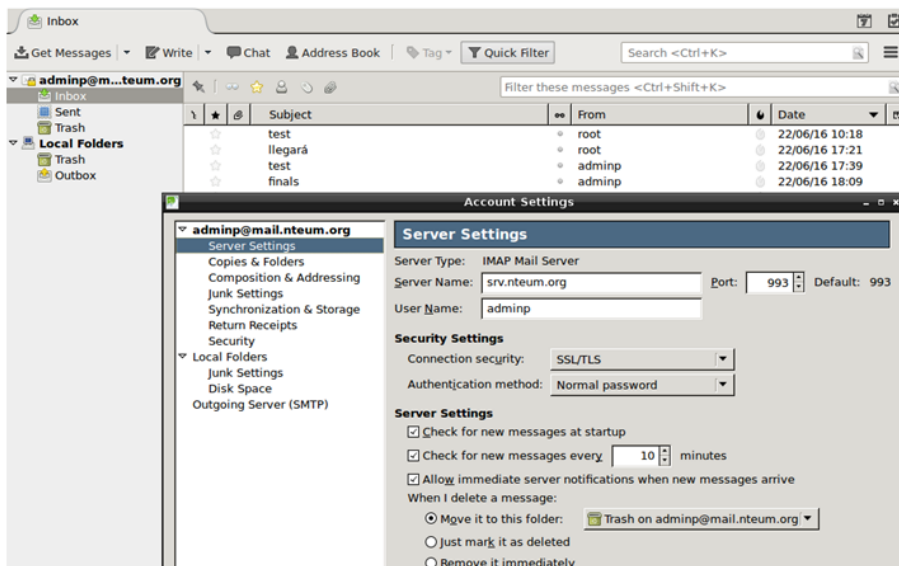
```
ssl = yes                canviar línia 6
ssl_cert = </etc/ssl/certs/mail.pem  canviar línia 12
ssl_key = </etc/ssl/private/mail.key  canviar línia 13
```

d) Reiniciem Postfix, `systemctl restart postfix`

i) Reiniciem Dovecot, `systemctl restart dovecot`

A continuació, a Icedove, canviem la configuració del servidor en les opcions de *security setting* a SSL/TLS i autèntiquem amb *normal passwd*, i ja es podrà accedir per *imaps* al servidor Postfix, tal com ho mostra la figura 4.

Figura 4



Per instal·lar un antivirus associat a Postfix, haurem d'instal·lar primer ClamAV fent:

```
apt-get install clamav
service clamav-freshclam stop
```

per al servei


```
freshclam actualitzar les bases de dades
clamscan -r -i /home verificar el seu funcionament
```

A continuació, hem d'instal·lar els *daemons*:

```
apt-get install clamav-daemon clamsmtp
vi /etc/clamsmtpd.conf
```

```
Header: X-AV-Checked: ClamAV using ClamSMTP afegir header, línia 27
User: clamav canviar l'usuari
```

```
chown -R clamav. /var/spool/clamsmtp canviar uid i gid
chown -R clamav. /var/run/clamsmtp canviar uid i gid
vi /etc/postfix/main.cf afegir al final
```

```
content_filter = scan:127.0.0.1:10026
```

```
vi /etc/postfix/master.cf afegir al final
```

```
scan unix - - n - 16 smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-o disable_dns_lookups=yes
127.0.0.1:10025 inet n - n - 16 smtpd
-o content_filter=
-o local_recipient_maps=
-o relay_recipient_maps=
-o smtpd_restriction_classes=
-o smtpd_client_restrictions=
-o smtpd_helo_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o mynetworks_style=host
-o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

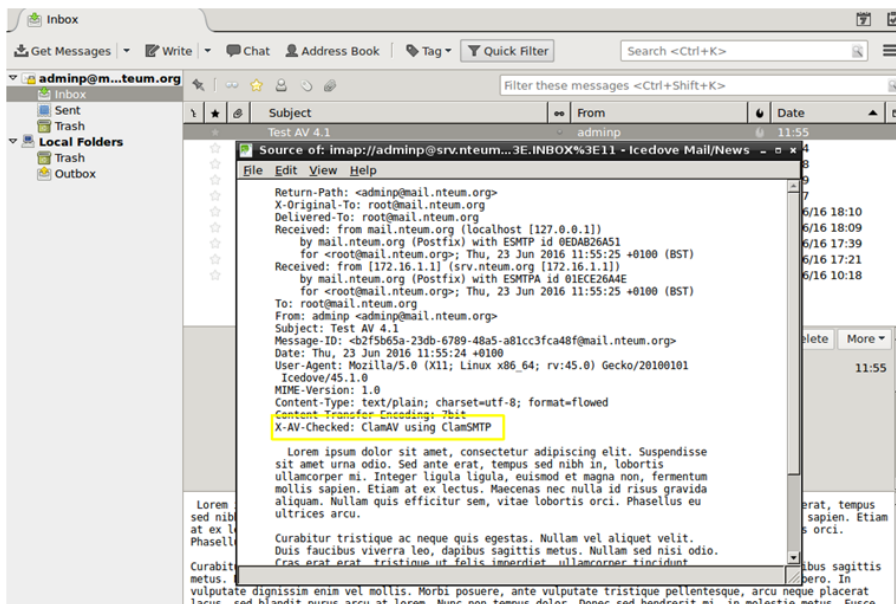
Reiniciem el sistema i enviem un correu per verificar que té el *header* tal com l'hem modificat (veure el requadre marcat en la figura 5).

8) L'últim paquet/servei essencial que ens falta instal·lar és el control de *spam* que es realitzarà mitjançant SpamAssassin. Per fer això s'ha d'executar:

```
apt-get install spamassassin spamc
vi /etc/postfix/master.cf modificar/afegir les línies 12, 14, 33

smtp inet n - - - smtpd
-o content_filter=spamassassin
submission inet n - - - smtpd
```

Figura 5



```

-o content_filter=spamassassin
smtps inet n - - - smtpd
-o syslog_name=postfix/smtps
-o smtpd_tls_wrappermode=yes
-o content_filter=spamassassin

```

Afegir al final de *master.cf* (tot en una línia):

```

spamassassin unix - n n - - pipe
user=debian-spamd argv=/usr/bin/spamc -f -e /usr/sbin/sendmail -oi -f ${sender} ${recipient}

```

```
vi /etc/spamassassin/local.cf modificar la línia 12
```

```
rewrite_header Subject *****SPAM*****
```

```
postfix reload per tornar a carregar la configuració de Postfix
```

Per assegurar-nos que el filtrat funciona*, es pot provar amb GTUBE (*Generic Test for Unsolicited Bulk Email* <http://spamassassin.apache.org/gtube/>) enviant-nos un correu que inclogui el següent *string*:

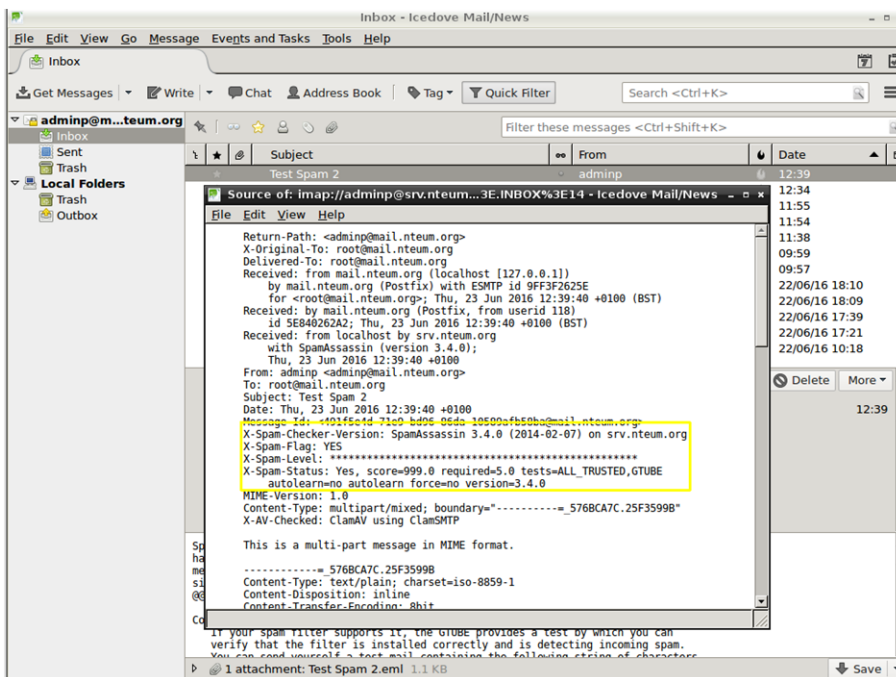
```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

A la figura 6 es pot veure el resultat en el quadrat marcat amb un *status*: *Yes* i un *score= 999.0* sobre un *required=5*. Amb això ja podem indicar al client que ha de fer amb l'Spam, si eliminar-lo o moure'l a la carpeta *Junk*.

9) Un aspecte important és el monitoratge del servidor i, entre els paquets útils per fer això, es poden instal·lar *pfllogsumm* (simple i en mode text), *MailGraph* (utilitza Apache i les llibreries *RRD*) o *AWstats* (també en Apache i la seva configuració es pot seguir des de [SerWorld])

*Podem enviar-nos un correu nosaltres mateixos, però només veurem a la capçalera que ha filtrat sense problemes.

Figura 6



Una opció interessant per al control de l'Spam i servidors dedicats amb un alt nombre d'incidències és Anti-Spam SMTP Proxy Server [ASSP]. L'ASSP Server és un projecte *Open Source* i independent de la plataforma, que permet llistes blanques autogenerades, autoaprenentatge mitjançant *Bayesian filters*, *Greylisting*, *DNSBL*, *DNSWL*, *URIBL*, *SPF*, *SRS*, *Backscatter*, filtrat de virus, bloqueig d'adjunts i altres característiques avançades.

1.3. Grups de discussió

Les *news* o grups de discussió són suportats mitjançant el protocol NNTP. Instal·lar un servidor de grups de discussió és necessari quan es desitja llegir *news* fora de línia, quan es vol tenir un repetidor dels servidors centrals o es vol un propi servidor mestre de *news*. Els servidors més comuns són INN o CNEWS, però són paquets complexos i destinats a grans servidors. Leafnode és un paquet USENET que implementa el servidor TNP, especialment indicat per a llocs amb grups reduïts d'usuaris, però on es desitja accedir a gran quantitat de grups de notícies. Aquest servidor s'instal·la en la configuració bàsica de Debian i es poden reconfigurar amb `dpkg-reconfigure leafnode` paràmetres com ara els servidors centrals, el tipus de connexió, etc. Aquest *daemon* s'engega des de `inetd` de manera similar al `imap` (o amb `xinetd`). Leafnode suporta filtres mitjançant expressions regulars indicades (del tipus `^Newsgroups:. * [,] alt.flame$`) en `/etc/news/leafnode/filters`, on per a cada missatge es compara la capçalera amb l'expressió regular i, si hi ha coincidència, el missatge es rebutja.

La configuració d'aquest servidor és simple i tots els arxius han de ser propietat d'un usuari de *news* amb permís d'escriptura (s'ha de verificar que aquest

Lectura recomanada

Per a detalls addicionals i altres aspectes interessants, es poden consultar les següents referències [SerMail], [WorkMail], [DebSpam], [ASSP].

propietari existeix en `/etc/passwd`). Tots els arxius de control, *news* i la configuració es troben en `/var/spool/news`, excepte la configuració del propi servidor que està en el fitxer `/etc/news/leafnode/config`. En la configuració, hi ha alguns paràmetres obligatoris que s'han de configurar (per exemple, perquè el servidor pugui connectar-se amb els servidors mestres), com són `server` (servidor de *news* des d'on s'obtiniran i enviaran les *news*) i `expire` (nombre de dies als quals un fil o sessió s'esborrarà després d'haver estat llegit). Tenim, així mateix, un conjunt de paràmetres opcionals d'àmbit general o específic del servidor que podrien configurar-se. Per a més informació, consulteu la documentació (`man leafnode 0 /usr/doc/leafnode/README.Debian`). Per a verificar el funcionament del servidor, es pot fer servir la instrucció `telnet localhost nntp i`, si tot funciona correctament, sortirà la identificació del servidor i es quedarà esperant una ordre. Com a prova, es pot introduir `help` (per a avortar, feu `Ctrl+[` i després `Quit`).

1.4. World Wide Web (httpd)

Apache és un dels servidors més populars i amb majors prestacions d'HTTP (*HyperText Transfer Protocol*). Apache té un disseny modular i suporta extensions dinàmiques de mòduls durant la seva execució. És molt configurable quant al nombre de servidors i de mòduls disponibles i suporta diversos mecanismes d'autenticació, control d'accés, *metafiles*, *proxy caching*, servidors virtuals, etc. Amb mòduls (inclosos en Debian) és possible tenir PHP3, Perl, Java Servlets, SSL i altres extensions*.

*Podeu consultar la documentació en <http://www.apache.org>.

Apache està dissenyat per a executar-se com un procés *daemon standalone*. En aquesta forma, crea un conjunt de processos fills que gestionaran les peticions d'entrada. També pot executar-se com *Internet daemon* mitjançant `inetd` o `xinetd`, per la qual cosa s'engegarà cada vegada que es rebi una petició, però no és recomanat. La configuració del servidor pot ser extremadament complexa segons les necessitats (consulteu la documentació); no obstant això, aquí veurem una configuració mínima acceptable. La seva instal·lació és simple, per exemple en Debian,

```
apt-get install apache2 apache2-doc apache2-utils
```

La configuració del servidor estarà en `/etc/apache2` i per defecte el *RootDirectory* en `/var/www/html`. Després de la seva instal·lació, s'engegarà i posant com URL en un navegador `http://localhost` veurem que funciona (ens mostrarà el famós **It works!**). Hi ha 5 ordres que hauran d'estar a la ment de tot administrador:

- `a2enmod|a2dismod` per a habilitar/deshabilitar mòduls,
- `a2ensite|a2dissite` per a habilitar/deshabilitar llocs (virtuals) i
- `apachectl` per a gestionar la configuració del servidor (`start|stop|restart|graceful|graceful-stop|configtest|status|fullstatus|help`).

Si bé tots els paràmetres tenen valors funcionals per defecte, és important tenir present que en la instal·lació per defecte no es troba definida la variable `ServerName` i que s'hauria de configurar a `/etc/apache2/apache2.conf` o en els arxius de configuració dels llocs dins del `tag` `Virtualhost` com `ServerName srv.nteum.org*`.

Vegeu exemples de *virtualhosts* en el subapartat següent.

La configuració d'Apache2 en Debian és una mica diferent de la distribució general ja que intenta facilitar al màxim la configuració del servidor quant a mòduls, *hosts* virtuals i directives de configuració (no obstant això, ràpidament es poden trobar les equivalències amb altres distribucions). Els principals arxius que es troben en el directori `/etc/apache2/` són `apache2.conf`, `ports.conf` i cinc directoris `mods-available|mods-enabled`, `sites-available|sites-enabled` i `conf.d`. Per a informació addicional, podeu llegir `/usr/share/doc/apache2.2*` i en particular `/usr/share/doc/apache2.2-common/README.Debian`.

- 1) `apache2.conf` és l'arxiu principal de configuració on es defineixen en un nivell funcional les prestacions del servidor i es criden els arxius de configuració corresponents (`ports`, `conf.d`, `sites-enabled`). Es recomana posar com a sufix `.load` per als mòduls que hagin de ser carregats i `.conf` per a les configuracions, però hi ha regles més extenses quant als sufixos/noms que poden ampliar-se en la documentació (p. ex., s'ignoren tots els arxius que no comencen per lletra o nom).
- 2) `ports.conf` (s'inclou en l'arxiu de configuració global) defineix els ports on s'atendran les connexions entrants, i quins d'aquests són utilitzats en els *hosts* virtuals.
- 3) Els arxius de configuració en `mods-enabled/` i `sites-enabled/` són per als llocs actius i els mòduls que desitgen ser carregats en el servidor. Aquestes configuracions s'activen creant un enllaç simbòlic des dels directoris respectius `*-available/` fent servir `a2enmod/a2dismod`, `a2ensite/a2dissite`.
- 4) Els arxius de `conf.d` són per a configuracions d'altres paquets o agregats per l'administrador i es recomana que acabin amb `.conf`.
- 5) Perquè sigui efectiva la configuració per defecte en aquests directoris, `apache2` ha de ser gestionat a través de `/etc/init.d/apache2` o `service` o `apache2ctl` (o també amb `systemctl`).
- 6) L'arxiu `envvars` és el que contindrà la definició de les variables d'entorn i és necessari modificar bàsicament dos `USER/GROUP` que seran amb les quals s'executarà i obtindrà els permisos. Per defecte es crea l'usuari `www-data` i el grup `www-data` (es poden canviar). Per aquest motiu, s'haurà de fer servir `APACHE_RUN_USER=www-data` i `APACHE_RUN_GROUP=www-data`.

Apache també pot necessitar integrar diversos mòduls en funció de la tecnologia que suporti, per la qual cosa s'hauran d'agregar les biblioteques/paquets corresponents, per exemple:

- 1) Perl: `apt-get install libapache2-mod-perl2`
- 2) Rugby: `apt-get install libapache2-mod-ruby`
- 3) Python: `apt-get install libapache2-mod-python`
- 4) MySQL in Python: `apt-get install python-mysqldb`
- 5) PHP: `apt-get install php5 php5-cgi libapache2-mod-php5 php5-common php-pear`
- 6) PHP with MySQL: `apt-get install php5-mysql`

1.4.1. Servidors virtuals

Per *servidors virtuals* s'entenen els llocs aïllats que seran servits cadascun de manera independent de l'altre amb els seus propis arxius i configuració. En primer lloc, deshabilitarem el lloc per defecte amb `a2dissite default`. Els llocs que crearem seran `remix.world` i `lucix.world`, que disposaran de dos arxius de configuració en `/etc/apache2/sites-available/` anomenats com el domini.

Contingut de l'arxiu `/etc/apache2/sites-available/remix.world.conf`

```
<VirtualHost *:80>
  ServerAdmin adminpSySDW.nteum.org
  ServerName remix.world
  ServerAlias www.remix.world
  DocumentRoot /var/www/remix/
  ErrorLog /var/log/apache2/remix-error.log
  CustomLog /var/log/apache2/remix-access.log combined
  Options ExecCGI # habilitar Script en Perl
  AddHandler cgi-script .pl
</VirtualHost>
```

Contingut de l'arxiu `/etc/apache2/sites-available/lucix.world.conf`

```
<VirtualHost *:80>
  ServerAdmin adminpSySDW.nteum.org
  ServerName lucix.world
  ServerAlias www.lucix.world
  DocumentRoot /var/www/lucix/
  ErrorLog /var/log/apache2/lucix-error.log
  CustomLog /var/log/apache2/lucix-access.log combined
  Options ExecCGI # habilitar Script en Perl
  AddHandler cgi-script .pl
</VirtualHost>
```

Aquesta configuració és molt bàsica i l'estudiant haurà de consultar la informació detallada en [apa]. Com es pot observar, els directoris arrel per a cada domini estaran en `/var/www/remix|lucix` i els arxius de `log` en `/errors/accessos` en `/var/log/apache2/mmmm-error.log` i `var/log/apache2/nnnn-access.log/`. Per a crear els directoris `mkdir -p /var/www/remix; mkdir -p /var/www/lucix` i en els quals es podria posar un `index.html` amb alguna identificació que mostres quin domini s'està carregant. Per exemple, per a `remix.world`:

```
<html><body><h1>REMIX: It works!</h1>
<p>This is the default web page for this server.</p>
```

```
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
```

I el mateix per a `lucix.world`, però canviant la línia en `<h1></h1>`. Per als *logs* no hem de fer res ja que el directori `/var/log/apache2` ja existeix i els arxius els crearà el servidor. Finalment, hem d'activar els llocs (crear l'enllaç des de *sites-available* a *sites-enabled*) amb `a2ensite remix.world.conf`; `a2ensite lucix.world.conf` i reiniciar `apache2` amb `service apache2 reload`. Com que no disposem dels dominis en un DNS primari, podem editar `/etc/hosts` i agregar per a la IP del nostre servidor (p. ex., `192.168.1.37`) dues línies:

```
192.168.1.37 remix.world
192.168.1.37 lucix.world
```

Després, des d'un navegador podrem introduir l'URL `remix.world` i el resultat serà la visualització de l'`index.html` que ens dirà: **REMIX: It works!**

Un dels avantatges d'Apache és que pot agregar funcionalitat mitjançant mòduls especialitzats i que es trobaran en `/etc/apache2/mods-available/`. Per a obtenir la llista de mòduls disponibles per a Apache podem fer, per exemple, `apt-cache search libapache2*`, i per a instal·lar-lo `apt-get install [module-name]`, els quals estaran disponibles per al seu ús (recordeu que pot ser necessària alguna configuració addicional en els arxius del lloc). Podem mirar els disponibles amb `ls -al /etc/apache2/mods-available/` i instal·lar-lo amb `a2enmod [module-name]`. Per a posar en una llista els mòduls carregats, podem fer servir `apachectl -M` que ens posarà en una llista amb *shared* els carregats dinàmicament i amb *static* els que es troben compilats amb el servidor (aquests es poden obtenir també amb `apache2 -l`). Els mòduls en el directori `mods-available` tenen extensions `.load` (indica la biblioteca que cal carregar) i `.conf` (configuració addicional del mòdul), però quan utilitzem l'ordre `a2enmod` només s'ha d'indicar el nom del mòdul sense extensió. Per a deshabilitar un mòdul, `a2dismod [module-name]`.

Com a mostra d'aquestes propietats, configurarem un lloc segur (`https`) sota el domini `remix.world` però que redirigirem al directori `/var/www/remix.ssl`. En primer lloc, crearem un certificat (autosignat) per al nostre lloc amb

```
make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/ssl/private/remix.crt
```

indicant-hi el domini que volem validar (`remix.world` en el nostre cas) –només cal introduir el domini i deixar els àlies en blanc– i si no podem executar `make-ssl-cert`, hem d'assegurar-nos que tenim el paquet `ssl-cert`. Després activem el mòdul SSL amb `a2enmod ssl`, creem el directori `/var/www/remix.ssl` i modifiquem l'`index.html` com vam fer amb els anteriors. Després, creem la configuració del lloc (podem utilitzar la que ve per defecte i modificar-la):

```
cd /etc/apache2/sites-available; cp default-ssl remix.world.ssl.conf
```

Editem l'arxiu *remix.world.ssl.conf* (només mostrem les línies principals/modificades):

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
  ServerAdmin adminpSySDW.nteum.org
  DocumentRoot /var/www/remix.ssl
  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>
  <Directory /var/www/remix.ssl>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>
# línies igual que l'arxiu original...
  ErrorLog $APACHE_LOG_DIR/remix.world.ssl_error.log
  CustomLog $APACHE_LOG_DIR/remix.world.ssl_access.log combined

  SSLEngine on
  SSLCertificateFile /etc/ssl/private/remix.crt
  #SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
# línies igual que l'arxiu original...
</VirtualHost>
</IfModule>
```

Finalment, ens queda activar el lloc (`a2ensite remix.world.ssl.conf`), reiniciar Apache2 (amb `service apache2 reload`) i des del navegador fer *https://remix.world* que, com que el certificat és autosignat, ens farà un advertiment i acceptarem el certificat i haurem d'obtenir **SSL - REMIX: It works!** També es poden configurar els certificats sense utilitzar l'*script* `make-ssl-cert` i utilitzar les ordres com es va fer en el subapartat de la instal·lació de Postfix i RoundCube.

Un aspecte interessant és la funció de l'arxiu *.htaccess** en els directoris del nostre domini. Aquest arxiu es pot utilitzar per al control d'accés al lloc (p. ex., habilitar/restringir IP), control d'accés a carpetes, llistes, reencaminaments (p. ex., a una altra pàgina/*site*, a una altra carpeta, a un altre domini, a https, etc.), evitar el *hotlinking* (per a evitar que ens facin enllaços a fitxers, generalment vídeos, que consumeixen amplada de banda del nostre servidor), canviar la pàgina per defecte, crear URL amigables, afavorir el cau del nostre lloc, etc. Com a mostra d'això, per a evitar per exemple que una carpeta sigui inaccessible, n'hi ha prou amb posar un arxiu *.htaccess* en la mateixa amb el contingut `deny from all`. Per a permetre que una carpeta del nostre lloc (p. ex., del domini *remix.com/valid-user*) tingui accés amb un usuari i passwd, haurem de crear dins d'aquesta un arxiu *.htaccess* amb el següent contingut (també podem crear un *index.html* modificat dins d'aquesta carpeta per a verificar-ne el funcionament):

```
AuthName "Restricted Area"
AuthType Basic
AuthUserFile /etc/apache2/htpasswd
AuthGroupFile /dev/null
require valid-user
```

*<http://httpd.apache.org/docs/2.2/howto/htaccess.htm>

Per a crear l'usuari, fem `htpasswd -c /etc/apache2/htpasswd adminp` que ens demanarà el *passwd* per a aquest usuari i l'emmagatzemarà en l'arxiu indicat. Després, posem com a URL `http://remix.world/valid-user/`. Ens demanarà l'usuari (adminp) i el *passwd* que emmagatzemem i veurem **REMIX->Valid-User: It works!** En cas contrari, ens continuarà demanant l'usuari */passwd* i si fem *Cancel* no indicarà un missatge d'*Authorization Required* impeding l'accés.

Per provar si PHP funciona després d'instal·lar-ho (vegeu el subapartat anterior) haurem de modificar el *Timezone* editant l'arxiu `vi /etc/php5/apache2/php.ini` i modificant `date.timezone = "Europe/Madrid"`. Després hem de reiniciar el servidor `systemctl restart apache2`.

Per provar el seu funcionament hem de crear una pàgina (fent servir la instrucció `vi /var/www/html/index.php`) amb el següent contingut:

```
<html>
<body>
<div style="width: 100%; font-size: 40px; font-weight: bold;
text-align:center;">
<?php
    print Date("Y/m/d");
?>
</div>
</body>
</html>
```

Cridant a la URL `http://srv.nteum.org/index.php` s'hauria de veure la data. En general funciona correctament però si presenta errors i no es visualitza la data, es poden revisar els *logs* d'Apache (`var/log/apache2/`) per a comprovar si els mòduls estan habilitats (`mods-enabled/ php5.conf`). També es pot forçar agregant un *handler* a `vi /etc/apache2/mods-enabled/mime.conf` i agregar `AddHandler php5-script .php`.

Ja hem estudiat l'autenticació d'usuaris a través del servidor; és molt útil per a l'accés d'usuaris que no formen part del sistema operatiu. Si es desitja habilitar a aquests usuaris és necessari vincular-los a Apache a través del sistema d'autenticació PAM. Per a això s'instal·larà

```
apt-get install libapache2-mod-authnz-external pwauth
```

A continuació haurem de crear un arxiu de configuració fent servir la instrucció `vi /etc/apache2/sites-available/ auth-pam.conf` amb el següent contingut:

```
AddExternalAuth pwauth /usr/sbin/pwauth
SetExternalAuthMethod pwauth pipe
<Directory /var/www/html/pam>
    AuthType Basic
```

```
AuthName "PAM Authentication"
AuthBasicProvider external
AuthExternal pwauth
require valid-user
</Directory>
```

Es crearà un directori `mkdir /var/www/html/pam` i dins es posarà un arxiu `index.html` amb un identificador que s'està accedint a PAM. Finalment, s'habilitarà el lloc `a2ensite auth-pam`, es reiniciarà el servidor `systemctl restart apache2` i es podrà accedir a `http://srv.nteum.org/pam/` després d'introduir un usuari local i el *passwd*. L'autenticació es realitza amb la integració del mòdul *authnz-external* i l'ordre `pwauth`.

1.4.2. Apache + PHP + Mysql + PhpMyAdmin

Una qüestió important per als servidors web dinàmics és aprofitar els avantatges d'Apache PHP i una base de dades com MySQL incloent un programa administrador de MySQL com PHPMyAdmin, tot això funcionant conjuntament. Les distribucions han evolucionat molt i en Debian és summament fàcil engegar aquest conjunt (però tampoc representa cap dificultat descarregar-se el programari font, compilar-lo i instal·lar-lo si es desitja, per exemple, tenir les últimes versions dels paquets per algun motiu però recordeu que implicarà més treball i dedicació).

En primer lloc, suposem que tenim PHP instal·lat i funcionant (vegeu el subapartat anterior). Com es pot observar quan s'instal·la PHP, es canvia del mode MPM-Worker a MPM-prefork. MPM-worker és un mòdul de multiprocessament que pot manejar múltiples peticions ràpidament utilitzant múltiples *threads* per procés client. No obstant això, aquest no és compatible amb algunes extensions PHP i per això l'*MPM-worker* és reemplaçat per *MPM-prefork*, que permet manejar totes les peticions PHP (en mode compatibilitat) i evitar que si una petició falla pugui afectar altres peticions. Hi ha un altre mòdul anomenat `mpm-itk` (<http://mpm-itk.sesse.net/>) que és similar a *prefork* però té millors prestacions i gestió de permisos (consulteu la bibliografia en apache.org). Per a verificar que PHP funciona, creem un fitxer per exemple dins de *RootDirectory* de `remix.world` anomenat `test.php` amb el següent contingut: `<?php phpinfo() ?>`, i si en l'URL introduïm `http://remix.world/test.php` haurem de veure una taula amb la versió i informació sobre el paquet PHP instal·lat.

Per instal·lar els paquets MySQL i PHPMyAdmin, farem

```
apt-get install mysql-server
```

(és molt important que recordeu la contrasenya d'accés que introduïm, però sempre podem fer `dpkg-reconfigure mysql-server`; tenint en compte que perdrem tot el que hi hagi en la BD). També hi ha altres mètodes (menys

agressius) per a recuperar la contrasenya del *root*). Després, per a instal·lar PHPMyAdmin farem `apt-get install phpmyadmin` i prestar atenció, ja que ens demanarà la clau d'accés per a entrar en la base de dades i crear una clau d'accés per a entrar en l'aplicació via navegador. Després, podrem posar en l'URL del nostre navegador `http://localhost/phpmyadmin`, ens sol·licitarà l'usuari (*root* generalment) i el *passwd* que hem introduït i ja podrem gestionar el servidor de bases de dades MySQL.

1.4.3. Altres servidors httpd

Lighttpd és un servidor web (amb llicència BSD) dissenyat per a ser ràpid, segur, flexible, que implementa la majoria d'estàndards i està optimitzat per a entorns on la velocitat és molt important (consumeix menys CPU/RAM que altres servidors) i és molt apropiat per a qualsevol servidor que hagi de donar suport a grans càrregues. Entre les seves principals característiques, hi ha la de *virtual hosting*, reencaminaments http i reescriptures d'URL, donar suport a CGI, SCGI i FastCGI, PHP, Ruby, Python entre d'altres i a més amb consum de memòria constant.

La seva instal·lació en Debian és `apt-get install lighttpd`, i si tenim Apache sobre el port 80 ens donarà un error. Per a això, haurem d'editar l'arxiu `/etc/lighttpd/lighttpd.conf` i canviar la línia `server.port = 8080` i reiniciar `service lighttpd start`. Des del navegador, es pot escriure l'adreça `http://localhost:8080index.lighttpd.html` i llavors veurem la pàgina inicial de lighttpd. Per defecte, Lighttpd té el seu directori arrel en `/var/www` (en Debian) i l'arxiu de configuració en `/etc/lighttpd/lighttpd.conf`. Configuracions addicionals són en `/etc/lighttpd/conf-available` i poden ser habilitades amb l'ordre `lighttpd-enable-mod`, la qual crea enllaços entre `conf-enabled` i `conf-available`. Es poden deshabilitar amb `lighttpd-disable-mod`.

Per a habilitar el servidor de FastCGI per a executar PHP, haurem d'instal·lar PHP-FPM amb la instrucció `apt-get install php5-fpm php5` i sobre l'arxiu `/etc/php5/fpm/php.ini` treure el comentari a la línia `cgi.fix_pathinfo=1`. Després haurem d'activar el servidor PHP-FPM, per la qual cosa farem una còpia de l'arxiu original i el modificarem:

```
cd /etc/lighttpd/conf-available/  
cp 15-fastcgi-php.conf 15-fastcgi-php-spawncgi.conf
```

Modificar `15-fastcgi-php.conf` amb:

```
# -- depends: fastcgi --  
  
# Start an FastCGI server for php  
fastcgi.server += ( ".php" =>  
    (
```

```

        "socket" => "/var/run/php5-fpm.sock",
        "broken-scriptfilename" => "enable"
    ))
)

```

Per a habilitar fastcgi, haurem de carregar els mòduls `lighttpd-enable-mod fastcgi` i `lighttpd-enable-mod fastcgi-php`, la qual cosa ens crea els enllaços corresponents, que amb la instrucció `ls` podem visualitzar: `ls -l /etc/lighttpd/conf-enabled`. Després, podem reiniciar amb la instrucció `service lighttpd force-reload`. Per a visualitzar si el servidor i FastCGI funcionen, creem un arxiu `/var/www/info.php` amb el següent contingut `<?php phpinfo(); ?>` i podem visualitzar la pàgina de configuració de PHP on indica com Server API = FPM/FastCGI (<http://localhost:8080/info.php>).

Un altre servidor molt utilitzat actualment és **Nginx** (<http://nginx.org/>) programat en C i llicència BSD. Les seves funcions principals són com a servidor web/*proxy* invers de molt alt rendiment (pot suportar més de 10.000 connexions simultànies) i també pot funcionar com a *proxy* per a protocols de correu electrònic (IMAP/POP3). És un servidor utilitzat per grans instal·lacions (WordPress, Netflix, Hulu, GitHub i parts de Facebook, entre d'altres) i entre les seves principals característiques hi ha (a més de servidor d'arxius estàtics, índexs i autoindexat i *proxy* invers amb opcions de cau) el balanç de càrrega, tolerància a fallades, SSL, FastCGI, servidors virtuals, *streaming* d'arxius (FLV i MP4.8) i suport per a autenticació, compatible amb IPv6 i SPDY. La seva instal·lació bàsica és simple, i per a la seva configuració bàsica* feu `apt-get install nginx`; després haureu d'executar `cd /var/www/html; mv index.nginx-debian.html index.html` i carregar des del navegador la URL <http://srv.nteum.org/>. Es veurà la pàgina bàsica d'inici de Nginx.

Enllaç d'interès

Una referència interessant (tot i que és una mica antiga, la major part de la configuració és útil) és <https://www.howtoforge.com/perfect-server-ubuntu-12.04-lts-nginx-bind-dovecot-ispconfig-3> on sobre Ubuntu s'instal·la i configura nginx, BIND, Dovecot per a la instal·lació d'ISPConfig 3. ISPConfig 3 és un panell de control que permet configurar diferents serveis a través d'un navegador (Apache or nginx, Postfix, Courier/Dovecot IMAP/POP3, MySQL, BIND/MyDNS, PureFTPd, SpamAssassin, ClamAV, entre d'altres).

1.4.4. Test de validació i prestacions d'Apache2

Una vegada instal·lat i configurat Apache2, es pot provar la creació d'algunes pàgines amb diferents elements i fer una validació funcional, però per a un administrador és molt important fer certes recerques sobre el programari instal·lat i veure com respon davant de càrregues intenses i diferents situacions/protocols, o simplement davant paquets TCP. Aquestes proves proporcionaran informació objectiva sobre l'adaptació del servei a l'entorn i la seva qualitat, de manera que permetran a l'administrador trobar els punts febles i/o disfuncions, i també, ajustar la gran quantitat de paràmetres de què disposa.

Enllaç d'interès

Per a una configuració detallada de Nginx podeu consultar la seva documentació a <http://nginx.org/>.

*Consulteu la wiki de nginx a <http://wiki.nginx.org/Configuration>

En primer lloc i de manera senzilla, es pot utilitzar **ApacheBench** [ABench], que és una eina d'avaluació comparativa per a servidors web. És una eina potent, instal·lada ja amb Apache2 dins del paquet `apache2-utils` (si no el tenim instal·lat, caldrà executar `apt-get install apache2-utils`) i fàcil d'utilitzar. Permet extreure resultats interessants sense experiència prèvia en plans de càrrega ni monitoratge de serveis. Un aspecte interessant és que, per no produir desviacions en el programari complementari d'anàlitiqes web, ApacheBench utilitza un *user agent* específic perquè pugui ser ignorat per la majoria de programaris d'anàlitiqes (tot i que, com es descriu en la documentació, pot haver-hi alguns casos en què les estadístiques es vegin afectades amb un determinat programari d'anàlítica Web). Una execució senzilla seria:

```
ab -n 10000 -c 100 http://srv.nteum.org/ no oblildeu posar la / final
```

On s'estan generant 10.000 trucades a `srv.nteum.org` distribuïdes en 100 *threads* (fils) per analitzar la capacitat de concurrència i comprovar situacions de bloquejos o condicions de carrera que puguin deixar inutilitzat el servidor. Els resultats obtinguts són (mostrem aquí les línies més rellevants):

```
This is ApacheBench, Version 2.3 <$Revision: 1604373 $>
...
Document Path:          /
Document Length:       280 bytes
Concurrency Level:     100
Time taken for tests:   1.695 seconds
Complete requests:     10000
Failed requests:       0
Total transferred:     5510000 bytes
HTML transferred:     2800000 bytes
Requests per second:   5898.34 [# /sec] (mean)
Time per request:      16.954 [ms] (mean)
Time per request:      0.170 [ms] (mean, across all concurrent requests)
Transfer rate:         3173.82 [Kbytes/sec] received
```

```
Connection Times (ms)
                min          mean[+/-sd]    median      max
Connect:        0           0  0.2         0         3
Processing:     1          17  6.7        15        62
Waiting:        1          15  6.7        14        62
Total:          2          17  6.7        16        63
```

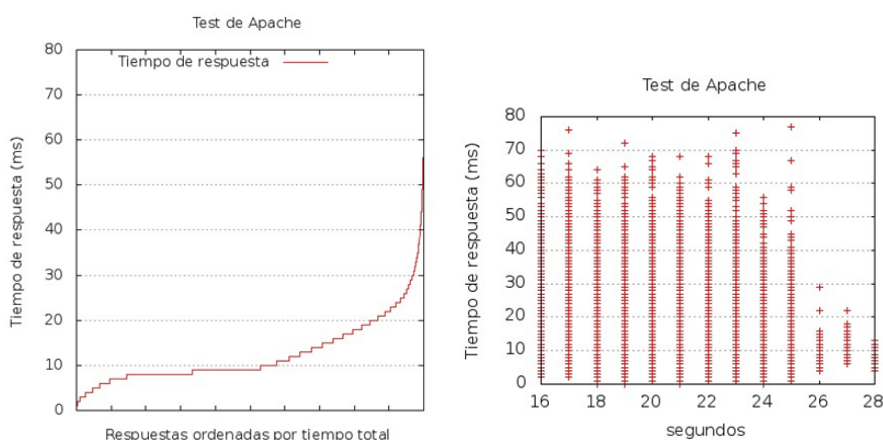
```
Percentage of the requests served within a certain time (ms)
```

```
 50%    16
 66%    18
 75%    19
 80%    21
 90%    25
 95%    29
 98%    36
 99%    42
100%    63 (longest request)
```

Els valors més interessants són el *Requests per second* (peticions ateses per segon), *Time per request* (temps mitjà a l'hora d'atendre un grup de peticions concurrents) i *Time per request* (temps mitjà per atendre una petició individual) i els valors mínims, mitjana, moda i màxims, i si el servidor ha pogut servir totes les peticions o no (*Failed requests*). Si agreguem el paràmetre `-g /tmp/output.txt` podem generar les dades en format *gnuplot* per després visualitzar-les. Per fer això, és interessant utilitzar *scripts* com els de

[BranScripts] per obtenir, en primer lloc, el rang de temps de les peticions, entre les que triguen menys i les que triguen més (cal anar amb compte amb la gràfica esquerra de la figura 7 perquè no està ordenada per l'ordre de les peticions, sinó del valor de *ttime* –total time–). La gràfica de la dreta de la figura 7 conté les mateixes dades, però, per a cada segon de la prova (en total 12 segons), s'indica la distribució del temps de resposta on es pot observar la densitat i el màxim/mínim de cada petició (les dades van ser obtingudes en local posant en el servidor una pàgina de certa complexitat i fent 100.000 peticions en 100 *threads* concurrents i transferint 19 Gi-gaoctets amb la instrucció `ab -n 100000 -c 100 -g /tmp/output.txt http://srv.nteum.org/test.htm`).

Figura 7



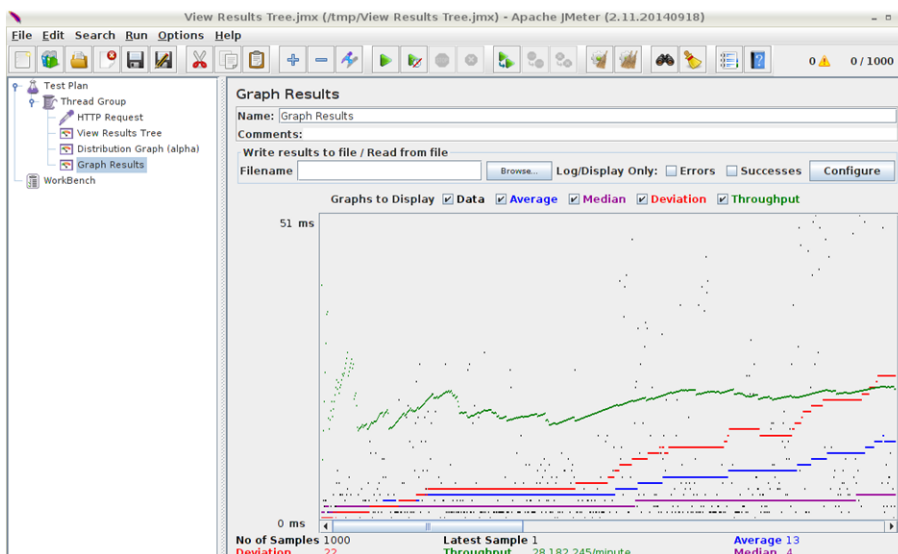
Podem extreure com a conclusió que, en aquesta configuració d'Apache, el servidor respon molt bé a una gran quantitat de transaccions concurrents i temps de resposta molt acceptables. Aquesta eina admet una gran quantitat de paràmetres [ABench], entre d'altres, als quals puguem passar una *cookie* de sessió o per avaluar pàgines que tinguin usuari i *passwd* (o simplement els valors, si l'autenticació de bàsica), protocols que cal analitzar o la utilització de proxy.

Una eina més sofisticada i adequada per realitzar anàlisis complexes i detallades, sota diferents tipus de càrrega, és **JMeter** [Jmet]. Aquesta eina està escrita en Java i inicialment va ser dissenyada per mesurar les prestacions d'aplicacions web, però es pot estendre a altres tipus de prova i és molt potent per provar-ne el rendiment tant en recursos estàtics com en dinàmics (serveis web –SOAP/REST–, web amb diferents llenguatges –PHP, Java, ASP.NET, Files, etc.–, objectes Java, consultes a bases de dades, servidors FTP, etc.). Es pot utilitzar per simular una càrrega pesada concurrent en un servidor o grup de servidors, en la xarxa o en un objecte, analitzar-ne el rendiment general sota diferents perfils i realitzar una anàlisi gràfica del rendiment. La seva instal·lació en Debian és molt simple `apt-get install jmeter` i per executar-lo fem simplement `jmeter`. Quan s'inicia, es mostren dues possibilitats de treball *Test Plan*,

que permetran configurar i desar el test que s'ha de realitzar, i un espai temporal de treball (*Workbench*) on ens permetrà provar i utilitzar opcions que no formaran part, però, del test de treball. La manera de treballar és molt simple, ja que sobre l'opció de *Test Pla* i el botó dret es poden agregar diferents proves (per exemple, *Add->Thread Users-> Thread Group*) i, després, configurar-lo. És necessari salvar el *Test Pla* des del menú principal, però també es pot salvar la configuració d'un determinat element (*Save Selection*) i després carregar tan sols aquesta part (*Merge*). Sempre que se seleccioni un element en la pantalla dreta, es podran veure/modificar les configuracions específiques d'aquest objecte.

A la figura 8 es mostra la distribució del temps de resposta en una prova contra el nostre servidor (srv.nteum.org) amb 1000 *threads*. La seva configuració és molt simple: s'ha agregat al *Test Plan* un *Thread Group* i a aquest, un *Sampler=HTTP Request* (configurat per fer la petició a la IP del servidor) i després, tres *Listeners* per recollir els resultats (només es mostren els resultats del tercer Listener *Graph Results*).

Figura 8



Lectura recomanada

Existeix una gran quantitat de tutorials [TutPoint] [GuTut] i és també molt útil el manual d'usuari [JmUM] per obtenir informació sobre tots els aspectes d'utilització i prova per a diferents llenguatges, bases de dades, serveis, etc.

Finalment, per completar l'anàlisi d'un servidor Web és necessari comptar amb eines que ens permetin analitzar els registres de connexió per saber tota la informació que rep/proveeix el nostre servidor. Dues de les eines més comunes per a aquesta finalitat són AWStats [AW] i Webalizer [WA] (o un projecte derivat d'aquest AWFFull [AWFull]); a continuació, veurem alguns detalls i la instal·lació de la primera eina.

AWStats és una de les eines que permeten fer anàlisis i obtenir estadístiques d'un servidor web presentant informes detallats de taules i gràfics de barra. La manera habitual d'accedir-hi és a través del mateix servidor web protegit amb usuari i contrasenya (Apache en el nostre cas) per visualitzar les estadístiques generades periòdicament a través del servei *cron*. Un dels principals avantatges

és que és molt versàtil, ja que suporta la majoria dels formats d'arxius *log* de servidor web coneguts (Apache, WebStar, IIS ...) i incorpora *plugins* per llegir altres tipus de *logs* (per exemple, els d'un servidor de correu –per a la seva configuració, consulteu [SerWorld]–).

Per instal·lar-lo en una configuració mínima farem el següent:

1) `apt-get install awstats perl` i posteriorment s'haurà d'editar l'arxiu `/etc/awstats/awstats.conf` i editar/verificar les següents línies:

```
# Log d'Apache
LogFile="/var/log/apache2/access.log"
# Format per a Apache (combined logs)
LogFormat=1
# domini utilitzat per al lloc
SiteDomain="srv.nteum.org "
# Àlies i directoris d'Icons
HostAliases="localhost 127.0.0.1"
DirIcons="./icon"
```

2) Crear el directori dins del *DocumentRoot* d'Apache:

```
mkdir /var/www/html /awstats
```

3) Fer un enllaç al directori d'Icons:

```
ln -s /usr/share/awstats/icon /var/www/html/awstats/icon
```

4) Crear l'arxiu de configuració per a Apache:

```
vi /etc/apache2/sites-available/awstats.conf
```

```
<VirtualHost awstats.nteum.org:80>
  ServerAdmin webmaster@example.com
  ServerName awstats.nteum.org
  DocumentRoot /var/www/html/awstats
  <Directory /var/www/html/awstats>
    #AuthGroupFile /dev/null
    AuthType Basic
    AuthUserFile /var/www/.htpasswd
    AuthName "Access Restricted"
    Require valid-user
    AuthType Basic
    Order deny,allow
    Deny from all
    Allow from 172.16.1.0/16
  </Directory>

  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order deny,allow
    Deny from all
    Allow from 172.16.1.0/16
  </Directory>
</VirtualHost>
```

5) Crear un usuari i *passwd*: `htpasswd -c /var/www/.htpasswd adminp`

6) Habilitar el lloc (`a2ensite awstats.conf`) i verificar la configuració

```
apachectl configtest
```

7) Reiniciar Apache: `systemctl restart apache2`

8) Generar el contingut i els reports:

```
/usr/lib/cgi-bin/awstats.pl -config=apache -update
/usr/lib/cgi-bin/awstats.pl -config=apache -output -staticlink
> /var/www/html/awstats/index.html
```

9) Accedir a <http://awstats.nteum.org> introduint l'usuari i *passwd* generats per visualitzar el contingut.

10) Editar el *cron* perquè s'actualitzin les dades:

`crontab -l` i inserint:

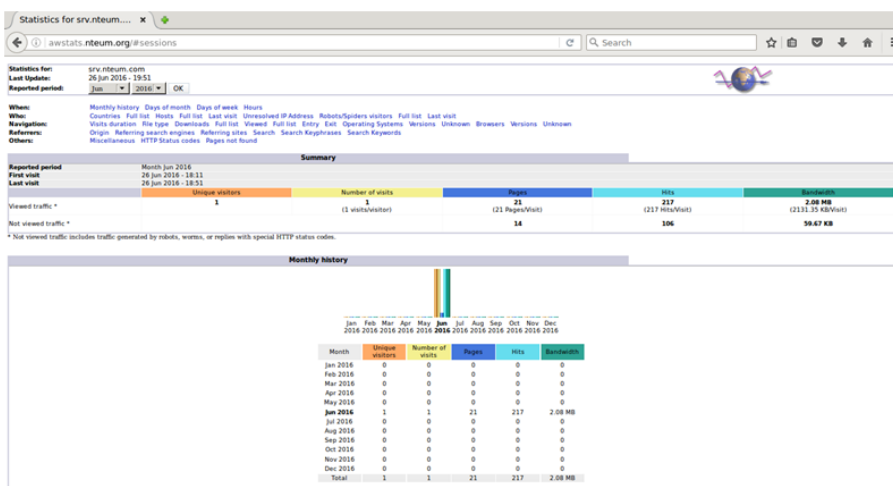
```
0 * * * * /usr/lib/cgi-bin/awstats.pl -config=apache -update
5 * * * * /usr/lib/cgi-bin/awstats.pl -config=apache -output -staticlink
> /var/www/html/awstats/index.html
```

És important tenir en compte que per a algunes opcions hem de tenir l'execució de *script perl* activada, per la qual cosa s'haurà de fer

```
vi /etc/apache2/mods-enabled/mime.conf
```

i en la línia 219 es traurà el comentari i s'agregarà l'extensió `.pl` *AddHandler cgi-script .cgi .pl*. També s'ha d'habilitar el mòdul `a2enmod cgid` i reiniciar el servidor `systemctl restart apache2`. La figura 9 mostra una vista (parcial) de l'eina.

Figura 9



1.5. Servidor de WebDAV

El nom WebDAV són les sigles de *Web Based Distributed Authoring and Versioning* (també es refereix al grup de treball d'Internet Engineering Task Force) i és un protocol que permet que el web es transformi en un mitjà llegible i

editable i proporciona funcionalitats per a crear, canviar i moure documents en un servidor remot (típicament, un servidor web). Això s'utilitza sobretot per a permetre l'edició dels documents que envia un servidor web, però també es pot aplicar a sistemes d'emmagatzematge generals basats en el web i als quals es pot accedir des de qualsevol lloc. En aquest subapartat, instal·larem un servidor WebDAV sobre Apache. El procés és el següent:

1) Verificar que tenim instal·lat Apache2 i, si no, fer la seva instal·lació com hem vist anteriorment i verificar que funciona (`apt-get install apache2`).

2) Habilitar els mòduls d'Apache que són necessaris per a WebDAV:

```
a2enmod dav_fs i a2enmod dav.
```

3) Crear el directori per al directori virtual (podem fer servir, per exemple, `mkdir -p /var/www/webdav`) i permetre que Apache sigui el propietari del directori `chown www-data /var/www/webdav/`

4) Crear l'arxiu `/etc/apache2/sites-available/webdav.conf` per a definir la funcionalitat del servidor (en aquesta configuració, estem configurant tot el servidor com a WebDAV però podria ser un servidor virtual i en mode SSL per a major seguretat):

```
<VirtualHost *:80>
  ServerAdmin admin@SySDW.nteum.org
  DocumentRoot /var/www/webdav/
  ErrorLog /var/log/apache2/webdav-error.log
  CustomLog /var/log/apache2/webdav-access.log combined
  <Directory /var/www/webdav>
    Options Indexes MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
    DAV On
    AuthName "Restricted WeDav Area"
    AuthType Basic
    AuthUserFile /etc/apache2/htpasswd
    AuthGroupFile /dev/null
    require valid-user
  </Directory>
</VirtualHost>
```

Es pot comprovar que la configuració és correcta amb

```
apache2ctl configtest.
```

A continuació, s'habilita el lloc amb `a2ensite webdav`.

5) Com s'ha fet anteriorment, es creen els usuaris indicant el `-c` si és el primer usuari que cal crear (`htpasswd [-c] /etc/apache2/htpasswd usuari`).

6) Es reinicia Apache perquè llegeixi la configuració `/etc/init.d/apache2 reload` (o amb `service restart apache2` o amb `apache2ctl restart`) i ja ens podem connectar a `http://localhost` (o `http://srv.nteum.org/`), després de fer l'autenticació.

Enllaç d'interès

Sobre la integració de WebDAV amb Apache, podeu consultar l'article "WebDAV on Apache2" disponible en:
<http://www.debian-administration.org/articles/285>

7) Des d'un GNU/Linux, podem provar la funcionalitat del servidor obrint el `nautilus -o` equivalent, p. ex. `PCManFM` (gestor de fitxers) i des del menú `File->Connect to Server` podem seleccionar `Servidor WebDAV` introduint les dades (IP, directori, usuari, passwd) i ja tindrem accés com si d'una carpeta local es tractés.

8) Des de MacOS, podem utilitzar el mateix procediment que l'anterior des del gestor d'arxius o instal·lar un client específic (igualment per a Windows). El més recomanat per a això és `CyberDuck` (<http://cyberduck.io/>), que té llicència GPL i és una excel·lent aplicació (suporta múltiples protocols) i molt fàcil de configurar.

9) Una altra forma de provar-ho és amb un client WebDAV (en mode text), per exemple `Cadaver*`, amb `apt-get install cadaver`. A continuació, ens connectem al servidor amb `cadaver IP-nom del servidor`, i després d'autenticar-nos, podem crear un directori (`mkdir`), editar un arxiu, fer la llista d'un directori (`ls`), canviar de directori (`cd`), canviar els permisos d'execució (`chexec`), esborrar-lo (`rm`), etc.

*<http://www.webdav.org/cadaver>

En moltes ocasions, i atès que estarem fent transferències d'arxius, és important preservar la privadesa, per la qual cosa seria adequat treballar amb WebDAV però sobre SSL. La seva configuració no implica majors complicacions i veurem una manera diferent de generar els certificats (seran autosignats, fins que puguem tenir la nostra pròpia entitat certificadora) per a un domini en particular, en el nostre cas `webdav.nteum.org`. Per a això fem:

1) Ens canviem al directori on emmagatzemarem els certificats:

```
cd /etc/ssl/private
```

Fem la petició del certificat:

```
openssl req -config /etc/ssl/openssl.cnf -new -out webdav.csr
```

Aquesta ordre ens demanarà un `passwd` i una sèrie d'informació que quedarà en el certificat, però la més important és `Common Name (CN)`, que serà on validarà el certificat (en el nostre cas, `webdav.nteum.org`).

Podem verificar la petició amb:

```
openssl req -in /etc/ssl/private/webdav.csr -noout -text
```

2) Creem la clau (`key`):

```
openssl rsa -in privkey.pem -out webdav.key
```

3) Signem:

```
openssl x509 -in webdav.csr -out webdav.crt -req -signkey webdav.key  
-days 3650
```

Ho podem verificar amb:

```
openssl x509 -noout -in /etc/ssl/private/webdav.crt -text
```

4) Generem un certificat en format DER:

```
openssl x509 -in webdav.crt -out webdav.der.crt -outform DER
```

Es pot verificar amb:

```
openssl x509 -in /etc/ssl/private/webdav.der.crt -inform der -noout -text
```

5) Ara, generem l'arxiu de configuració d'Apache a partir de l'anterior: `cd`

```
/etc/apache2/sites-available; cp webdav.conf webdav-ssl.conf
```

Modifiquem per a incloure les següents quatre línies a l'inici i modificar el `VirtualHost`:

```
<VirtualHost *:443>
ServerName webdav.nteum.org
SSLEngine on
SSLCertificateFile /etc/ssl/private/webdav.crt
SSLCertificateKeyFile /etc/ssl/private/webdav.key
```

...

Només ens queda activar el lloc (`a2ensite webdav-ssl.conf`), reiniciar Apache (`service apache2 restart`), afegir una línia a l'arxiu `/etc/hosts` amb `172.16.1.1 webdav.nteum.org webdav`, i finalment verificar que funciona en l'adreça `https://webdav.nteum.org` prèvia acceptació del certificat (recordar posar una entrada en `/etc/hosts` amb el nom del domini i l'IP de la màquina similar a com es va fer en `remix.world`).

1.6. Proxies

La funció d'un *proxy* és jugar el paper d'intermediari en les peticions que sol·licita el client a un altre servidor, és a dir, el servidor *proxy* coneix tots dos recursos i els posa en contacte sense que un conegui l'altre. Com que actua de punt d'unió entre les peticions i els serveis, permet dur a terme diferents accions com ara control d'accés, registre/control del trànsit (inclòs bloqueig), millora del rendiment de la transacció (emmagatzematge intermedi), anonimat en la comunicació, entre d'altres. Atès que actua com a intermediari, existeixen diverses opinions/controvèrsies sobre la utilització de *proxies* quant a la seguretat i anonimat. És per això que és necessari cuidar bé la seva configuració i el servei que presta perquè no sigui possible utilitzar-lo amb una altra finalitat que no sigui aquella per la qual ha estat concebut.

Si bé es poden trobar diferents tipus de *proxies*, normalment diferenciats pel protocol/aplicació que gestionen (web, ftp, ARP, dns...), aquell que s'utilitza en serveis web és probablement el més habitual. Entre els avantatges d'un *proxy* web, es poden enumerar: el control de trànsit, la velocitat (*proxy cache*) i el filtrat a nivell d'aplicació/protocol, i entre els desavantatges, l'anonimat/abús (el client mai és responsable de la petició del servei), càrrega i coll d'ampolla, pas addicional en la comunicació entre client i servidor, i irregularitat (temps de resposta variable en funció de la càrrega del *proxy*).

En funció del rol que compleixin, tenim diferents definicions de *proxies*:

- 1) Un servidor *proxy* que passa les peticions i les respostes no modificades generalment s'anomena **porta d'enllaç** (*gateway* o *tunneling proxy*).
- 2) Un *proxy forward* és un servidor que connecta Internet amb clients interns que realitzen peticions a recursos externs. Generalment es combina amb un *proxy cache* per accelerar l'accés als recursos, ja que només el primer usuari que els sol·licita és qui accedeix al recurs, i els subsegüents accedeixen a la còpia en el servidor *proxy*.
- 3) Un *reverse proxy* (invers) és, generalment, un servidor que rep les peticions d'Internet i les deriva a servidors interns (per exemple, en una xarxa privada) protegint-los i permetent fer un balanç de la càrrega entre diversos servidors.

Si el *proxy* està connectat des de i cap a Internet, es considera un *proxy* obert (*open proxy*) i la seva funció és reexpedir tots els paquets que rep permetent ocultar la IP del client al servidor, la qual cosa és una forma d'anonimat (feble). Existeixen llistes extenses d'*open proxies* (n'hi ha prou amb fer la consulta <https://www.google.es/search?q=open+proxies+list>), però ningú pot donar fe de l'anonimat que permeten. Si es desitja tenir anonimat real, s'han d'utilitzar xarxes com ara Tor (*The Onion Router*), que permet tenir garanties d'anonimat utilitzant comunicacions encriptades (diverses vegades) i que passen a través d'una xarxa mundial de servidors (voluntaris), de manera que permeten obtenir l'anonimat de la comunicació i impedeixen que aquesta pugui ser vigilada o supervisada [Tor]. Una altra xarxa similar que permet l'anonimat és la xarxa I2P [I2P] del projecte *Invisible Internet Project* amb objectius similars a Tor, però que no està tan difosa.

Una pregunta freqüent és la diferència entre un *proxy* i un *firewall* (actuant com NAT). Generalment, quan s'especifica *proxy*, ens referim a una aplicació de capa 7 del model OSI, mentre que NAT es refereix a la capa 3 d'aquest model. En la configuració d'un client en capa 3 (NAT) només s'ha de conèixer la porta d'enllaç (*gateway*) i aquest (normalment denominat *router*) realitzarà la translació, mentre que en la configuració del client en capa 7 s'han d'enviar els paquets al servidor *proxy*, que llegirà cada paquet per conèixer la seva destinació i reenviar-lo.

Atès que NAT opera en capa 3, utilitza menys recursos, però també és menys flexible que en capa 7, ja que tan sols actua sobre les adreces de paquet i no sobre el contingut, com fa el *proxy* (en capa 7). Sovint, en els sistemes GNU/Linux el NAT es realitza amb IPTables (*firewall*), mentre que, com a *proxy*, s'utilitzen diferents servidors; per exemple, per a http/https/ftp s'utilitzen Apache, Squid, Nginx i Varnishm, entre d'altres.

1.6.1. Apache com a *reverse proxy* i amb balanç de càrrega

Apache és un servidor http molt versàtil i eficient que posseeix una àmplia té de mòduls que estenen la seva funcionalitat, entre elles, la de *proxy*. En aquest subapartat analitzarem la configuració primer, en un servidor intern com a *reverse proxy*. Després, estudiarem com fer un balanç de càrrega en més d'un servidor, redirigint les peticions en funció de diferents polítiques. Apache suporta diferents mòduls [AMod]. Per a *proxy* tenim `mod_proxy` (*proxy* multiprotocol), `mod_http` (suport http per a *proxy*), `mod_cache`, `mod_proxy_balancer` (balanç per a *reverse proxy*), `mod_proxy_html` (reescriptura dels enllaços HTML per assegurar-se que funcionen fora del *proxy*). Comencem:

1) En primer lloc, instal·lem el paquet amb els mòduls corresponents:

```
apt-get install libapache2-mod-proxy-html
```

2) Habilitem els mòduls: `a2enmod proxy proxy_http` (verifiquem amb l'ordre `apachectl -M`).

3) Generem un *host* nou a `/etc/hosts` (per exemple, `172.16.1.1 proxy.nteum.org proxy`).

4) Creguem *virtualhost* a `/etc/apache2/sites-avalabile/proxyr.conf`:

```
<VirtualHost proxy.nteum.org:80>
  ErrorLog "/var/log/apache2/proxy-error.log"
  CustomLog "/var/log/apache2/proxy-access.log" common
  ServerName proxy.nteum.org
  ProxyRequests Off
  ProxyPreserveHost On
  ProxyPass / http://ubub.nteum.org/
  ProxyPassReverse / http://ubub.nteum.org/
</VirtualHost>
```

On

- `ServerName` ha d'estar definit al `/etc/hosts`, i indica com anomenarem el servidor d'entrada,
- `ProxyRequests Off` evita que sigui utilitzat com a *open proxy*, és a dir, que els usuaris puguin anar al *proxy* i, d'aquí, a qualsevol altra adreça (per la qual cosa, en tot el que facin constarà la nostra IP) i és molt important deixar-lo deshabilitat per evitar problemes de seguretat o fins i tot legals,
- `ProxyPreserveHost On` permet que el salt del servidor de *proxy* al de *backend* sigui transparent per a l'usuari (si no estigués habilitada, l'usuari es dirigiria a `http://proxy.nteum.org`, però immediatament veuria com l'adreça canvia a `http://ubub.nteum.org`, que és el servidor intern *-backend-*) i a més, si el servidor de *backend* no és visible des d'Internet, el client veu un error,
- `ProxyPass` i `ProxyPassReverse` gestionen el salt i el retorn del servidor de *frontend* al de *backend*.

5) Habilitem la configuració del nou lloc (`a2ensite proxyr`) i reiniciem el servei (`systemctl restart apache2`). També hem de tenir en el *backend* un servidor `apache2` funcionant en una pàgina diferent de la del *proxy* per verificar que s’hi accedeix quan posem en un navegador `http://proxy.nteum.org`.

Un dels aspectes interessants en el servei web és poder realitzar un balanç de càrrega de les peticions sobre diferents servidors per evitar l’efecte “coll d’ampolla” en el servei i millorar el temps de resposta i incrementar el nombre de peticions ateses per unitat de temps. Això es pot fer mitjançant maquinari específic o mitjançant un *reverse proxy (frontend)* que distribueixi la càrrega a una granja interna de servidors (*backend*) d’acord amb una política determinada. Apache disposa d’un mòdul addicional al *proxy* (`mod_proxy_balance`) que permet realitzar el balanç de càrrega sobre un conjunt de servidors web i diferents mòduls per implementar les polítiques (`lbmethod_byrequests` `lbmethod_bytraffic` `lbmethod_bybusyness` `lbmethod_heartbeat`).

Per configurar aquest mòdul haurem de fer el següent:

1) Carregar els mòduls:

```
proxy, proxy_balancer proxy_connect proxy_html proxy_http lbmethod_byrequests
lbmethod_bytraffic lbmethod_bybusyness lbmethod_heartbeat status
```

(per veure els mòduls carregats `apachectl -M`)

2) Crear un *virtualhost*: `vi /etc/apache2/sites-available/proxy-bal.conf`

```
<VirtualHost proxy.nteum.org:80>
ProxyRequests off
ServerName proxy.nteum.org
DocumentRoot /var/www
<Proxy balancer://mycluster>
    BalancerMember http://172.16.1.2:80
    BalancerMember http://172.16.1.3:80
    Options Indexes FollowSymlinks Multiviews
    AllowOverride None
    Order Allow, Deny
    Allow from all
    ProxySet lbmethod=bytraffic
    #ProxySet lbmethod=byrequests
</Proxy>
# Habilitar el Balancer Manager
<Location /balancer-manager>
    SetHandler balancer-manager
    Order deny,allow
    Allow from all
</Location>
ProxyPass / balancer-manager !
ProxyPass / balancer://mycluster/
ProxyPassReverse / balancer://mycluster

ProxyPass / http://cloneuno.nteum.org
ProxyPassReverse / http://cloneuno.nteum.org
ProxyPass / http://clonedos.nteum.org
ProxyPassReverse / http://clonedos.nteum.org
</VirtualHost>
```

Haurem de tenir en `/etc/hosts` les màquines a les quals es redirigiran les peticions (en el nostre cas 2, `cloneuno` i `clonedos`):

```
172.16.1.1    proxy.nteum.org proxy
172.16.1.2    cloneuno.nteum.org cloneuno
172.16.1.3    clonados.nteum.org clonados
```

El *balancer manager* és una eina que integra el mòdul i que permetrà veure de manera senzilla les estadístiques simples de l'activitat del mòdul i algunes modificacions (simples també). És per això que les peticions a l'adreça <http://proxy.nteum.org/balancer-manager> no s'hauran de redirigir ni ser ateses pel *proxy*.

La configuració inclou els següents elements:

- La secció *Proxy balancer*: on s'identifica el balancejador.
- *BalancerMember*: cadascuna de les IP del *backend*
- *ProxySet lbmethod=byrequests|bytraffic*: la política de balanç

3) Habilitar la configuració del nou lloc (`a2ensite proxy-bal`) i reiniciar el servei (`systemctl restart apache2`). També cal tenir en el *backend* els dos servidors (*cloneuno* i *clonados*) `apache2` funcionant en una pàgina diferent de la del *proxy*, per verificar que s'hi accedeix quan posem en un navegador <http://proxy.nteum.org>, i recarreguem la pàgina repetidament (veurem com va canviant la pàgina en funció del servidor del *backend* que la serveix). Per obtenir més informació, es poden veure les estadístiques del balancejador a <http://proxy.nteum.org/balancer-manager> i canviar els paràmetres per adequar-los a les necessitats de càrrega (es poden utilitzar les eines de càrrega esmentades per analitzar Apache).

Lectura recomanada

Per a més informació, es pot consultar la documentació del mòdul [AModBal]

1.6.2. Apache com a *Forward Proxy* i *Proxy cache*

Per configurar Apache com a *Forward Proxy* s'haurà de fer el següent:

1) Carregar el mòdul (si no està carregat, ho verifiquem amb `apachectl -M`):

```
a2enmod proxy proxy_http
```

2) Agregar en la configuració d'un lloc:

```
vi /etc/apache2/sites-available/proxy-f.conf
```

```
Listen 172.16.1.1:8080          #On sentirà les peticions
<VirtualHost 172.16.1.1:8080>
ProxyRequests On              #Activa el Proxy Forward
<Proxy *>                      #També pot ser <Directory> ... </Directory>
    Order deny,allow          #Regles del servei
    Deny from all
    Allow from 172.16.1.0/24   # clients habilitats 172.16.1.*
</Proxy>
ProxyBlock marca.es as.es     #bloqueja l'accés a aquests dos llocs
```

3) Habilitar el lloc (`a2ensite proxy-f`) i reiniciar el servei amb la instrucció `systemctl restart apache2`. A continuació, s'han de modificar els clients perquè sol·licitin les peticions al servidor *proxy*, per exemple, a Firefox cal anar a *Options->Advanced->Network->Settings* i a *Proxy*, posar la IP 172.16.1.1 i el port 8080. Per evitar un error comú quan els usuaris no configuren el port

específic, es pot posar `iptables` amb una regla que redirigeixi al port 8080 tot el que ve al port 80 (o on es trobi el `proxy forward`).

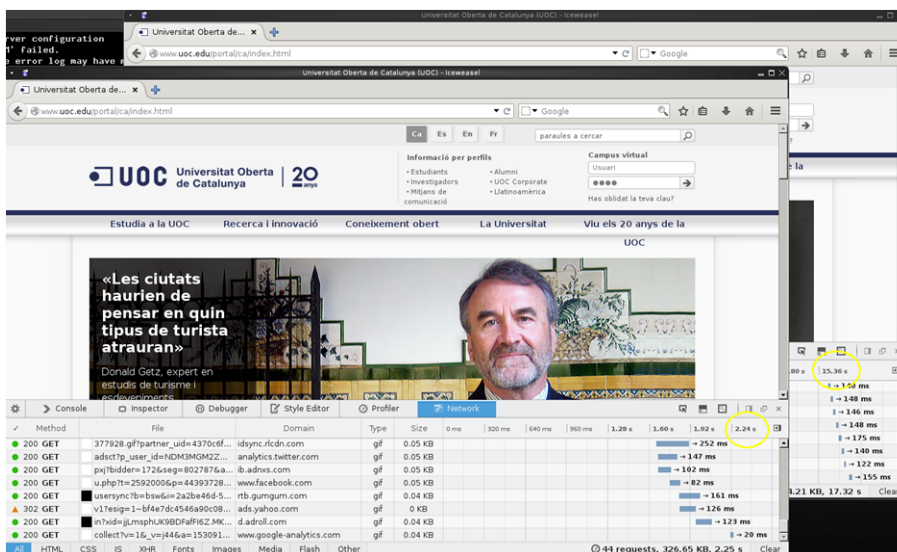
4) Per activar la capacitat de `cache` del `proxy` s'hauria d'afegir al `virtualhost`:

```
<IfModule mod_disk_cache.c>
    CacheRoot "/usr/local/apache/proxy"
    #CacheSize 500
    CacheEnable disk /
    CacheDirLevels 5
    CacheDirLength 3
</IfModule>
```

Posteriorment, hem d'afegir els mòduls `a2enmod cache disk_cache`, (o `cache_disk` en Apache 2.4) i habilitar el lloc, si no ho està, i reiniciar Apache.

Com es pot observar a la figura 10, que mostra la càrrega de l'URL de la UOC, en els dos cercles grocs podem veure l'efecte de la memòria cau, on la primera vegada triga 15,36 segons i la segona, només 2,24 segons, és a dir, el 14% del temps. S'ha d'anar amb compte amb els paràmetres de memòria cau per a Apache 2.2 i 2.4, ja que difereixen en la seva configuració [AModCach].

Figura 10



Existeixen altres aspectes de memòria cau sobre Apache que no han estat tractats, com ara `File Caching` per accelerar l'accés als arxius que serveix Apache i `Key-Value Caching` utilitzada per SSL i `authentication caching`.

1.6.3. Servei de `proxy`: Squid

`Squid` és un `proxy caching server` per a web i dóna suport als protocols HTTP, HTTPS, FTP, entre d'altres. Aquest redueix l'amplada de banda, millora el temps de resposta emmagatzemat en cau i reutilitza les pàgines més freqüents.

Lectura recomanada

Podeu consultar [DigOCCach] per a configuracions i comentaris sobre `caching` i Apache.

Squid té un extens conjunt de regles de control que permeten optimitzar el flux de dades entre el client i el servidor aportant seguretat i control i encaminant les peticions correctament, la qual cosa permet el seu control i millora la utilització de l'amplada de banda de la xarxa. Squid té diferents modes de funcionament, però com a més importants podem esmentar *Forward Proxy* (és la manera bàsica sobre la qual es configura tota la resta), *Transparent* o *Interception Proxy* (permet incloure un *proxy* en una xarxa sense que els clients hagin de configurar res) i *Reverse Proxy* o *Accelerator-mode* (permet executar Squid per a millorar la resposta d'una granja de servidors web).

Per a instal·lar Squid com a *proxy-cache* (<http://www.squid-cache.org/>), en Debian fem `apt-get install squid3` i editarem l'arxiu de configuració `/etc/squid3/squid.conf` per a portar a terme una configuració bàsica. S'ha de tenir en compte que Squid és molt potent, però això es tradueix en una configuració que pot ser complexa; com a idea, simplement hem de considerar que l'arxiu de configuració, que està molt ben explicat, té aproximadament 7.660 línies (no obstant això, si executem

```
grep -v "^#" /etc/squid3/squid.conf | awk '$1 != "" {print $0}'
```

podrem veure que la configuració bàsica són unes 40 línies mentre que la resta són comentaris i opcions comentades) [squid, squide]

Definir l'ACL (*access control list*) per a habilitar la xarxa/IP que desitgem fer de *proxy*, en la línia 1.056 agreguem: `acl lan src 192.168.1.0/24`

Permetre l'accés, en la línia 1.220 (aprox.) agreguem:

```
http_access allow lan
```

Canviar el port d'accés (línia 1.622), per defecte és `http_port 3128` i es pot deixar que sigui l'estàndard per a Squid o posar el que es prefereixi (per exemple, 8080). Amb això tindriem una configuració mínima però també es pot afegir la següent configuració:

Definim el nom visible, línia 5.273 (aprox.):

```
visible_hostname remix.world
```

Modifiquem la visibilitat de l'IP, línia 7.359 (aprox.): `forwarded_for off`

Finalment, reiniciem el servidor: `service squid3 restart`

Ens donarà un missatge similar a

```
[ ok ] Restarting Squid HTTP Proxy 3.x: squid3[....] Waiting.....done.
```

(trigarà uns segons).

Amb el servidor en marxa, podem configurar els navegadors perquè utilitzin el *proxy*. Per exemple, en IceWeasel/Firefox en l'apartat de *Preferences/Options->Advanced->Network->Proxy* podem introduir les dades del nostre servidor (ip o

nom.domini i port). Sobre Chrome, per exemple en Windows, s'ha d'anar a *Settings->Advanced->Change proxy setting->Connections->Lan Settings* i introduir les dades del nostre servidor (ip o nom.domini i port).

Per a bloquejar dominis hem d'afegir el següent:

1) A la línia 1058 (aprox.) però sempre abans de l'*http_access allow lan*:

```
acl block_tld dstdomain .tv .xxx
http_access deny block_tld
deny_info TCP_RESET block_tld
```

Amb això bloquegem els dominis *.tv* i *.xxx* i amb *TCP_RESET* es resetejarà la connexió i el client no sabrà què ha passat.

2) Per controlar paraules podem fer (sempre abans d'*http_access allow lan*):

```
acl bad_keywords url_regex "/etc/squid3/denegado.txt"
http_access deny bad_keywords
```

L'arxiu */etc/squid3/denegado.txt* tindrà per exemple:

```
as
marca
lavanguardia
```

Això indicarà que la URL que tingui aquestes paraules/dominis no podrà ser accedida.

Una altra de les opcions que permet Squid és actuar com a *reverse proxy*, que és un tipus de *proxy* on les dades es recuperen d'un servidor i retornen als clients com si s'originessin en el *proxy*, de manera que els servidors queden ocults als clients com es mostra en la figura*. Això permet fer polítiques de balanç de càrrega i que les peticions estiguin en un únic domini, malgrat que internament poden estar distribuïdes en diversos servidors. Per a la seva configuració, hem de fer:

*https://upload.wikimedia.org/wikipedia/commons/6/67/Reverse_proxy_h2g2bob.svg

Especificar l'adreça del servidor intern, en la línia 1.626 modificar:

```
http_port 80 defaultsite=192.168.1.33
```

Agregar *cache_peer*, en la línia 2.470 (aprox.) agregar:

```
cache_peer 192.168.1.33 parent 80 0 no-query originserver
```

Canviar l'*acl* per a permetre qualsevol connexió, en la línia 1.174 (aprox.) modificar *http_access allow all*

Finalment, reiniciem el servidor: `service squid3 restart`

Quan ens connectem a l'IP/domini del servidor *proxy*, en realitat veurem les pàgines enviades pel servidor 192.168.1.33. Com a prova de concepte (si volem fer la prova amb una única màquina), podem posar, en lloc del servidor 192.168.1.33, un servidor extern (p. ex., debian.org o la seva IP), i quan posem com a URL el del nostre domini, visualitzarem la pàgina de debian.org.

Altres de les configuracions àmpliament utilitzades és *interception proxy* (o *transparent proxy*), que intercepta la comunicació normal a la capa de xarxa sense necessitat de configuracions específiques en el client, i per la qual cosa no sabran que estan darrere d'un *proxy*. Generalment, un *transparent proxy* està normalment localitzat entre el client i Internet amb el *proxy* fent les funcions de *router* o *gateway*. És habitual en les institucions que desitgen filtrar algun trànsit dels seus usuaris, ISP per a fer cau i estalviar amplada de banda o països que controlen l'accés a determinats llocs per part dels seus ciutadans. Per a implementar-lo sobre una institució, l'habitual és disposar d'una màquina amb dues interfícies de xarxa amb una de connectada a la xarxa interna i una altra cap a la xarxa externa. Els passos per a la seva configuració estan descrits en <http://wiki.squid-cache.org/configexamples/intercept/linuxdnat>:

Sobre *squid.conf*:

Modificar la `acl/http_access` sobre les IP, IP/Mask permeses com en el primer cas.

Configurar el port/mode com `http_port 3129 transparent` o en Squid 3.1+ s'haurà d'utilitzar `http_port 3129 intercept` per a interceptar paquets DNAT.

Sobre */etc/sysctl.conf* modificar:

Permetre el *packet forwarding*: `net.ipv4.ip_forward = 1`

Controlar el *source route verification*: `net.ipv4.conf.default.rp_filter = 0`

No acceptar *source routing*: `net.ipv4.conf.default.accept_source_route = 0`

Considerant que l'IP de *proxy* està en la variable SQUIDIP i el port està en SQUIDPORT, incloure les següents regles de DNAT:

```
iptables -t nat -A PREROUTING -s $SQUIDIP -p tcp -dport 80 -j ACCEPT
iptables -t nat -A PREROUTING -p tcp -dport 80 -j DNAT -to-destination
    $SQUIDIP:$SQUIDPORT
iptables -t nat -A POSTROUTING -j MASQUERADE
iptables -t mangle -A PREROUTING -p tcp -dport $SQUIDPORT -j DROP
```

1.6.4. Proxy SOCKS

SOCKS (abreujament de SOCKeT S) és un protocol d'Internet (en el model OSI estaria en una capa intermèdia entre la d'aplicació i la de transport) que per-

met a les aplicacions en mode client-servidor travessar de manera transparent un *firewall* de xarxa. Els clients que hi ha darrere d'un *firewall*, els quals necessiten accedir als servidors de l'exterior, poden connectar-se en el seu lloc a un servidor *proxy* SOCKS. Aquest servidor *proxy* controla quin client pot accedir al servidor extern i passa la petició al servidor. SOCKS pot ser utilitzat també de la forma contrària, la qual cosa permet que els clients de fora del *firewall* (clients externs) es connectin als servidors de dins del *firewall* (servidors interns) [socks, psocks].

S'ha de tenir en compte que SOCKS només serveix en mode client/servidor, per la qual cosa un usuari ha de tenir instal·lat un client SOCKS, ja sigui en l'aplicació (com Firefox, Chrome) o dins de la pila TCP/IP des d'on el programari del client redirigeix els paquets en un túnel SOCKS. El procediment habitual comença quan el client SOCKS (p. ex., intern en una xarxa privada) inicia una connexió a un servidor SOCKS (el protocol SOCKS permet l'autenticació i el registre de les sol·licituds de connexió) i aquest (servidor SOCKS) actuarà com a client IP per a la sol·licitud de connexió (del client intern en nom seu), la qual cosa significa que el servidor extern només serà conscient de les peticions del servidor SOCKS (així que actuarà com a *proxy forwarding*).

La pregunta habitual és si SOCKS resol de manera diferent el problema d'accés extern mitjançant NAT. La resposta és sí, ho fa de manera diferent, ja que els paquets en NAT només modifiquen les adreces (p. ex., canvien les IP privades per les públiques del *router* com succeeix en un *router* ADSL) i el servidor rep/contesta les peticions. La sessió IP s'estableix directament des del client al servidor i el *router*/FW només modifica/filtra el paquet però no hi ha autenticació ni inspecció del paquet/aplicació/dades (es podria fer, però és complex). Els avantatges de SOCKS rau en el fet que proveeix autenticació per a protocols que no ho permeten, pot traspasar el *routing* per defecte d'una xarxa interna. Si bé HTTP i Telnet suporten autenticació per *firewall* (p. ex., utilitzant *Authenticated Proxy* on a Cisco *firewall*), els protocols encriptats mai poden ser autenticats per un FW, i en canvi SOCKS sí pot fer-ho. No obstant això, hi ha desavantatges, ja que el client ha de tenir interfície a SOCKS, el SO client ha de tenir interfície a SOCKS (per a interceptar el trànsit i reexpedir-lo al SOCKS *proxy*) i necessitem un servidor SOCKS específic per a aquesta fi.

Un cas d'ús habitual és si considerem que estem en un punt de connexió sense fil oberta (p. ex., Wi-Fi) i no es desitja enviar dades de navegació sobre text net o es vol accedir a una pàgina web filtrada per *router*/FW perimetral. Una solució molt simple és utilitzar SSH (que inclou un servidor SOCKS) que pot xifrar tot el trànsit de navegació pel web i reencaminar-lo a través d'un equip de confiança quan s'està en algun altre punt de la xarxa. Per a això, haurem de disposar d'un servidor SSH perquè actuï com a representant en un ordinador remot (que li permeti connectar-se al mateix a través de SSH) i un client d'SSH en l'equip que està utilitzant. El que es farà amb un *proxy* és la creació d'un *middle-person* entre l'usuari i Internet. El navegador farà les sol·licituds de pàgines web al servidor *proxy*, que controla la sol·licitud i obté

la pàgina des d'Internet i les retorna al client. El lloc web en realitat pensa que la sol·licitud prové del servidor *proxy*, no de l'equip que l'ha originat ocultant l'adreça IP d'origen. A més, la connexió entre l'ordinador i el *proxy* que passa a través d'SSH és xifrada i això evita que algú pugui obtenir els paquets des de la Wi-Fi (*sniffers* de Wi-Fi) en el lloc de la connexió.

Per a la seva configuració des d'on ens connectem, hem de tenir accés a un servidor SSH, sobre el qual crearem un túnel que passarà el trànsit web entre la nostra màquina local i el *proxy* SOCKS sobre SSH. Per a això, executem sobre la nostra màquina `ssh -ND 9999 login@remote-server.org` on haurem de reemplaçar `login@remote-server.org` amb l'usuari i nom o IP del servidor remot. El que està fent aquesta ordre és un *port forwarding* a través del port 9999 (pot ser qualsevol altre, però convé que sigui superior a 1024 per a evitar que només ho pugui fer *root*) i la connexió es reenvia a través d'un canal segur on el protocol d'aplicació s'utilitza per a determinar on connectar des de la màquina remota. Actualment, OpenSSH suporta els protocols SOCKS4-5 i per això actuarà com un servidor SOCKS. A continuació, se sol·licitarà el *passwd* i una vegada autenticat no passarà res (el `-N` indica que no obri un *prompt* interactiu, però continuarà funcionant). Si pel *firewall* només podem sortir pel port 443, per exemple, hauríem de configurar el *ssh server* per a escoltar pel port 443 i en el seu lloc executar `ssh -ND 9999 login@remote-server.org -p 443`. Ara és necessari configurar el client per a connectar-se al *proxy*, per exemple Firefox: *Options* -> *Advanced* -> *Network* -> *Connection* i seleccionar SOCKS, com a nom del servidor *localhost* (o el seu nom real si en té) i el port (9999) i guardar els ajustos i verificar que podem navegar sense problemes. Es pot utilitzar el *plugin* Foxy-Proxy* per a Firefox, que permet canviar entre el *proxy* i la connexió directa en funció del lloc o d'un control. Com a mesura addicional (d'anonimat), es pot configurar el servidor *proxy* per a resoldre peticions DNS en lloc del mètode habitual en Firefox posant com a URL `about:config` i modificant `network.proxy.socks_remote_dns=true`. També per a connexions lentes es pot utilitzar l'opció `-C` de *ssh* per a fer servir la compressió de SSH per *gzip*. En Thunderbird o altres clients, la configuració és similar.

*<https://addons.mozilla.org/es/firefox/addon/foxyproxy-standard>

Si el túnel deixa de funcionar (acostuma a ocórrer en xarxes molt ocupades), es pot utilitzar el paquet *autossh* en lloc del *ssh* per a establir la connexió que s'encarregarà de mantenir el túnel obert reiniciant automàticament la connexió. Un altre paquet interessant és *tsocks* (<http://tsocks.sourceforge.net/>), que es pot utilitzar quan el client que desitgem utilitzar no suporta el protocol SOCKS. *tsocks* monitora la trucada d'inici de sessió d'una aplicació (*connect*) i redirecciona la comunicació cap al *server* SOCKS sense que l'aplicació tingui cap informació. Per a això, s'ha d'instal·lar *tsocks* i configurar el *proxy* SOCKS que haurà d'utilitzar en el fitxer `/etc/tsocks.conf` indicant-hi els valors (p. ex., `server = 127.0.0.1`, `server_type = 5`, `server_port = 9999`). Després, bastarà a cridar l'aplicació amb `tsocks aplicació` o simplement l'ordre que obrirà una nova *shell* redirigida al *proxy* i per la qual cosa tot el que s'executi allà serà enviat al *proxy* SOCKS.

1.7. Seguretat a Apache

Si bé el tema de la seguretat serà tractat posteriorment, en aquest subapartat, i continuant amb la funció d'Apache com a proveïdor de diferents serveis, farem un avanç sobre la seguretat de les aplicacions web, que és coneguda com a *Web Application Firewalls* (WAF), donada la seva importància i les necessitats cada vegada majors d'evitar intrusions i reduir els riscos. És important indicar que, a més, comencen a sorgir estàndards, com per exemple PCI DSS - Payment Card Industry Data Security Standard v.1.1, en els quals les revisions regulars del codi i l'ús d'un WAF són un dels criteris que ha de complir l'entorn de producció.

Les aplicacions web (botigues en línia, portals corporatius/privats/personals, etc.) són avui dia un dels punts d'atac habituals per la informació que gestionen. És freqüent veure en els mitjans de comunicació incidents d'accés a la informació (o divulgació d'informació privada), que en general comencen amb la intrusió a través d'un portal web, utilitzant l'explotació dels punts febles en l'aplicació web, i que no es detecten (o no es detecten amb suficient precisió) amb els sistemes tradicionals de seguretat (tallafocs o IDS/IPS). És per això que l'administrador ha de disposar d'eines específiques per conèixer les vulnerabilitats dels seus servidors, així com detectar i prevenir aquest tipus d'atacs.

Donat el seu interès, existeix una gran quantitat d'aplicacions (complexes i de cost elevat) que permeten les funcions de WAF, però en l'àmbit d'*open source* es poden trobar dos projectes que, per la seva qualitat i potencialitat, són molt interessants: OWASP (*Open Web Application Security Project*) i ModSecurity.

OWASP [Owasp] és un projecte que té com a objectiu desenvolupar eines i bones pràctiques per recolzar els desenvolupadors, els administradors de projectes i els analistes de seguretat en el desenvolupament i funcionament de les aplicacions web segures. És important destacar que OWASP no és només un conjunt d'eines (WebGoat [WGoat], Zed Attack Proxy [ZAP], JBroFuzz [JBF], o LabRat [LabRat]) sinó tot un conjunt de projectes, guies i distribucions vinculades al tema de WAF [OProj]. Cal destacar també que OWASP és en ella mateixa una fundació sense ànim de lucre que gestiona els projectes i la infraestructura OWASP, i que està formada per empreses, institucions educatives i particulars de manera que constitueix una comunitat de seguretat informàtica que treballa per crear metodologies, documentació, eines i tecnologies sota les premisses de l'*open source*.

ModSecurity [MSec] és un mòdul de codi obert i multiplataforma que actua com a *firewall* d'aplicacions Web (WAF). És coneguda per molts administradors com la "navalla suïssa" dels WAF, ja que permet augmentar la visibilitat del trànsit HTTP(S), proporciona un potent llenguatge de regles i, a més, una API per implementar proteccions avançades. ModSecurity inclou regles des d'OWASP (ModSecurity Core Rule Set) que li permeten configurar regles de

detecció per a atacs en general (com *HTTP Protocol Protection*, *Real-estafi Blacklist Lookups*, *HTTP Denial of Service Protections*, *Generic Web Attack Protection*, *Error Detection and Hiding*). També permet incloure regles d'altres proveïdors comercials (amb cost econòmic), com Trustwave SpiderLabs, que estan basades sobre la intel·ligència recollida d'atacs, proves de penetració i de casos reals.

ModSecurity permet el monitoratge, el registre i el control d'accés en temps real d'aplicacions web. Segons el seu desenvolupador, pot ser considerat com un "facilitador", ja que no indica quines regles ha de posar, sinó quines possibilitats té l'administrador entre les funcions disponibles d'acord amb el camí que es desitgi seguir. Una llista dels escenaris d'ús més importants seria:

- **Monitoratge i control d'accés.** Monitoratge de seguretat de les aplicacions en temps real i control d'accés, que permet analitzar el flux de trànsit HTTP en temps real, juntament amb la capacitat d'inspeccionar-lo i emmagatzemar-lo per a un posterior seguiment, i correlació d'esdeveniments a través del temps.
- **Virtual Patching.** És un concepte de mitigació de les vulnerabilitats en una capa separada, on s'arriben a resoldre problemes en aplicacions sense haver de tocar les aplicacions pròpies. ModSecurity permet realitzar pegats virtuals a causa de les seves capacitats fiables de bloqueig i del llenguatge flexible de regles adaptables a qualsevol necessitat.
- **Registre complet de trànsit HTTP.** Per motius de seguretat, tradicionalment, els servidors web no tenen registres dels inicis de sessió. ModSecurity permet registrar qualsevol cosa que necessiti, incloent-hi dades de la transacció en brut, cosa que és essencial per a l'anàlisi forense, ja que permet escollir quines transaccions es registren, quines parts d'una transacció es registren o quines parts no són necessàries.
- **Avaluació de la seguretat passiva contínua.** Funciona com a sistema d'alerta primerenca que permet detectar rastres d'anomalies i fallades de seguretat abans que siguin explotats.
- **Enduriment d'aplicacions Web.** Redueix la "superfície d'atac" ajudant a l'execució de nombroses restriccions similars, directament o a través de la col·laboració amb altres mòduls d'Apache (per exemple, resoldre els problemes de gestió de sessions o vulnerabilitats de *cross-site request forgery*). Entre els principis de disseny, es poden enumerar la flexibilitat, la passivitat (no interacció si no està indicat), la previsibilitat i la qualitat. Admet dues opcions de funcionament: incrustat (com a mòdul) o com a *reverse proxy*. Com a incrustat, s'adapta a la infraestructura ja desplegada i escala amb ella, no introdueix nous punts de fallades, però té com a repte que els recursos són compartits amb el servidor web. Com a *reverse proxy*, juga el paper de *router HTTP* col·locat entre els servidors web i els seus clients.

En aquest cas, ModSecurity protegirà de manera global tot el trànsit, independentment d'on vagi, permetent, així, separar la capa seguretat i aïllant totalment els sistemes que està protegint. A més, donat que compta amb recursos dedicats específicament a això, permetrà tenir regles més complexes i adequades al global de la informació que rep. El principal desavantatge d'aquest enfocament és que afegeix un nou punt de fallada i s'haurà de tenir en compte amb una configuració d'alta disponibilitat de dos o més servidors *proxy* inversos.

La instal·lació i configuració (bàsica) de ModSecurity ha de seguir els passos següents:

1) Instal·lem Modsecurity: `apt-get install libapache2-modsecurity` (cal verificar amb `apachectl -M` que apareix el mòdul `security2_moduli (shared)`).

2) Instal·larem també PHP i MySQL per fer les proves (cal recordar el `passwd` introduït en l'usuari `root` de MySQL):

```
apt-get install php5 php5-cgi libapache2-mod-php5 php5-common
php-pear mysql-server php5-mysql
```

3) Activem la configuració bàsica:

```
cd /etc/modsecurity
mv modsecurity.conf-recommened modsecurity.conf
```

4) Editem aquest arxiu `vi /etc/modsecurity/modsecurity.conf` i canviem les següents línies:

```
SecRuleEngine On                #activació de ModSecurity
SecRequestBodyAccess On #permet llegir la resposta -deshabilitar-la si no és
                             #necessari, ja que redueix prestacions i pren espai de log
SecRequestBodyLimit 13107200    #Ajustar d'acord amb les necessitats
SecRequestBodyNoFilesLimit 131072 #ja que impliquen recursos
SecRequestBodyInMemoryLimit 131072
```

5) Reiniciem Apache (`apachectl restart` o també es pot verificar primer la configuració amb `apachectl configtest` i, després, reiniciar).

Com a prova de concepte, detectarem una injecció i una introducció de Spam. Per al primer cas, crearem una pàgina amb una consulta simple a Mysql (vegeu [DigOcMS]):

```
vi /var/www/html/login.php
```

```
<html><body>
<?php
    if (isset($_POST['login']))
    {
        $username = $_POST['username'];
        $password = $_POST['password'];
        $con = mysqli_connect('localhost', 'root', 'psswd_de_la_BD', 'test');
        $result = mysqli_query($con, "SELECT * FROM `users` WHERE username='$username'
        AND password='$password'");
```

```

        if (mysqli_num_rows($result) == 0)
            echo 'Usuari o Passwd INCORRECTES!';
        else
            echo '<h1>Logged in</h1><p> For Your Eyes Only!! </p>';
    }
    else
    {
?>
        <form action="" method="post">
            Usuari: <input type="text" name="username"/><br/>
            Contrasenya: <input type="password" name="password"/><br/>
            <input type="submit" name="login" value="Login"/>
        </form>
<?php
    }
?>
</body></html>

```

A continuació, creem la base de dades amb `mysql -o root -p i`, després d'introduir la contrasenya, executem:

```

create database test;
connect test;
create table users(username VARCHAR(100),password VARCHAR(100));
insert into users values('pirulo','123456');
quit;

```

Si ens connectem ara al servidor `http://srv.nteum.org/login.php`, ens apareixerà la pàgina i podrem fer el procés de *Login* (usuari *pirulo* i *passwd* 123456) de manera que ens mostrarà el text de ***Logged in For Your Eyes Only!!***

Si ara fem una injecció introduint com a usuari `' or true --` (cal tenir en compte que hi ha d'haver un espai després de `--` si no, no funcionarà), podrem veure que la injecció funciona i que ens hem saltat el procediment de *login*.

Per detectar aquest tipus d'injecció, farem l'activació de les regles en el directori `/usr/share/modsecurity-crs/activated_rules/` fent un enllaç des de les múltiples regles que té (vegeu la documentació a [ModSRef]) fent:

```

cd /usr/share/modsecurity-crs/activated_rules/
ln -s ../base_rules/modsecurity_crs_41_sql_injection_attacks.conf .

```

Afegim, a més, a `/etc/apache2/mods-enabled/security2.conf` dins del bloc `<IfModule security2_module>`

```

Include "/usr/share/modsecurity-crs/*.conf"
Include "/usr/share/modsecurity-crs/activated_rules/*.conf"

```

Reiniciem Apache (`apachectl restart`), tornem a carregar la pàgina i realitzem la injecció, però ara veurem *Forbidden. You don't have permission to access /login.php on this server* y en `/var/log/apache2/modsec_audit.log` una cosa com ara:

```
Message: Access denied with code 403 (phase 2). Pattern match "([\~\!|\@|\#|\$|\%|\^|\&|\*|\(|\)|\|
-\|+|\|=\\{\|\}\|\[\|\]\|\|\:\|\;\|'|\'|\\xc2\\xb4\\|\\xe2\\x80\\x99\\|\\xe2\\x80\\x98\\|'\\|<|\|>|.*){8,}" at REQUES
T_COOKIES: __ar_v4. [file "/usr/share/modsecurity-crs/activated_rules/
modsecurity_crs_41_sql_injection_attacks.conf"]
[line "157"] [id "981172"] [rev "2"] [msg "Restricted SQL Character Anomaly Detection Alert - Total #
of special characters exceeded"] [data "Matched Data: : found within REQUEST_COOK
IES: __ar_v4: 5SKFSKF2FJD2TP2KUCUHQ4:20160626:19|GTWLEK3HFHKPMLGC4VUAQ:20160626:19|5Q2SJSWEVZDJTDUCO
MQNW:20160626:19"] [ver "OWASP_CRS/2.2.9"] [maturity "9"] [accuracy "8"] [tag "OWASP_CRS/WEB_ATTACK
/SQL_INJECTION"]
```

On detecta clarament la injecció i proveeix tota la informació necessària. Per evitar que introdueixin *Spam* (vegeu més informació a [DigOcModS]), creem una pàgina web i una caixa de text en php: `vi /var/www/html/test.php`.

```
<html>
  <body>
    <?php
      if (isset($_POST['data']))
        echo $_POST['data'];
      else
      {
        ?>
          <form method="post" action="">
            Introduir algun text:<textarea name="data"></textarea>
            <input type="submit"/>
          </form>
        <?php
          }
        ?>
      </body>
</html>
```

Això permetrà introduir qualsevol text sense control (en aquest cas, només es visualitza el text introduït, però podria emmagatzemar-se amb el consegüent espai perdut si és *spam*). Per fer això, crearem un arxiu

```
vi /etc/modsecurity/custom.conf
```

```
SecRule REQUEST_FILENAME "form.php" "id:'400001',chain,deny,log,msg:'Spam detected'"
SecRule REQUEST_METHOD "POST" chain
SecRule REQUEST_BODY "@rx (?i:(casino|lottery|rolex))"
```

És important verificar que tenim a `/etc/modsecurity/modsecurity.conf` el paràmetre `SecRequestBodyAccess On`.

Amb això ja podem fer una crida a `http://srv.nteum.org/test.php` i introduir qualsevol paraula, que les mostrarà a continuació. Però, si en el text introduïm una de les paraules filtrades, (*casino|lottery|rolex*), per exemple *alskjhfalks-drolexhggakjsfagsdav*, veurem que obtenim un *Forbidden* i, en el *log*, el missatge que hem indicat (msg "Spam detected"):

Lectura recomanada

Donada la potencialitat de ModSecurity, és recomanable ampliar aquest subapartat amb [ModSRef] [ModSBook] [ASec] per controlar altres tipus d'atacs.

```
Message: Access denied with code 403 (phase 2). Pattern match "(?i:(casino|lottery|rolex))" at
REQUEST_BODY. [file "/etc/modsecurity/custom.conf"] [line "1"] [id "400001"] [msg "Spam detected"]
```

1.8. Servidor de wiki

Un (o una) **wiki** (del hawaià *wiki wiki*, “ràpid”) és un lloc web col·laboratiu que pot ser editat per diversos usuaris que poden crear, editar, esborrar o modificar el contingut d’una pàgina web, de manera interactiva, fàcil i ràpida; aquestes facilitats fan d’una wiki una eina eficaç per a l’escriptura col·laborativa. La tecnologia wiki permet que pàgines web allotjades en un servidor públic (les pàgines wiki) siguin escrites de manera col·laborativa mitjançant un navegador, utilitzant una notació senzilla per a donar format, crear enllaços, etc. i conservant un historial de canvis que permet recuperar de manera senzilla qualsevol estat anterior de la pàgina. Quan algú edita una pàgina wiki, els canvis apareixen immediatament a la web, sense passar per cap tipus de revisió prèvia. *Wiki* també es pot referir a una col·lecció de pàgines d’hipertext, que qualsevol persona pot visitar i editar (definició de Wikipedia). Debian té el seu wiki en <http://wiki.debian.org/> o també Apache en <http://wiki.apache.org/general/> i ambdues estan basades en **MoinMoin**. MoinMoin és una *Python WikiClone* que permet inicialitzar ràpidament la seva pròpia wiki i només es necessiten un servidor de web i el llenguatge Python instal·lat. A la web de MoinMoin, es troben les instruccions detallades per a instal·lar MoinMoin, però hi ha dues maneres principals de fer-ho: instal·lació ràpida i instal·lació de servidor.

Enllaç d’interès

Per a saber més sobre MoinMoin, podeu visitar la seva pàgina web en: <http://moinmo.in>. En concret, trobareu les instruccions detallades per a instal·lar MoinMoin en: <http://master19.moinmo.in/InstallDocs>.

1.8.1. Instal·lació ràpida

1) Descarregar el paquet des de <http://moinmo.in/moinmoindownload> que serà, per exemple, per a la versió 1.9 `moin-1.9.8.tar.gz`. Si es vol verificar la integritat del paquet, es pot fer `md5sum moin-x.x.x.tar.gz` i verificar que coincideixin el *hash* generat amb aquell que hi ha a la pàgina de descàrrega.

2) Desempaquetar MoinMoin `tar xvzf moin-x.x.x.tar.gz`. Això crearà un directori `moin-x.x.x` en el directori actual amb els arxius en el seu interior.

3) Com que MoinMoin està escrita en Python, és necessari utilitzar l’interpret de Python:

```
cd moin-x.x.x; python wikiserver.py
```

Aquesta ordre mostrarà per pantalla els missatges d’execució del servidor. Entre aquesta informació es mostrarà l’adreça IP sobre la qual està corrent el servidor, que podrà ser alguna cosa com `http://127.0.0.1:8080`. Aquesta opció fa servir un servidor web intern, serà accessible des de la direcció `http://localhost:8080/` i funcionarà fins que es pressioni `Ctrl-C` en el terminal.

1.8.2. Instal·lació de servidor

MoinMoin és una aplicació WSGI (*Web Server Gateway Interface*) i, per tant, el millor entorn per a executar Moin Moin és un que permeti WSGI com, per exemple, Apache amb `mod_wsgi`. En Debian, podem instal·lar el mòdul instal·lant `apt-get install libapache2-mod-wsgi`.

Instal·lació de MoinMoin

Per a instal·lar MoinMoin, s'ha de descarregar l'última versió i descompactar l'arxiu (per exemple, `tar xvzf moin-1.9.7.tar.gz`) i després fer una `cd moin-1.9.7/` i, a continuació, executar:

```
python setup.py install --force --record=install.log --prefix='/usr/local'
```

Per a fer un test simple:

```
cd /usr/local/share/moin/server
python test.wsgi
```

En el navegador, introduir com a URL `localhost:8000` i veurem la pàgina de test de WSGI.

Copiar la configuració:

```
cd /usr/local/share/moin
cp server/moin.wsgi .
cp config/wikiconfig.py .
```

Agregar un arxiu en `/etc/apache2/conf.d/moin.conf` amb el següent contingut:

```
# MoinMoin WSGI configuration
# you will invoke your moin wiki at the root url, like http://servername/FrontPage:
WSGIScriptAlias /usr/local/share/moin/moin.wsgi
# create some wsgi daemons - use these parameters for a simple setup
WSGIDaemonProcess moin user=www-data group=www-data processes=5 \ threads=10 maximum-requests=1000 umask=0007
# use the daemons we defined above to process requests!
WSGIProcessGroup moin
```

Modificar l'arxiu `/usr/local/share/moin/moin.wsgi` agregant al final del paràgraf `a2: sys.path.insert(0, '/usr/local/share/moin')`

Modificar els permisos dels directoris/pàgines:

```
cd /usr/local/share; chown -R www-data:www-data moin;
chmod -R ug+rx moin; chmod -R o-rwx moin
```

Verificar que tinguem un *site* per defecte i que a dins tinguem un directori que permeti l'execució dels *scripts*. Per exemple, fem

Enllaç d'interès

Les instruccions per a instal·lar WSGI per a Apache i configurar MoinMoin en aquest cas es poden trobar en la següent adreça:
<http://moinmo.in/HowTo/ApacheWithModWSGI>.

```
vi /etc/apache2/sites-available/000-default.conf
```

i dins de `<VirtualHost>`:

```
<Directory /usr/local/share/moin>
Require all granted
</Directory>
```

Després fem `a2ensite 000-default` i reiniciar Apache (amb la instrucció `service apache2 restart`).

Si ens connectem a l'URL `localhost`, tindrem la pàgina inicial de MoinMoin. Per a configurar el nom de la wiki i l'usuari administrador, podem editar l'arxiu `/usr/local/share/moin/wikiconfig.py`, traiem el comentari de `page_front_page = u"FrontPage"` i indiquem el nom de l'administrador, p. ex., `superuser = [u"WikiAdmin",]`, i reiniciem Apache novament. Per a configurar el llenguatge, hem d'entrar com a administrador (WikiAdmin) (si no tenim un usuari, seleccionem *login*, seleccionem 'you can create one now' i i el creem; ha de coincidir amb aquell que introduïm com a `superuser`). Després, podrem configurar l'idioma des de

```
http://localhost/LanguageSetup?action=language_setup
```

i a partir d'aquesta acció, ja podrem començar a crear la nostra primera wiki (informació addicional en <http://moinmo.in/howto> i particularment en l'adreça <https://moinmo.in/HowTo/Debian8> tenen més informació sobre MoinMoin i Debian).

Per a configurar múltiples wikis, primer heu de copiar `config/wikifarm/*` de la distribució en el directori `moin/config/`. Després, s'han de seguir les instruccions anteriors per a cadascuna de les wikis de la col·lecció (*farm*), tenint en compte que:

- 1) és necessari tenir `data_dir` i `data_underlay_dir` separats per a cada wiki,
- 2) si busqueu que comparteixin alguna configuració, llavors aquesta ha d'estar en `farmconfig.py` i les específiques han d'estar en `mywiki.py`.

1.9. Gestió de còpies de seguretat (*backups*)

Les còpies de seguretat (*backup*) es refereixen a una còpia de les dades originals que es fa amb la finalitat de disposar d'un mitjà per a recuperar-les en cas de pèrdua total o parcial a causa de fallades en els dispositius físics, esborrats per accident o atacs informàtics, infectats per virus o altres causes que fan que la informació no existeixi o no sigui la desitjada. El procés de còpia de seguretat

es complementa amb un procés de restauració de les dades (*restore*) que pot ser total o parcial/selectiu, que permet retornar el sistema informàtic al punt en el qual es van emmagatzemar les dades. Això pot significar la pèrdua d'informació entre el moment en què es fa la còpia de seguretat i el moment en què es detecta que les dades no existeixen o estan corrompudes, per la qual cosa la política de planificació de còpies de seguretat ha de ser una de les activitats importants en tot administrador de sistemes.

S'ha de tenir en compte que la pèrdua de dades és habitual (i no per això sense conseqüències i, en alguns casos, fatals), ja que, d'acord amb estadístiques recents, més del 60% dels usuaris d'Internet declaren haver patit una seriosa pèrdua de dades en alguna ocasió, i segons un estudi de la Universitat de Texas, només el 6% d'empreses amb pèrdua catastròfica de dades sobreviurà, enfront d'un 43% que mai reobrirà el negoci i un 51% que haurà de tancar en un termini de 2 anys. Per a reafirmar més encara la necessitat de còpies de seguretat, i pel que fa a aquells sistemes que continguin dades de caràcter personal i que estiguin subjectes a la legislació del país (p. ex., a l'Estat espanyol la LOPD, Llei Orgànica de Protecció de Dades), una de les obligacions que han de complir les empreses/institucions/individus és tenir còpies de seguretat per a preservar les dades que tenen emmagatzemades i que estan subjectes a aquesta normativa.

1.9.1. Programes habituals de còpies de seguretat

Hi ha diverses opcions per a fer còpies de seguretat amb diferents objectius, prestacions i interfícies en totes les distribucions GNU/Linux (p. ex., *amanda*, *bareos*, *backintime*, *bacula*, *backup2l*, *backupper*, *bup*, *chiark*, *dejadup*, *dirvish*, *flexbackup*, *lucky*, *rdiff*, *vbackup*, entre d'altres). Una de les més utilitzades és **Bacula*** (<http://blog.bacula.org/>) o **Bareos** (<https://www.bareos.org/en/>, bifurcació de Bacula amb millores), que és una col·lecció d'eines per a fer còpies de seguretat en una xarxa. Bacula es basa en una arquitectura client/servidor que resulta molt eficaç i fàcil de manejar, ja que presenta un conjunt molt ampli de característiques i és eficient tant per a un conjunt d'ordinadors personals com per a grans instal·lacions. El paquet està format per diferents components, entre els més importants es poden trobar:

- **Bacula-director**, *daemon* que gestiona la lògica dels processos de *backup*.
- **Bacula-storage**, *daemon* encarregat de manejar els dispositius d'emmagatzematge.
- **Bacula-file**, *daemon* per mitjà del qual Bacula obté els fitxers que necessita per a fer la còpia de seguretat i que s'hauran d'instal·lar en les màquines font dels fitxers que cal fer còpia de seguretat, i
- **Bacula-console**, que permet interactuar amb el servei de *backup*.

Bacula suporta discs durs, cintes, DVD, USB i també diferents bases de dades (MySQL, PostgreSQL i SQLite), però com a contrapartida és necessari dispo-

*<http://www.bacula.org>

sar de tots els paquets instal·lats i la seva instal·lació i posada a punt poden resultar complexes.

Un altre paquet interessant és **BackupPC***, que permet fer còpies de seguretat de disc a disc amb una interfície basada en el web. El servidor s'executa en qualsevol sistema Gnu/Linux i admet diferents protocols perquè els clients puguin escollir la forma de connectar-se al servidor. Aquest programa no és adequat com a sistema de còpia de seguretat d'imatges de disc o particions, ja que no suporta còpies de seguretat en un nivell de bloc de disc; no obstant això, és molt simple de configurar i la possible intrusió sobre la xarxa d'ordinadors en la qual es desitja fer el suport és mínima. Aquest servidor incorpora un client *Server Message Block* (SMB) que es pot utilitzar per a fer còpia de seguretat de recursos compartits de xarxa d'equips que executen Windows.

*<http://backuppc.sourceforge.net/info.html>

La seva instal·lació és senzilla fent, en Debian, `apt-get install backuppc`. Ens indicarà que seleccionem el servidor web (Apache2) i ens indicarà l'usuari i *passwd* que ha creat, no obstant això, aquests *passwd*/usuari es poden canviar amb `htpasswd /etc/backuppc/htpasswd backuppc`. Després, en el nostre navegador posem com a URL `http://localhost/backuppc` i amb l'usuari/*passwd* accedirem a la pàgina principal de l'aplicació on ens donarà l'estat del servidor i les opcions.

Hi ha diverses formes de configurar els clients per a fer les còpies de seguretat i dependrà del mètode per a fer-ho i del sistema operatiu. Per a això, cal consultar en l'apartat de documentació com es configurarà cadascuna de les transferències (<http://backuppc.sourceforge.net/faq/BackupPC.html>).

En el cas d'un sistema (remot) Gnu/Linux, des de la interfície d'administració cal editar-ne la configuració (*host* i *xfer*) i confirmar que s'ha definit com a mètode `rsync` i el directori per a fer la còpia. Com que les còpies de seguretat es fan mitjançant `rsync` en combinació amb `ssh`, és necessari que l'usuari *backuppc* del servidor pugui accedir com a *root* sense clau a la màquina remota. Per a això, s'ha d'adoptar l'entitat de l'usuari *backuppc* (`su - backuppc`) i generar les claus (sense *passwd*) `ssh-keygen -t dsa` i després utilitzar l'ordre `ssh-copy-id root@client` per a copiar la clau. S'ha de verificar que es pot accedir a l'usuari *root* del client a través de `ssh` i sense *passwd*. A partir d'aquest punt, n'hi haurà prou de seleccionar l'equip remot en què s'ha de fer el *backup* des de la interfície d'administració i iniciar una primera còpia seleccionant el botó *Començar còpia de seguretat completa*.

En sistemes Windows, la manera més simple de portar a terme *backups* és mitjançant el protocol SMB, per la qual cosa sobre el sistema Windows s'haurà d'ingressar com a administrador i configurar el sistema per a compartir carpetes de les quals es vulgui fer la còpia de seguretat o bé el disc dur complet (per exemple, C:), i definir un usuari/*passwd* per a compartir aquest recurs. Des de la interfície d'administració, s'ha d'editar la configuració del host remot amb Windows del qual s'han de fer les còpies de seguretat (per la seva IP, per exemple), l'usuari i el mètode `smb` (no oblideu fer *Save* després d'haver

modificat aquestes dades). Definiu en la pestanya *Xfer* el nom del recurs compartit que cal fer les còpies de seguretat en l'equip remot i el nom de l'usuari i clau d'accés de recurs compartit de l'equip Windows remot. A partir d'aquest punt, n'hi haurà prou de seleccionar l'equip remot des de la interfície d'administració i iniciar un primer suport seleccionant el botó Començar còpia de seguretat completa.

Amb Backuppc, es pot definir la freqüència dels suports totals i suports incrementals. De manera predeterminada, el valor per als suports totals és cada 7 dies i per als incrementals és cada dia. Es recomana utilitzar un valor lleugerament inferior als dies. Per exemple, 6,97 en lloc de 7 dies i 0,97 en lloc d'1 dia per a millorar la granularitat del suport (recordeu sempre fer *Save* després de modificar cada opció).

Si per les necessitats de l'entorn, l'empresa o institució necessita un sistema de suport amb opcions i funcionalitats equivalents a les de paquets comercials (ARCServeIT, Tivoli Storage Manager o PerfectBackup), Bareos serà l'opció apropiada [Bareos]. Bareos és una bifurcació (*fork*) del projecte Bacula amb notables millores i que està suportat per un grup actiu de desenvolupadors. De la mateixa manera que Bacula, està format per un conjunt de programes que funcionen en una estructura client-servidor i permet la gestió de suport, recuperació i verificació de les dades a través d'una xarxa d'ordinadors de diferents tipus. Bareos també pot funcionar en un sol ordinador i fer còpies de seguretat en cinta i/o discos. És relativament eficient i fàcil d'usar, però, alhora, ofereix característiques avançades de gestió d'emmagatzematge que fan que siguin fàcils de trobar i recuperar arxius perduts o danyats.

L'estructura de serveis és similar a Bacula on es compta amb *Director*, *Consola*, *Files*, *Storage*, *Catalog*, *Monitor*:

- **Director** (*daemon*): és el que supervisa les còpies de seguretat, la restauració, la verificació i l'emmagatzematge, i s'utilitza per programar còpies de seguretat i recuperar arxius.
- **Consola**: serveix d'interfície de comunicació amb Director (pot ser text o gràfic) i per planificar o gestionar les accions que cal realitzar.
- **Files** (client, s'executa com a *daemon*): s'instal·la en la màquina de la qual es desitja fer còpies de seguretat. És específic per al sistema operatiu en què s'executa i és responsable de proporcionar els arxius (dades i atributs) quan sigui sol·licitat per Director.
- **Storage** (*daemon*): és el que realitza l'emmagatzematge/recuperació específic d'arxius a/des dels mitjans d'emmagatzematge físics o volums.
- **Catalog**: manté els índexs dels arxius i bases de dades de volum per a tots els arxius de còpia de seguretat permetent localitzar i restaurar ràpidament qualsevol arxiu desitjat.

- **Monitor:** controla i supervisa l'estat actual de l'administració Bareos, els *daemons* de *File* i *Storage*.

Per realitzar la instal·lació i una primera prova de concepte amb Bareos (en aquest cas ho farem amb PostgreSQL com a base de dades, ja que, segons alguns administradors, té millor comportament sobre Debian que amb altres bases de dades) farem:

1) Instal·lar PostgreSQL: `apt-get install postgresql`

2) Instal·lar Bareos: `apt-get install bareos bareos-bat`

3) Actualitzar les entrades en la base de dades:

```
su postgres -c /usr/lib/bareos/scripts/update_bareos_tables
```

```
su postgres -c /usr/lib/bareos/scripts/grant_bareos_privileges
```

4) Reiniciar els *daemons*:

```
service bareos-dir start service bareos-sd start service bareos-fd start
```

5) Es pot provar el servei mirant des de la interfície text amb `bconsole` o des de la interfície gràfica `bat`. Allí ens indicarà si ha pogut connectar-se amb el servidor i l'estat. A partir d'això, ja tenim el sistema funcionant i podem veure els treballs planificats (definitos en `/etc/bareos`) i, per exemple, forçar en aquest moment un suport. Per a un primer contacte, es recomana seguir la interfície gràfica amb el tutorial [BareosTut] dels propis desenvolupadors, per conèixer els paràmetres i funcionalitats que presta el servei i també com és el procediment per recuperar o afegir un segon client.

Bareos inclou una interfície web (Bareos-webui) que pot ser instal·lada des del lloc de Bareos [Bareos-WebUI] i algunes restriccions, quant a la utilització de les llibreries TLS (GnuTLS), que permeten el xifrat durant la transmissió, però no en l'emmagatzematge (per a un detall més exhaustiu, consulteu el manual).

1.9.2. **rdiff-backup i rdiff-backups-fs**

Per a administradors o usuaris avançats, hi ha l'opció de `rdiff-backup`, que és una ordre per a la còpia de seguretat d'un directori a un altre i que també pot ser a través d'una xarxa. El directori de destinació posseirà una còpia del directori font però també informació addicional (diffs) per a gestionar millor les còpies incrementals (encara d'arxius antics). L'aplicació també preserva sub-directoris, enllaços no simbòlics (*hard links*), arxius dev, permisos, propietat (uid/gid), dates de modificació, atributs estesos i ACL i pot operar de manera eficient a través d'un *pipe* a `rsync` per exemple, o utilitzar `rdiff-backup + ssh` per a fer *backups* incrementals en lloc remot transmetent només les diferències. També és habitual (i molt simple) tenir un proveïdor de serveis (Gdrive, Dropbox, etc.) amb un directori sobre l'ordinador local que serà sincronitzat sobre el *cloud* del proveïdor i fer la còpia `rdiff-backup` sobre aquest

directori perquè després es transmeti al *cloud* del proveïdor. La forma habitual de treball (després d'instal·lar l'ordre) és molt simple: `rdiff-backup font destinació` i en el cas remot `/algun/dir-local` a `/algun/dir-remot` sobre la màquina `hostname.org` serà

```
rdiff-backup /algun/dir-local hostname.org::/algun/dir-remot
```

però pot ser al revés

```
rdiff-backup user@hostname.org::/remote-dir local-dir
```

i també podrien ser sobre dues màquines remotes

```
rdiff-backup -v5 -print-statistics user1@host1::/source-dir user2@host2::/dest-dir.
```

Per a recuperar un directori local, es fa simplement mitjançant la còpia i si és remot,

```
rdiff-backup -r now hostname.org::/remote-dir/file local-dir/file
```

(podeu veure uns quants exemples més en la següent adreça web: <http://www.nongnu.org/rdiff-backup/examples.html>).

Un dels problemes de recuperar la informació prèviament guardada és que accedir als arxius de còpia de seguretat més recent és fàcil (solament s'ha d'introduir el directori de còpia de seguretat), però és complicat si desitgem accedir i navegar a través de les versions anteriors. `rdiff-backup` permet accedir-hi per un conjunt d'instruccions, que requereixen un coneixement precís dels noms d'arxiu i els temps de còpia de seguretat i que pot ser complicat de recordar o seleccionar. `rdiff-backup-fs` (<https://code.google.com/p/rdiff-backup-fs/>) permet crear un sistema d'arxius en espai d'usuari basat en la informació de `rdiff`. Per a això s'utilitza FUSE (*Filesystem in userspace**), que permet que qualsevol usuari pugui muntar el sistema d'arxius guardat per `rdiff` i navegar per cada increment i còpia efectuada.

*<http://fuse.sourceforge.net>

Una alternativa per a les còpies de seguretat d'un sistema remot per a `ssh` és utilitzar `sshfs` (<http://fuse.sourceforge.net/sshfs.html>), que permet l'accés a l'espai d'usuari a un directori remot mostrant-lo localment, per la qual cosa després aplicant `rdiff` és possible fer còpies incrementals d'aquest recurs remot.

1.10. **Public Key Infrastructure (PKI)**

Per PKI (*Public Key Infrastructure*) s'entén un conjunt de maquinari i programari, polítiques i procediments de seguretat que permeten l'execució amb garanties d'operacions com el xifratge, la signatura digital o el no-repudi de transaccions electròniques. El terme PKI s'utilitza per a referir-se tant a l'autoritat de certificació com a la resta de components, si bé de vegades, de manera errònia,

s'utilitza per a referir-se a l'ús d'algorismes de clau pública. En una operació en què s'utilitzi PKI intervenen, a més de qui inicia l'acció i el seu destinatari, un conjunt de serveis que donen validesa a l'operació i garanteixen que els certificats implicats són vàlids. Entre aquests podem comptar amb l'autoritat de certificació, l'autoritat de registre i el sistema de segellament de temps. La infraestructura PKI utilitza procediments basats en operacions criptogràfiques de clau pública, amb algorismes de xifratge ben coneguts, accessibles i segurs i és per això que la seguretat està fortament lligada a la privadesa de la clau privada i les polítiques de seguretat aplicades per a protegir-la. Els principals usos de la PKI són, entre d'altres els següents: autenticació d'usuaris i sistemes (*login*), identificació, xifratge, signatura digital, comunicacions segures i garantia de no-repudi [pki], [apachessl], [pkicry].

Com s'ha vist anteriorment (per a Apache+SSL), la generació de certificats digitals és extremadament necessària dins de les necessitats habituals dels administradors i usuaris dels sistemes d'informació, per exemple, per a accedir a una pàgina SSL o per a signar un correu electrònic. Aquests certificats es poden obtenir de les entitats certificadores privades com per exemple StartComm (<https://www.startssl.com/>) i Verisign (<http://www.verisign.es/>), que tenen alguns productes gratuïts (per exemple per a signar correus), però, en la majoria, els productes tenen un cost elevat. També podem recórrer a utilitzar GNUPG (<https://www.gnupg.org/>), que és *Open-Source* i per a determinats finalitats és correcte però no per a altres, o bé considerar entitats de certificació institucionals, per exemple a Catalunya IdCat (<http://idcat.cat/>), que ens permetran obtenir certificats de ciutadà (gratuïts) per a signatura, encriptació, autenticació, no repudi però no per a servidors (Idcat sí que pot expedir un altre tipus de certificats però només és per a l'Administració pública i universitats del país). En el present apartat, instal·larem i configurarem una entitat certificadora arrel (i subentitats CA per als diferents dominis de control d'aquesta CA) basada en l'aplicació TinyCA (si bé existeixen altres paquets com OpenCA –<https://pki.openca.org/>– que són més potents i escalables però són més complexos en la seva configuració) que s'adapta molt bé per a petites i mitjanes institucions/empreses. Una entitat certificadora és la base de la infraestructura PKI i emet certificats per a donar garanties d'autenticitat d'una comunicació, un lloc o una informació. Quan instal·lem Apache, hem autosignat el certificat digital (és a dir, nosaltres hem fet tots els passos: petició, generació i signatura) que codifica la comunicació però això no dóna garantia si no es verifica la signatura del certificat autosignat. Per a solucionar aquest problema, i sense recórrer a un proveïdor públic/privat, crearem la nostra pròpia estructura de CA i distribuïrem per canals segurs als nostres usuaris/clients el certificat personal / de servidors i el certificat arrel de la CA perquè els instal·lin en els seus navegadors (com ja ho estan els de StartComm, Verisign, Catcert/Idcat o altres entitats de certificació). Això es fa de manera que quan un lloc web, per exemple, presenti al navegador un certificat digital per a codificar la comunicació SSL, el navegador reconegui el lloc pel certificat arrel que té instal·lat i hi confii (i no surti la típica finestra d'avertiment de lloc insegur) mantenint la privadesa de les comunicacions. Els usos d'aquests certificats creats per aques-

ta CA poden ser diversos: per a signar/criptar un correu, per a validar els nostres servidors SSL o per a configurar la VPN, entre d'altres.

La pràctica habitual és tenir una CA i crear Sub-CA (una per a cada domini) perquè el sistema sigui escalable, per la qual cosa la CA signarà la creació de noves Sub-CA i després d'aquestes (el seu responsable), tindran capacitat per a signar els seus certificats i totes tindran el mateix certificat arrel de la rootCA (guia molt detallada en [tinyca]). Una vegada instal·lada (`apt-get install tinyca`), executem `tinyca2` i ens presentarà la pantalla principal i indicacions per a crear una nova CA que serà la rootCA. Completem la pantalla amb les dades, p. ex., *Name=Rootca-nteum.org, Data for CA=Rootca-nteum.org, SP, passwd, BCN, BCN, NTEUM, NTEUM, adminp@sysdw.nteum.org, 7300, 4096, SHA-1* per a tots els camps. Quan fem OK, apareixerà una nova pantalla per a configurar la rootCA. Cal seleccionar "*Certificate Signing/CRL Signing*", "*critical*" i en Netscape *Certificate Type="SSL CA, S/MIME CA, Object Signing CA.*", i si es desitja posar un URL per a revocar els certificats (la qual cosa pot ser una bona idea) es poden emplenar els camps corresponents, després finalment OK. Amb això es crearan els certificats i apareixerà la pantalla principal de tinyCA amb 4 tabuladors on indica CA (informació sobre la CA activa), *Certificates* (mostra els certificats creats per la CA) *Keys* (mostra les claus dels certificats) i *Requests* (peticions de certificats que esperen ser signats per la CA). Per damunt, apareixen una sèrie d'icones per a accedir a funcions directes però Tinyca2 té un error i no mostra els noms de les icones.

El següent pas és crear una nova sub-CA i, per a fer-ho, verificar que estem en el tab de la CA i fer clic en la tercera icona des de la dreta, que ens mostrarà una finestra que com a subtítol té "*Create a new Sub CA*". Haurem d'introduir el *passwd* que vam posar en la rootCA, donar-hi un nom (subca-sysdw.nteum.org), *Data-for-CA* (sysdw.nteum.org) i la resta de dades com vam fer anteriorment (el *passwd* no ha de ser necessàriament el mateix de la rootCA) i quan fem OK ens sortirà la finestra principal però amb la Sub-CA seleccionada, si volem tornar a la CA haurem d'anar al menú *File->Open* i obrir la rootCA. Recordeu que la Sub-CA serà qui gestionarà les peticions i signarà els certificats per a aquest domini, per la qual cosa hem de seleccionar l'adequada a cada moment. La rootCA només la utilitzarem per a crear/revocar noves sub-CA i per a exportar el certificat arrel de la rootCA (necessari per a enviar-los als nostres clients perquè l'instal·lin en els seus navegadors).

Per a crear un certificat que permeti certificar el nostre *web-server* quan utilitza SSL (<https://sysdw.nteum.org>), amb la nostra Sub-CA seleccionada anem al tab de *Request* i amb el botó dret seleccionem "*New request*" i s'obrirà una finestra on haurem d'introduir l'URL del nostre servidor (sysdw.nteum.org) com a *CommonName* i emplenar la resta de dades. Una vegada creat, el seleccionem i amb el botó dret indiquem "*Sign request*" i seleccionem "*Server request*", que ens mostrarà una finestra amb el *password* de la CA i la validesa. Aquest procediment és el que genera confiança ja que estem validant la informació aportada en la petició i signant el certificat (que ja podrem veure en els tabs corresponents).

Ara haurem d'exportar els certificats (del web i la rootCA), la *key* i configurar Apache perquè els incorpori. En primer lloc, exportarem la rootCA i la introduïrem en Firefox/Iceweasel. Per a això, en la finestra principal *File->OpenCA* seleccionem la nostra rootCA i seleccionant la segona icona de la dreta que correspon a "*Export CA Certificate*" indiquem un nom d'arxiu i el format (PEM és l'adequat) i desem el certificat en el nostre sistema d'arxiu (per exemple, */tmp/Rootca-nteum.org.pem*). S'ha de tenir en compte fer un `chmod 644 /tmp/Rootca-nteum.org.pem`, ja que es desarà com a 600 i altres usuaris no el podran importar. Des de Firefox/Iceweasel seleccionem *Preferences/Setting->Advanced->Certificates->View Certificates->Authorities->Import* i seleccionem l'arxiu abans creat marcant totes les *trust settings* que ens presenta en la finestra següent (3). Després, podrem veure amb el nom que vam donar a *Organization* el certificat corresponent.

A continuació, en TinyCA2 obrim la nostra sub-CA, seleccionem el tab *Certificates* i sobre el certificat seleccionem el botó dret i indiquem *Export certificate* i donem un nom d'arxiu, per exemple *sysdw.nteum.org-cert.pem*, després repetim el procés amb la *key* en el *Key* tab, fem *Export key* i el desem, per exemple com a *sysdw.nteum.org-key.pem*. És important decidir si posem *passwd* o no, ja que si el posem cada vegada que arrenqui el servidor ens sol·licitarà el *passwd*. Sobre el tab *CA* haurem d'exportar ara el *certificate chain*, que és la primera icona des de la dreta i desar-lo com a *sysdw.nteum.org-chain.pem*. Mourem aquests tres arxius a */etc/ssl/private/* i passarem a configurar Apache modificant l'arxiu que configuri el nostre lloc SSL, per exemple nosaltres hem utilitzat */etc/apache2/sites-available/default-ssl*, en el qual hem modificat (només es mostra les línies principals):

```
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
    ServerName sysdw.nteum.org
    ...
    SSLEngine on
    SSLCertificateFile /etc/ssl/private/sysdw.nteum.org-cert.pem
    SSLCertificateKeyFile /etc/ssl/private/sysdw.nteum.org-key.pem
    SSLCertificateChainFile /etc/ssl/private/sysdw.nteum.org-chain.pem
    ...
</VirtualHost>
</IfModule>
```

Només ens queda habilitar el lloc (`a2ensite default-ssl`) i reiniciar el servidor Apache (`service apache2 restart`) (ens demanarà el *passwd* si l'hem posat en la *key*) i provar (en el navegador en què tenim instal·lat el certificat arrel de la rootCA) l'URL `https://sysdw.nteum.org`. Si tot està correcte, carregarà la pàgina sense la típica finestra que informa que el lloc no és segur [tinyca].

Per a crear certificats per a una adreça de correu i distribuir-los als nostres usuaris juntament amb el certificat de la rootCA, podem fer en el tab *Certificates* de la nostra sub-CA, seleccionar *New - Create Key and Certificate (Client)* i entrar el nom i l'adreça de correu per la qual volem validar, així com el *passwd* per a

protegir-lo fins que arribi al seu nou destinatari (i que després el podrà o l'haurà de canviar). A continuació, hem d'exportar-lo tenint en compte utilitzar el format PKCD#12 que inclou el certificat i la clau. Després d'enviar a l'usuari l'arxiu amb el certificat més el de la *root-CA*, podrà agregar-lo al seu gestor de correu de manera similar a *root-CA* però com a *personal certificates*. Després, podrà enviar correus signats digitalment i el destinatari (que haurà de tenir instal·lat el certificat de la *rootCA*) podrà verificar la signatura del correu.

Una altra opció interessant per configurar una CA és el paquet *XCA*, *X Certificate and key management* [*XCA*], que permet crear i gestionar de manera molt simple certificats X.509, peticions de certificats, claus privades RSA, DSA i EC, *Smartcards* i llistes de revocacions (CRLs), és a dir, disposa de tot el necessari per implementar una CA i es poden crear sub-CA de manera recursiva i, a més, inclou plantilles per estendre les necessitats de les peticions que es tinguin.

La seva instal·lació és molt fàcil, `apt-get install xca`. Després s'haurà d'executar com a `xca`. La primera acció que s'ha de fer és crear/seleccionar una base de dades des del menú *File* (o obrir-la si ja s'ha creat en un pas anterior), la qual està protegida per una contrasenya utilitzada per xifrar-la, i mantenir la seguretat de la CA. L'aplicació presenta 5 pestanyes (*Keys*, *Requests*, *Certificates*, *Templates and Revocation lists*), i mitjançant menús, botons a la dreta de l'aplicació o menús contextuais (botó dret del ratolí), es pot accedir a totes les opcions de configuració.

Com a cas d'ús, generarem un certificat per SSL per tal que, després, el puguem carregar a Apache i en el navegador per utilitzar-lo en HTTPS, i que no ens doni l'avís típic de seguretat de "Certificat autofirmado".

Creació del certificat de l'entitat certificadora (RootCA): aquest certificat és el que haurem d'instal·lar en el navegador perquè validi la cadena de confiança quan el servidor web li presenti el certificat SSL del servidor. Per generar-lo, seguirem els següents passos:

- 1) Iniciem *XCA*, i generem o obrim la base de dades introduint la contrasenya.
- 2) Anem a la pestanya *Certificates* -> *New Certificate* i s'obrirà una nova finestra anomenada *Create X509 Certificate*.
- 3) Introduïm la informació d'identificació i anem a la pestanya *Subject*, configurem els valors de la secció *Distinguished name*, després anem a *Generate a new key* i, a la finestra que s'obre, seleccionem el *Name*, *Type (RSA)*, *Size* i fem un *Create*.
- 4) Configurem les extensions X.509: anem a la pestanya *Extensions* i, des de *Type list*, seleccionem *Certification Authority* i modifiquem *Validity dates*. Generalment *RootCA* són vàlids durant 5 anys.
- 5) Configurem l'ús: anem a la pestanya *Key usage* i, des del panell esquerre, seleccionem *Digital Signature*, *Key Agreement*, *Certificate Sign*. Cal anar amb

compte a escollir altres opcions, ja que podria ser que alguns sistemes operatius rebutgessin el certificat.

6) Finalment fem *OK* per crear el certificat que ens apareixerà a la pestanya *Certificates*.

7) Per exportar el certificat, dins de la pestanya *Certificates*, seleccionem el certificat a exportar i fem *Export*. A la nova finestra, seleccionem el nom de l'arxiu i el format PEM des de la llista *Export Format* i, després, fem *OK*.

8) Per instal·lar el certificat en Firefox/Iceweasel, p. ex., anem a *Preferences* ->*Advanced* ->*Certificates*->*View Certificates*->*Authorities*->*Import* i carreguem l'arxiu que hem salvat prèviament. Després de realitzar aquesta operació, el veurem en la llista amb el nom introduït en la secció *Distinguished name* (punt 3), com a *OrganizationalName* i indexat també pel *CommonName* d'aquesta secció.

Creació del certificat SSL i de la clau privada per al servidor: es crearà la clau privada i el certificat que haurem d'instal·lar a Apache:

1) Un cop a XCA, anem a la pestanya *Certificate signing requests*, seleccionem *New Request* i s'obrirà una finestra de *Create Certificate Signing Request*.

2) Anem a la pestanya *Source*, des de *Template* seleccionem *[default] HTTPS_Server* i fem un clic a *Apply extensions*.

3) Anem a la pestanya *Subject*, omplim els camps de la secció *Distinguished name section* (amb cura de posar el FDQN del nostre servidor en *el Common-Name*), generem una nova clau amb *Generate a new key* i, a la nova finestra, seleccionem el *name* per a la clau privada i la grandària (*key size*), després, fem un clic a *Create*.

4) Acabem la petició fent un *OK*.

5) Signem el certificat: anem a la pestanya *Certificate signing requests*, seleccionem el certificat que s'ha de signar i, fent clic al botó dret, seleccionem *Sign*; s'obrirà una finestra *Create x509 Certificate*. En la secció *Source*, a l'apartat *Signing*, seleccionem *Use this Certificate for signing* i, després, el certificat de la *RootCA* des del menú. Finalment, fem *OK*.

6) El certificat ara apareixerà com a signat i a la finestra *Certificate*, sota l'entitat que ha signat el certificat (haurem de fer un clic al + de la *RootCA*).

7) Per exportar-lo, anem a la pestanya *Certificates*, seleccionem el certificat prèviament signat i l'exportem a un arxiu (per exemple, *server.crt*) amb format PEM. Després anem a la pestanya de *Private Keys*, seleccionem la clau privada corresponent i, amb el botó dret, fem un *Exportar*: seleccionem l'arxiu (per exemple, *serverkey.pem*) sense contrasenya (per evitar que cada vegada que reiniciem Apache s'hagi d'introduir la contrasenya) i escollim el format PEM.

Finalment, a Apache, editem

```
vi /etc/apache2/sites-available/default-ssl.conf
```

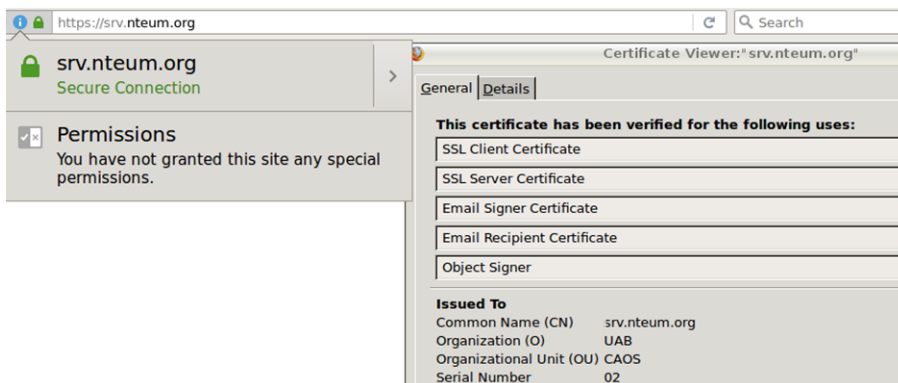
i modifiquem:

```
SSLCertificateFile /etc/ssl/private/server.crt
```

```
SSLCertificateKeyFile /etc/ssl/private/serverkey.pem
```

Copiem el certificat i la clau privada abans generada a `/etc/ssl/private`, habilitem el lloc (`a2ensite default-ssl`), habilitem el mòdul (`a2enmod ssl`) i reiniciem Apache (`apachectl restart`). Finalment, en el navegador verifiquem que amb l'URL del *site* (i amb la RootCA instal·lada) no ens dona errors de SSL (el certificat és validat i surt en verd, com es veu a la figura 11).

Figura 11



Com a anotació final en relació amb la PKI, tots els nostres usuaris/clients/visitants dels nostres serveis hauran de tenir instal·lat el certificat de la *root-CA* en els seus navegadors/clients per a, d'aquesta manera, validar els llocs/serveis que estan sota el nostre domini/subdominis ja que els navegadors/clients de correu no incorporen aquests certificats arrel per defecte. Si el nostre domini/serveis ho requereix, podem gestionar amb Mozilla la inclusió del nostre certificat en <http://www.mozilla.org/en-us/about/governance/policies/security-group/certs/> però no és un tràmit fàcil i és necessari complir amb una sèrie de requisits ja que les garanties del sistema rauen en la inclusió d'aquests certificats.

1.11. Open Computer and Software Inventory Next Generation (OCS)

OCS [OCS] és un servei que permet gestionar els inventaris dels actius d'una infraestructura tecnològica. Funciona mitjançant un servidor i un conjunt d'agents (que funcionen en cadascuna de les màquines que ha de ser inventariada), que recopilen la informació sobre el maquinari i programari d'equips.

Disposa d'una interfície web amb la possibilitat d'afegir diferents *plugins* i diversos criteris de cerques per facilitar la localització d'actius, buscar a la xarxa per mitjà de *IPDiscovery* o instal·lar aplicacions remotament, a través de la creació de *Builds*. La informació intercanviada entre els agents i el servidor està en format XML i el servidor utilitza Apache, MySQL i Perl per gestionar i visualitzar el repositori; necessita molt pocs recursos i és possible la seva instal·lació en diferents plataformes.

Per a la seva instal·lació, hem d'instal·lar alguns paquets previs:

- 1) `apt-get install apache2`
- 2) `apt-get install php5 libapache2-mod-php5 php5-cli php5-common php5-cgi php5-gd`
- 3) `apt-get install mysql-client mysql-server mysql-common php5-mysql`
- 4) `apt-get install libxml-simple-perl libio-compress-perl libdbi-perl libdbd-mysql-perl libapache-dbi-perl libnet-ip-perl libsoap-lite-perl`
- 5) Executem: `cpan -i XML::Entities`
- 6) `apt-get install ocsinventory-server ocsinventory-agent`
- 7) Carreguem en el navegador `http://localhost/ocsreports/install.php`.
- 8) Ens sol·licitarà l'usuari *root* de MySQL, la seva contrasenya i el nom de la base de dades, que serà *ocsweb*, i la ubicació que és *localhost*.
- 9) A continuació, podrem entrar al lloc (`http://localhost/ocsreports/`) amb l'usuari *admin* i la contrasenya *admin*. A la primera pàgina, veurem que surt un "Avis de Seguretat", ja que haurem de fer uns canvis per finalitzar la instal·lació.
- 10) Executem: `cd /usr/share/ocsinventory-reports`
`mv install.php install.php.org`
`mysql -u root -p`

`SET PASSWORD FOR 'ocs'@'localhost' = PASSWORD('psswd_user_ocs');`
`FLUSH PRIVILEGES;`

`vi dbconfig.inc.php` *Canviar el passwd.*

`define("COMPTE_BASE", "ocs");`
`define("PSWD_BASE", "psswd_user_ocs");`

`cd /etc/apache2/conf-enabled/`
`vi ocsinventory-server.conf` *Canviar el passwd.*

`PerlSetVar OCS_DB_PWD psswd_user_ocs`

`service apache2 reload`

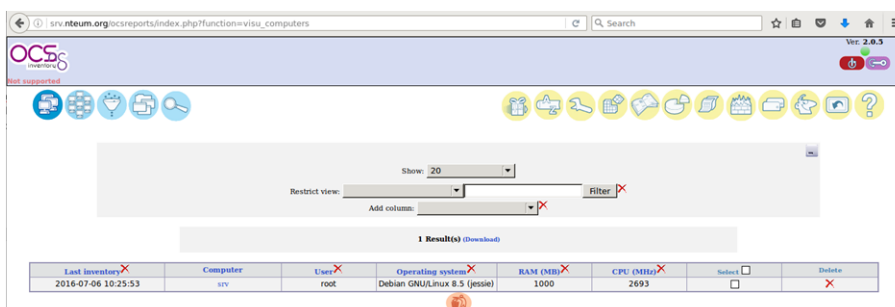
11) Accedim a `http://localhost/ocsreports/` i podrem veure que només hi ha un advertiment de la contrasenya per defecte de l'usuari administrador. A les icones, seleccionem el d'usuaris i canviem o generem un nou usuari amb el rol de *SuperAdministrator*. Cal tenir en compte que, si tenim *mod_security* habilitat a Apache, ens donarà l'error que no pot accedir a `localhost/ocs-reports/index.php`. La causa d'això és que *mod_security* el bloqueja, per la qual cosa s'ha d'afegir la regla adequada (o si només es desitja fer unes proves, es pot deshabilitar temporalment des de `/etc/modsecurity/modsecurity.conf` i reiniciar Apache).

12) Quan hem instal·lat l'agent per a la pròpia màquina (o per a qualsevol altra màquina), hem d'indicar la IP/nom del servidor (o també a l'arxiu `/etc/ocsinventory/ocsinventory-agent.cfg`) i es desarà al `/etc/cron.daily` per executar-se. També podem forçar l'execució amb `/usr/bin/ocsinventory-agent` i ja podrem veure'l en OCS i navegar per tots els paràmetres. Es poden trobar els agents per a Windows, MacOS, Android i altres Linux des del lloc d'OCS-Dev*.

*<https://launchpad.net/ocsinventoryx>

És important tenir en compte que és un paquet molt extens, amb un gran conjunt d'opcions i possibilitats, i l'administrador haurà d'analitzar i avaluar cadascuna d'elles i configurar-les/adaptar-les a l'entorn [OCS].

Figura 12



The screenshot shows the OCS Inventory X web interface. At the top, there's a navigation bar with the OCS logo and version 2.0.3. Below it, there are several icons for navigation and search. The main content area shows a table with one result. The table has columns for 'Last inventory', 'Computer', 'User', 'Operating system', 'RAM (MB)', 'CPU (MHz)', 'Select', and 'Delete'. The data row shows an inventory from 2016-07-06 10:25:53 for computer 'srv', user 'root', operating system 'Debian GNU/Linux 8.5 (jessie)', RAM of 1000 MB, and CPU of 2693 MHz.

Last inventory	Computer	User	Operating system	RAM (MB)	CPU (MHz)	Select	Delete
2016-07-06 10:25:53	srv	root	Debian GNU/Linux 8.5 (jessie)	1000	2693	<input type="checkbox"/>	<input type="checkbox"/>

1.12. GLPi

GLPi (del francès, *Gestionnaire Lliure de Parc Informatique*) és un paquet que permet la gestió d'actius i fallades d'un sistema informàtic (*IT Asset Management and issue tracking system*), o també conegut com a *service desk*. És un paquet que té una alta integració amb d'altres (per exemple, OCS). És una aplicació web i està escrita en PHP; existeix una comunitat molt activa de desenvolupadors i usuaris.

Entre les seves característiques, cal destacar que permet construir un inventari de tots els recursos de l'organització i gestionar les tasques administratives (i fins i tot financeres). A més, ajuda els administradors a tenir i a gestionar una

base de dades d'actius tecnològics, així com a emmagatzemar un historial de les intervencions de manteniment i també assistir els usuaris en la comunicació d'incidències (*help Desk*). [GLPi]

La seva instal·lació (per exemple, després d'haver instal·lat OCS, ja que necessita Apache, PHP i MySQL) és summament fàcil (`apt-get install glpi`) i, per gestionar-la, ens demanarà les contrasenyes del *root* de la base de dades i la de l'usuari *glpi*. A partir d'això, podem connectar-nos a la pàgina web (<http://srv.nteum.org/glpi/>) on sol·licitarà uns usuaris/*passwd*, que, per defecte, són:

- *glpi/glpi* per al compte d'administrador,
- *tech/tech* per al compte de tècnic,
- *normal/normal* per a un compte normal i
- *post-only/postonly* per a un compte només d'enviament.

Amb això, i entrant com a usuari *glpi*, veurem un seguit de tasques que haurem de fer:

1) Canviar els *passwords*. per defecte (pestanya *Administration->Users*)

2) Tornar a anomenar l'arxiu d'instal·lació:

```
mv /usr/share/glpi/install/install.php /usr/share/glpi/install/install.php.org
```

3) Per connectar GLPi i OCS, haurem de buscar el *plugin* apropiat a la nostra versió (per exemple, de Debian 8.5 disposem de GLPi V0.84 –vegeu al peu de pàgina de GLPi–) i, a la pàgina web

<https://forge.glpi-project.org/projects/ocsinventoryng/files>,

l'adequat és el *glpi-ocsinventoryng-1.0.3.tar.gz*, que descarregarem i instal·larem a */usr/share/glpi/plugins* amb

```
tar xzvf /sitio-donde-se-haya-descargado/glpi-ocsinventoryng-1.0.3.tar.gz
```

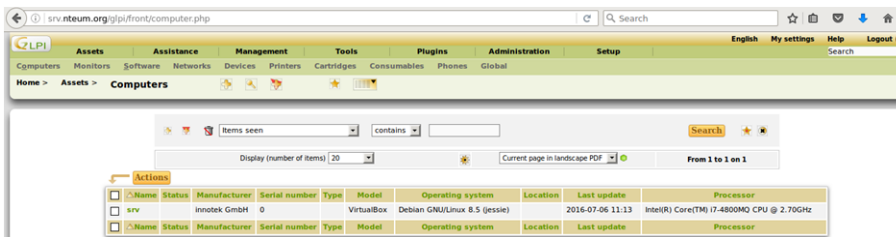
4) Anem a la pestanya de *Setup->Plugins* i veurem el *plugin OCS Inventory NG* que haurem d'actualitzar (*Update*) i activar (*Enable*). Farem un clic aquí (columna *Name*) i accedirem a la seva configuració.

5) Afegim un nou servidor (signe +), indicant el *Name* desitjat, *Host for the database* = *localhost*, *Synchronisation method* = *Standard*, *Database* = *ocsweb*, *User*=*ocs*, *Passwd*=*el-que-tingui-la-base-de-dades-OCS* (si no el recordem, podem consultar l'arxiu */etc/ocsinventory/dbconfig.inc.php*) i fer *Save*. Veurem, en un missatge en la part inferior de la finestra, si ha pogut connectar-se i accedir-hi.

6) A la pestanya *Import* (suggeriment: posar com a *global import*) i, a la pestanya *General*, seleccionem els paràmetres desitjats fent *Save* novament.

7) A continuació, es pot esperar que s'actualitzi automàticament o es pot anar a *Setup->Automatic Actions->ocsng*, canviar l'*Status* i executar-lo en aquest moment (*Execute*). Amb això veurem que l'inventari s'actualitza i la màquina que tenim en OCS (figura 13).

Figura 13



Plugins de GLPI

Són molt interessants les possibilitats d'extensió que té GLPI mitjançant la incorporació de *plugins*. Consulteu la pàgina <http://plugins.glpi-project.org/> on es troben ordenats per categories i tipus.

1.13. Supervisor (Process Control System)

Supervisor és una aplicació (client/servidor) que permet als usuaris/administradors monitoritzar i controlar un conjunt de processos [Sup].

La seva funció és simplificar l'escriptura de *scripts* en *rc.d* (o en *systemd*) per a cada procés d'usuari o administrador; té com a funció el seu inici/reinici/gestió. Aquests *scripts* són complicats d'escriure (més en *systemd*). A més, cas que un procés/servei "caigui", permet definir com posar-lo en marxa automàticament. També permet saber de manera senzilla (mitjançant una pàgina web) l'estat dels processos i veure ràpidament els seus *logs*. A més, permet que un usuari (no administrador) pugui gestionar els processos sense haver de donar-li permisos de `sudo` o accés a la consola, agrupar processos i enviar ordres al grup (inici, aturada, reinici). I tot això mitjançant una interfície senzilla i centralitzada, amb una configuració simplificada, una execució eficient i totalment extensible.

Els components de Supervisor són **supervisord**, que és el servidor responsable d'iniciar els processos (subprocessos) fills registrant la sortida i la sortida d'error estàndard, i la generació i gestió d'esdeveniments per a aquests processos/subprocessos. Aquest servidor es configura amb `/etc/supervisor/supervisord.conf`, que s'haurà de protegir amb els permisos adequats, ja que conté contrasenyes en text sense xifrar. **supervisorctl** és el client (CLI) que proporciona la interacció amb el servidor a través d'un conjunt de subordre (o pot treballar també interactivament). **Servidor web** és una interfície simplificada a la qual s'accedeix a través del port 9001 (p. ex. en `http://localhost:9001/`) i permet veure l'estat del procés de control i executar operacions col·lectives o individuals sobre els processos.

La seva instal·lació és senzilla:

1) `apt-get install supervisor`

2) A continuació, tornem a anomenar la configuració inicial:

```
cd /etc/supervisor
```

```
mv supervisord.conf supervisord.conf.org
```

3) Generem una nova configuració:

```
echo_supervisord_conf > supervisord.conf
```

4) La modifiquem per una de similar a aquesta:

```
[unix_http_server]
file=/var/run/supervisor.sock           ; (the path to the socket file)
chmod=0700                               ; socket file mode (default 0700)

[inet_http_server]
; inet (TCP) server disabled by default
port=127.0.0.1:9001                     ; (ip_address:port specifier, *:port for all iface)
username=admin                           ; (default is no username (open server))
password=nuestro_passwd                  ; (default is no password (open server))

[supervisord]
logfile=/var/log/supervisor/supervisord.log ; (main log file;default $CWD/supervisord.log)
logfile_maxbytes=50MB                    ; (max main logfile bytes b4 rotation;default 50MB)
logfile_backups=10                        ; (num of main logfile rotation backups;default 10)
loglevel=info                             ; (log level;default info; others: debug,warn,trace)
pidfile=/var/run/supervisord.pid          ; (supervisord pidfile;default supervisord.pid)
nodaemon=false                            ; (start in foreground if true;default false)
minfds=1024                               ; (min. avail startup file descriptors;default 1024)
minprocs=200                              ; (min. avail process descriptors;default 200)

[rpcinterface:supervisor] supervisor.rpcinterface_factory=supervisor.rpcinterface:make_main_rpcinterface

[supervisorctl]
serverurl=unix:///var/run/supervisor.sock ; use a unix:// URL  for a unix socket

[include]
files = /etc/supervisor/conf.d/*.conf
```

5) Reiniciarem el servei: `systemctl restart supervisor` (o també amb `service supervisor restart`) i ja podrem accedir via `supervisorctl` o per l'adreça `http://localhost:9001`. Cal indicar que no veurem cap procés i que els haurem de configurar.

6) Per gestionar processos, hem d'escriure un arxiu que contingui les ordres i les indicacions per poder gestionar-los. En el nostre cas, com a prova de concepte, gestionarem els servidors d'Apache i SSH. Per fer això, escriurem a `/etc/supervisor/conf.d` amb arxius, amb el que segueix [PVan]:

```
vi /etc/supervisor/conf.d/http.conf

[program:apache2]
command=apachectl -DFOREGROUND
autostart=true
autorestart=true
startretries=1
startsecs=1
stderr_logfile=/var/log/apache2/supervisor.error.log
stdout_logfile=/var/log/apache2/supervisor.access.log
user=root

vi /etc/supervisor/conf.d/ssh.conf

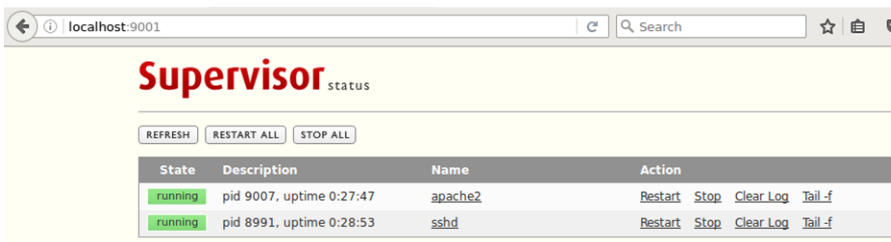
[program:sshd]
command=/usr/sbin/sshd -D -e -f /etc/ssh/sshd_config
autostart=true
autorestart=true
startretries=1
startsecs=1
stderr_logfile=/var/log/supervisor/ssh.error.log
stdout_logfile=/var/log/supervisor/ssh.access.log
user=root
```

7) Donat que aquests processos poden executar-se pels arxius de */etc/init.d*, primer els haurem d'aturar per poder-lo posar sota el control de supervisor. A continuació, fem que es torni a llegir la configuració i que s'actualitzi amb:

```
supervisorctl reread
supervisorctl reload
```

8) Ara podem accedir ja a la pàgina web i veurem alguna cosa semblant al que es mostra a la figura 14, on ja podrem gestionar-los i veure la informació corresponent.

Figura 14



The screenshot shows the Supervisor web interface at localhost:9001. The page title is 'Supervisor status'. There are three buttons: 'REFRESH', 'RESTART ALL', and 'STOP ALL'. Below these is a table with the following data:

State	Description	Name	Action
running	pid 9007, uptime 0:27:47	apache2	Restart Stop Clear Log Tail-f
running	pid 8991, uptime 0:28:53	sshd	Restart Stop Clear Log Tail-f

Lectura recomanada

Consulteu la documentació [Sup] per veure totes les possibilitats de Supervisor en la gestió d'esdeveniments, *logs* i API que permeten obtenir molta més funcionalitat de la mostrada.

És important tenir en compte que, quan posem *autostart=true*, hem d'assegurar-nos que traiem l'*script* d'inici habitual (*/etc/init.d* o *systemd*), ja que està sota el control de *supervisord*. Cas que no ho fem, dependrà de l'ordre d'arrancada, però l'habitual és que *supervisord* doni un error, ja que no podrà posar en funcionament el servei perquè ja existeix o perquè els ports estan ocupats).

1.14. OwnCloud. File Sync & Share Server

OwnCloud [OC] és un projecte que permet accedir, compartir i sincronitzar arxius sobre un servidor (equivalent a opcions comercials com Dropbox, Google Drive o OneDrive). Proveeix l'accés a arxius a través d'una interfície web molt intuïtiva i per WebDav, però també hi ha clients per a Windows, Linux, MacOS i sistemes operatius mòbils que permeten una sincronització fàcil entre el dispositiu i el servidor. També permet una gran quantitat de *plugins*, com ara un visualitzador de PDF, un client de correu, un calendari, un gestor de tasques, etc. La llista completa d'aplicacions es pot obtenir d'<https://apps.owncloud.com/>.

Per a la seva instal·lació seguirem els passos següents:

1) `apt-get install owncloud`

2) `mysql -u root -p` *Introduir el passwd de root per a MySQL i executar:*

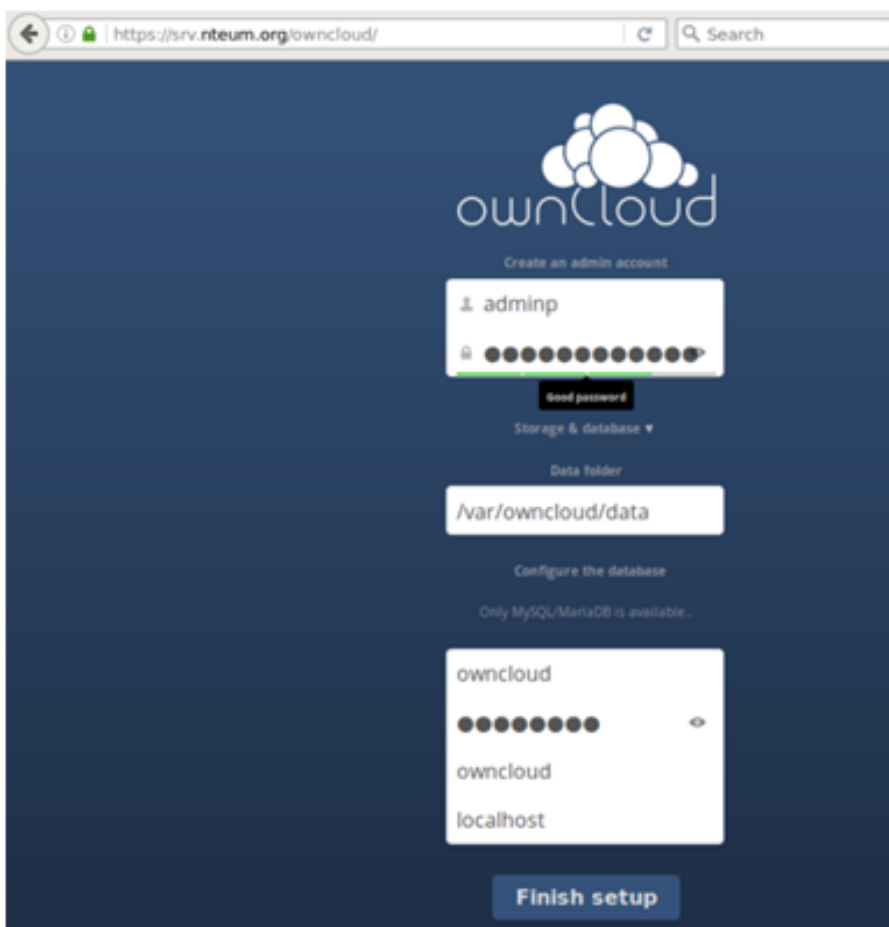
```
CREATE DATABASE owncloud;  
CREATE USER owncloud@localhost IDENTIFIED BY 'passwd-DB-OwnCloud';  
GRANT ALL PRIVILEGES ON owncloud.* TO owncloud@localhost;  
FLUSH PRIVILEGES;  
quit
```

3) Generem l'espai per al repositori (en un lloc on tinguem espai disponible):

```
mkdir /var/owncloud  
chown www-data:www-data /var/owncloud  
chmod 750 /var/owncloud
```

4) Anem a la web per acabar de configurar-la: <http://srv.nteum.org/owncloud/>. Escollim el nom de l'administrador i un *passwd* robust, també el repositori */var/owncloud/*, *Username=owncloud*, *Password=passwd-DB-OwnCloud* (introduït al punt 2), *Database-name=owncloud*; *Hostname=localhost* i fem clic a finalitzar la instal·lació.

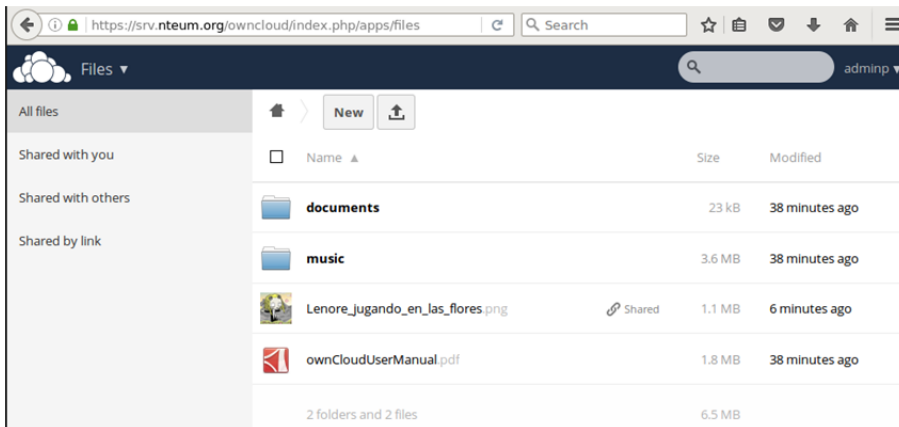
Figura 15



A partir d'aquí apareixerà la pantalla d'OwnCloud i l'avís per descarregar les *apps* per a dispositius mòbils, i ja podem interactuar amb la interfície web

pujant-hi arxius, visualitzant-los, gestionant el calendari/tasques/contactes, etc. (figura 16).

Figura 16



És important activar la connexió a través d'https per mantenir la privadesa i evitar problemes d'administració. També es recomana accedir a la interfície d'administració per configurar opcions com ara compartir documents entre diversos servidors, visualitzar documents OpenOffice/Office, permetre'ls compartir mitjançant un enllaç, forçar https, o també habilitar/deshabilitar les apps des de la pestanya corresponent (cantonada superior esquerra).

Activitats

1. Configureu un servidor Apache + SSL+ PHP+ MySQL+ PHPAdmin per a visualitzar els fulls personals dels usuaris.
2. Configureu un servidor Apache + un Reverse Proxy per a accedir al servidor web (que es troba en una màquina interna) des d'una màquina externa a través del Proxy. Amplieu l'activitat afegint un altre servidor *web* intern i feu que el *proxy* balancegi entre els dos servidors interns. Proveu diferents polítiques de balanceig.
3. Creeu i configureu un sistema de correu electrònic a través d'Exim, Fetchmail, SpamAssassin i un servidor IMAP per a rebre correus des de l'exterior i poder llegir-los des d'una màquina remota amb el client Mozilla (Thunderbird).
4. Instal·leu la wiki MoinMoin i creeu un conjunt de pàgines per verificar el seu funcionament.
5. Instal·leu el servidor de *backups* BackupPC i genereu una còpia de seguretat des d'una màquina Windows i una altra des d'una màquina Linux. Escolliu el mètode de comunicació amb els clients i justifiqueu la resposta. Realitzeu la mateixa experiència amb Bareos i obtingueu conclusions sobre eficiència, productivitat i simplicitat.
6. Configureu una CA amb TinyCA i genereu/proveu els certificats per a una pàgina web amb SSL i per enviar correus signats i verificar-los des d'un altre compte.
7. Configureu un sistema de correu (corporatiu) d'altres prestacions amb Postfix, SpamAssassin, ClamAV, Imap i RoundCube. Verifiqueu que totes les opcions i possibilitats funcionen així com l'accés via web i a través d'un client (Icedove, per exemple).
8. Configureu un servidor d'arxius amb OwnCloud. Comproveu la funcionalitat a través de la interfície web i verifiqueu la sincronització des d'un dispositiu mòbil amb l'app corresponent.
9. Creeu un Samba Active Directory Domain Controller (AD DC) i verifiqueu el seu funcionament des d'una màquina Windows.
10. Sobre Apache inseriu Mod_security i proveu les regles habituals (spam, injeccions, repetició d'usuaris, etc.)
11. Utilitzant tres màquines interconnectades (amb diferents distribucions) instal·leu un repositori OCS+GLPi i feu un inventari de les tres màquines i gestioneu les actualitzacions sobre OCS+GLPi.

Bibliografia

Tots els URL han estat visitats per última vegada el juny del 2016.

[ABench] *Apache HTTP server benchmarking tool.*

<<https://httpd.apache.org/docs/2.4/programs/ab.html>>

[AMod] *Apache Module Index.*

<<http://httpd.apache.org/docs/current/mod/>>

[AModBal] *Apache Module mod_proxy_balancer.*

<https://httpd.apache.org/docs/2.4/mod/mod_proxy_balancer.html>

[AModCach] *Apache Caching Guide.*

<<https://httpd.apache.org/docs/2.4/caching.html>>

[apa] *Apache HTTP Server Version 2.2.*

<<http://httpd.apache.org/docs/2.2/>>

[Apab] *Apache2 + WebDav.*

<<http://www.debian-administration.org/articles/285>>

[ASec] *Apache Security HandBook.*

<<https://www.feistyduck.com/library/apache%2dsecurity/online/>>

[ASSP] *Anti-Spam SMTP Proxy Server.*

<<https://sourceforge.net/projects/assp/>>

[AW] *Awstats.* <<http://www.awstats.org/>>

[AWFull] *AWFull.* <<https://launchpad.net/awfull>>

[Bareos] *Bareos Reference Manual.*

<<http://doc.bareos.org/master/html/bareos-manual-main-reference.html>>

[BareosTut] *Bareos: Tutorial.*

<<http://doc.bareos.org/master/html/bareos-manual-main-reference.html#x1-730006>>

[Bareos-WebUI] *Interfície web per a Bareos.*

<<http://doc.bareos.org/master/html/bareos-manual-main-reference.html#x1-440003>>

[BranScripts] *Apache Bench and Gnuplot: you're probably doing it wrong. Brad Landers.*

<<http://www.bradlanders.com/2013/04/15/apache-bench-and-gnuplot-youre-probably-doing-it-wrong/>>

[BXCA] *How to Create SSL Certificates*

<<https://campus.barracuda.com/product/campus/article/display/CP/30114587/>>

[Deb] **Debian.org.** *Debian Home.*

<<http://www.debian.org>>

[DebSpam] *DebianSpamAssassin.*

<<https://wiki.debian.org/DebianSpamAssassin>>

[DigOcCach] **Justin Ellingwood.** *How To Configure Apache Content Caching on Ubuntu 14.04.*

<<https://www.digitalocean.com/community/tutorials/how-to-configure-apache-content-caching-on-ubuntu-14-04>>

[DigOcModS] **Jesin A.** *How To Set Up mod_security with Apache on Debian/Ubuntu.*

<https://www.digitalocean.com/community/tutorials/how-to-set-up-mod_security-with-apache-on-debian-ubuntu>

[exim] *Exim*.

<<http://www.exim.org/docs.html>>

[GLPi] *Information Resource-Manager and Administration-Interface*.

<<http://www.glpi-project.org/spip.php?lang=en>>

[GurJm] *Free Jmeter Tutorial*.

<<http://www.guru99.com/jmeter-tutorials.html>>

[HandB] *Servicios de red: Postfix, Apache, NFS, Samba, Squid, LDAP*

<<http://debian-handbook.info/browse/es-ES/stable/network-services.html>>

[I2P] *The Invisible Internet Project*. <<https://geti2p.net/en/>>

[IET] IETF. Repositori de Request For Comment desenvolupats per Internet Engineering Task Force (IETF) al Network Information Center (NIC). <<http://www.ietf.org/rfc.html>>

[Ired] *Install iRedMail on Debian*.

<<http://www.iredmail.org/docs/install.iredmail.on.debian.ubuntu.html>>

[Ired-DNS] *Setup DNS records for your iRedMail server*.

<<http://www.iredmail.org/docs/setup.dns.html>>

[Ired-IMAP] *Configure mail client applications*.

<<http://www.iredmail.org/docs/index.html#configure-mail-client-applications>>

[Ired-Relay] *How-to: iRedmail with optional per-user freemail-addresses and relay*. <<http://www.iredmail.org/forum/topic3474-iredmail-support-howto-iredma>>

[ITE] **Instituto de Tecnologías Educativas**. *Redes de área local: Aplicaciones y Servicios Linux*.

<<http://www.ite.educacion.es/formacion/materiales/85/cd/linux/indice.htm>>

[JBF] *JBroFuzz web application fuzzer*.

<<https://www.owasp.org/index.php/JBroFuzz>>

[Jmet] *Jmeter*. <<http://jmeter.apache.org/index.html>>

[JmUM] *Jmeter User's Manual*.

<<http://jmeter.apache.org/usermanual/index.html>>

[LabRat] *OWASP Live CD Project*.

<https://www.owasp.org/index.php/Category:OWASP_Live_CD_Project/es>

[ModSec] *Mod_Security*. <<https://www.modsecurity.org/about.html>>

[ModSRef] *Mod_security Reference manual*.

<<https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual>>

[ModSBook] *ModSecurity Handbook*.

<<https://www.feistyduck.com/library/modsecurity%2dhandbook%2dfree/online/>>

[Mou] **Mourani, G.** (2001). *Securing and Optimizing Linux: The Ultimate Solution*. Open Network Architecture, Inc.

<<http://www.tldp.org/LDP/solrhe/Securing-Optimizing-Linux-The-Ultimate-Solution-v2.0.pdf>>

[OC] *Access, Sync and Share Your Data, Under Your Control*.

<<https://owncloud.org/features/>>

[OCS] *OCS Inventory NG Documentation Project*.

<http://wiki.ocsinventory-ng.org/index.php?title=Main_Page>

[OCS-Dev] *OCS Inventory Developers.*
<<https://launchpad.net/ocsinventory>>

[OProj] *OWASP LiveCD Education Project.*
<https://www.owasp.org/index.php/Category:OWASP_LiveCD_Education_Project>

[Owasp] *Use of Web Application Firewalls.*
<https://www.owasp.org/index.php/Category:OWASP_Best_Practices:_Use_of_Web_Application_Firewalls>

[pki] *PKI Public-key cryptography.*
<http://en.wikipedia.org/wiki/Public_key_cryptography >

[pkicry] *Christof Paar, Jan Pelzl Introduction to Public-Key Cryptography.*
<<http://wiki.crypto.rub.de/Buch/movies.php> >

[procmail] *ProcMail.*
<<http://www.debian-administration.org/articles/242>>

[psocks] *Proxy SOCKS.*
<<http://en.flossmanuals.net/bypassing-es/proxis-socks/> >

[PVan] *Patrick van Kouteren. Monitoring Apache with Supervisor.* <<http://www.vankouteren.eu/blog/2014/09/monitoring-apache-with-supervisord/>>

[s40] *Wiki - Samba.*
<<https://wiki.samba.org> >

[s41] *RSAT. Remote Server Administration Tools on a Windows workstation.*
<https://wiki.samba.org/index.php/Installing_RSAT_on_Windows_for_AD_Management>

[s42] *M. López, C. Alonso Samba 4: Controlador Active Directory.*
<<http://waytoit.wordpress.com/2013/05/12/samba-4-controlador-active-directory-parte-1-de-3/> >

[s43] *M. Rushing. Compiling Samba 4 on Debian Wheezy - Active Directory Domain Controllers..*
<<http://mark.orbum.net/2014/02/22/compiling-samba-4-on-debian-wheezy-active-directory-domain-controllers-ho/> >

[s44] *Guía Samba4 como Controlador de Dominio y Directorio Activo.*
<<http://fraterneo.wordpress.com/2013/08/19/guia-samba4-como-controlador-de-dominio-y-directorio-activo-actualizacion/> >

[Sam] *Samba Active Directory Domain Controller.*
<https://wiki.samba.org/index.php/Setup_a_Samba_Active_Directory_Domain_Controller>

[SerMail] *Install Postfix.*
<http://www.server-world.info/en/note?os=Debian_8&p=mail>

[SerWorld] *Mail Log Analyzer : AWstats.*
<http://www.server-world.info/en/note?os=Debian_8&p=mail&f=9>

[socks] *Greg Ferro Fast Introduction to SOCKS Proxy.*
<<http://etherealmind.com/fast-introduction-to-socks-proxy/> >

[squid] *Squid Proxy Server.*
<<http://www.squid-cache.org/> >

[squide] *Squid Configuration Examples.*
<<http://wiki.squid-cache.org/ConfigExamples> >

[StartSSL] *Creating a TLS encryption key and certificate. Christoph Haas.*
<<https://workaround.org/ispmail/jessie/create-certificate>>

[Sup] *Supervisor. A Process Control System.*
<<http://supervisord.org/index.html>>

[SW] *Server-World.*
<http://www.server-world.info/en/note?os=Debian_8>

[tinycal] **Magnus Runesson** (2007). *Create your own CA with TinyCA2.*
<<http://theworldofapenguin.blogspot.com.es/2007/06/create-your-own-ca-with-tinycal-part-1.html>>

[Tor] *Tor Project.* <<https://www.torproject.org/>>

[tproxy] **Kiracofe, D.** *Transparent Proxy with Linux and Squid mini-HOWTO –obs:EOL però interessant en conceptes–.*
<<http://tldp.org/HOWTO/TransparentProxy.html#toc1>>

[TutPointJm] *jMeter Tutorial.*
<<http://www.tutorialspoint.com/jmeter/index.htm>>

[WA] *WebAlizer.* <<http://www.webalizer.org/>>

[WGoat] *WebGoat 7.0.1 Release.*
<<https://github.com/WebGoat/WebGoat/releases>>

[WorkMail] *ISPmail guide for Debian Jessie. Christoph Haas.*
<<https://workaround.org/ispmail/jessie>>

[XCA] *X Certificate and key management.*
<<http://xca.sourceforge.net/xca.html#toc15>>

[ZAP] *Zed Attack Proxy.*
<https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project>