13. SERVERUL FTP

13.1. Protocolul FTP

13.1.1. Descriere generala

FTP este un protocol client-server ce faciliteaza transferul si manipularea fisierelor intre doua statii conectate intr-o retea TCP/IP. Statia care joaca rolul de server partajeaza in retea portiuni ale sistemului sau de fisiere; ea deschide un port (default: portul 21 TCP) pe care asteapta cereri de conectare din partea clientilor. Clientii se conecteaza pe portul serverului, trimitandu-i acestuia comenzi corespunzatoare diferitelor operatii dorite.

13.1.2. Clientul FTP

Clientul FTP este o aplicatie ce implementeaza protocolul FTP, conectandu-se la un server FTP si trimitandu-i acestuia comenzi. Comenzile pot solicita operatii precum:

- download de fisiere de pe server, cu posibilitatea de *resume* (continuare a transferului din punctul in care a fost intrerupt)
- upload de fisiere pe server
- creare, stergere, redenumire, schimbare de permisiuni ale fisierelor de pe server

Clientii pot fi:

- aplicatii de linie de comanda. Exemple de aplicatii Linux: ftp, ncftp, lftp, wget, mc
- aplicatii grafice. Exemplu Linux: gFTP, FileZilla

Nota: *diversele browsere (Opera,Firefox, Epiphany etc) au si ele capabilitati partiale de client FTP, ele permitandu-i utilizatorului sa se conecteze la un server FTP, sa navigheze prin structura de directoare partajata de acesta si sa download-eze fisere. Lipsesc in general facilitatile de upload si de modificare a fisierelor aflate pe server.*

13.1.3. Serverul FTP

Serverul FTP este o aplicatie ce implementeaza protocolul FTP, permitand accesul controlat al clientilor la portiuni ale sistemului său de fisiere, in scopul simplei vizualizari sau al producerii de modificari.

Serverul poate functiona in doua moduri:

- cu autentificare – clientii trebuie sa furnizeze o combinatie user/parola valida pentru a putea accesa resursele oferite de catre server
- anonim – este permis accesul oricarui client, in conditiile in care clientul furnizeaza username-ul *anonymous* si o parola oarecare (traditional, se foloseste ca parola adresa de email a utilizatorului)

De la inceputuri si pana in ziua de astazi, serverul ii asigura clientului acces la directorul personal al unui utilizator de pe masina server. Vom vedea ca acest lucru se intampla atat atunci cand clientul furnizeaza o combinatie user/parola, cat si atunci cand serverul ofera clientilor sai acces anonim.

Ca software de server FTP disponibil in Linux/Unix, amintim:

- **vsftpd** – un server FTP axat pe securitate si viteza, cu o lista de facilitati medie
- **proftpd** – server FTP caracterizat prin flexibilitate in configurare si bogatie de facilitati
- **pureftpd** – un server sigur si cu configurare facila

Acest material prezinta serverul *pure-ftpd*.

13.1.4. Detalii de functionare a protocolului

Portul traditional folosit de serverul FTP este 21, insa el poate fi modificat din configurarea serverului. Cand un client doreste sa acceseze resursele serverului, el se va conecta initial pe acest port, deschizand asa-numita *conexiune de control*, prin care circula comenzile trimise de catre client serverului. Pentru fiecare transfer de date efectuat intre client si server (ex: upload/download de fisier, listing de director etc) este deschisa cate o conexiune de date, separata de cea de control. Felul in care este deschisa aceasta conexiune depinde de modul in care comunica cele doua parti – activ sau pasiv.

In **modul activ**, atunci cand urmeaza sa transfere date, clientul deschide un port, i-l comunica serverului si asteapta ca acesta sa se conecteze la el. Serverul se va conecta la portul specificat de client, folosind ca port sursa 20. Acest mod de lucru nu este practic atunci cand clientul se afla in spatele unui firewall care nu permite cererilor de conexiune din exterior sa patrunda in retea; intr-un astfel de caz este recomandabil modul pasiv.

In **modul pasiv**, tot clientul deschide conexiunea de control, insa in cazul celei de date serverul este cel care deschide portul, i-l comunica clientului si acesta initiaza conexiunea. Asadar ambele conexiuni (control+date) sunt initiate de client, fapt util atunci cand conexiunile trec printr-un firewall.

In ambele moduri, odata conexiunea de date realizata, ea este folosita pentru transferarea informatiei intre server si client.

13.2. Serverul pure-ftpd

13.2.1. Capabilitati

Pure-ftpd este un soft de server FTP sigur, stabil si usor configurabil. Printre facilitatile sale se numara:

- permite accesul autentificat sau anonim
- utilizatorii, odata conectati, pot fi blocati in directorul lor personal (chroot)
- permite definirea de servere virtuale (in functie de IP-ul serverului pe care soseste cererea clientului)
- autentificarea utilizatorilor se poate realiza folosind diferite back-end-uri (fisiere de sistem, fisiere locale, PAM (Pluggable Authentication Modules), LDAP, MySQL etc)
- limitarea numarului de conexiuni global si per-IP
- limitarea vitezei de transfer a clientilor pentru upload si download
- suport SSL/TLS

13.2.2. Instalare

13.2.2.1. Compilare din surse

Sursele serverului sunt disponibile pe http://www.pureftpd.org/project/pure-ftpd/download sau pe unul dintre mirror-urile locale, romanesti.

Sursele dispun de obisnuitul script *configure*; lista de optiuni posibile poate fi obtinuta cu comanda *./configure --help*.

```
./configure --without-inetd --with-puredb --with-throttling --with-ftpwho \
        --with-peruserlimits --with-mysql --with-privsep –prefix=/pureftp
make -j2
make install
```

13.2.2.2. Instalare din pachete precompilate

In functie de distributia Linux folosita vom gasi unul sau mai multe pachete corespunzatoare serverului pure-ftpd. Spre exemplu, in Fedora sau SuSe exista un singur pachet numit *pure-ftpd*, pe cand in familia Debian avem pachetele de baza *pure-ftpd* si *pure-ftpd-common*, pe langa care exista pachete aditionale corespunzatoare diverselor baze de date de autentificare suplimentare: *pure-ftpd-mysql*, *pure-ftpd-postgresql*, *pure-ftpd-ldap*.

Debian:

```
apt-get install pure-ftpd # instaleaza automat si pure-ftpd-common
```

Fedora:

```
yum install pure-ftpd
```

SuSe:

```
zypper install pure-ftpd
```

13.2.3. Informatii utile

Serverul *pure-ftpd* introduce urmatoarele executabile/utilitare:

- **pure-ftpd** – daemonul FTP. Spre deosebire de alte servere, nu se obisnuieste ca acest executabil sa fie apelat direct, din cauza particularitatilor legate de configurarea sa (vezi sectiunea urmatoare)
- **pure-ftpwho** – utilitar folosit pentru afisarea tabelara a utilizatorilor logati si a activitatii lor in momentul rularii utilitarului
- **pure-pw** – utilitar folosit pentru managementul bazelor de date de autentificare, atunci cand acestea sunt memorate in fisiere locale cu extensia *.pdb*

13.2.4. Particularitati legate de configurare

Spre deosebire de majoritatea serverelor, *pure-ftpd* nu foloseste din oficiu un fisier de configurare, ci a fost gandit sa fie configurat prin optiuni pasate direct executabilului *pure-ftpd* in linia de comanda la pornire. Acesta este si motivul pentru care nu are sens apelarea directa a executabilului fara optiuni: el nu cauta un fisier de configurare implicit.

Pentru a configura totusi serverul in modul traditional, au fost dezvoltate scripturi de pornire a serverului (wrapper scripts), care interpreteaza directive aflate in unul sau mai multe fisiere de configurare si le transforma in optiuni pasate executabilului *pure-ftpd*. Felul in care este structurata configurarea poate diferi, existand doua abordari posibile in functie de distributie:

- abordarea clasica: unele distributii folosesc un singur fisier de configurare in care se gasesc directivele. Spre exemplu, in Fedora exista fisierul **/etc/pure-ftpd/pure-ftpd.conf**; wrapper script-ul care il parseaza si apeleaza serverul cu optiunile corecte este */usr/sbin/pure-config.pl*. Iata un exemplu de fragment de configurare:

```
VerboseLog yes
UnixAuthentication no
```

- abordarea Debian: exista directorul **/etc/pure-ftpd/conf** in care, pentru fiecare directiva ce se doreste modificata, trebuie creat un fisier cu numele directivei, iar continutul fisierului va fi valoarea directivei. Wrapper-script-ul se numeste *pure-ftpd-wrapper* si are propriul manpage.
  Pentru a realiza configurarea echivalenta cu exemplul anterior din Fedora, va trebui sa cream in */etc/pure-ftpd/conf* un fisier numit **VerboseLog** ce contine cuvantul **yes** si un fisier **UnixAuthentication** ce contine cuvantul **no**. *Atentie! Numele de directive (si implicit de fisiere) sunt case-sensitive!*

Nota: exista si utilitare grafice de administrare a serverului *pure-ftpd*. Spre exemplu, in Ubuntu exista pachetul *pureadmin*.

In sectiunile urmatoare ale materialului, exemplele de configurare vor folosi directive de configurare asa cum apar ele in fisierele citite de catre wrapper script, fara a se insista pe optiunile de executabil ce le corespund.

13.2.5. Configurare de baza

13.2.5.1. Elemente generale

Configurarea de inceput presupune cateva elemente generale de configurare:

- gradul de detaliu al logurilor. Se poate stabili cu directiva **VerboseLog** (valori posibile *yes* sau *no*). In cazul configurarii clasice (cu directivele plasate intr-un singur fisier) directiva VerboseLog poate fi folosita de doua ori pentru a creste gradul de detaliu
- activarea sau dezactivarea rezolutiei DNS inverse. Implicit, la conectarea unui client, serverul incearca sa determine numele DNS corespunzator adresei acestuia pentru a-l consemna in loguri. Operatia consuma timp suplimentar si deseori se soldeaza cu timeout, ceea ce duce la o intarziere a procesului de login. Rezolutie DNS inversa poate fi dezactivata folosind directiva **DontResolve** (valori posibile *yes* sau *no*)
- daca serverul FTP sa creeze automat directoarele personale ale utilizatorilor atunci cand ele nu exista. Directiva folosita este **CreateHomeDir** (valori posibile *yes* sau *no*)

```
VerboseLog yes     # logging detaliat
DontResolve yes    # fara rezolutie DNS inversa a adreselor IP ale clientilor
CreateHomeDir yes # directoarele personale sa fie create automat daca nu exista
```

13.2.5.2. Securizare

Iata cateva aspecte ce tin de securizarea minimala:

- **gestionarea accesului la fisierele ascunse**. Directivele utilizate sunt ***ProhibitDotFilesRead***, ***ProhibitDotFilesWrite*** si ***DisplayDotFiles***; toate au valori posibile *yes* sau *no*. Nu exista o configurare recomandabila universal; iata insa doua situatii tipice:

  - serverul FTP este folosit pentru a oferi acces de la distanta la directorul personal al userilor. In directorul in cauza se gasesc o multitudine de fisiere ascunse, care reprezinta de obicei configurari per-user ale diferitelor aplicatii din sistem. Intr-un astfel de caz am putea dori, din motive de protectie, sa interzicem citirea si modificarea acestora, sau poate chiar si afisarea lor in listing-urile de directoare:

```
ProhibitDotFilesWrite yes  # clientul nu poate modifica fisierele care incep cu .
ProhibitDotFilesRead yes   # clientul nu poate citi fisierele care incep cu .
DisplayDotFiles no         # clientul nu vede fisierele al caror nume incepe cu .
```

- serverul FTP este folosit pe o masina a unei companii de hosting, pentru a le da posibilitatea clientilor de a-si uploada site-urile. In astfel de cazuri exista deseori nevoia ca fisierele ascunse sa fie vizibile si editabile; spre exemplu, atunci cand serverul web folosit este Apache, clientii trebuie sa poata vedea/crea/edita fisiere de configurare per-director numite *.htaccess*.

```
ProhibitDotFilesWrite no
ProhibitDotFilesRead no
DisplayDotFiles yes
```

- **stabilirea permisiunilor implicite pe fisierele nou create.** Se realizeaza cu directiva ***Umask*** urmata de doua valori - prima pentru fisiere, a doua pentru directoare. Principiul este acelasi al umask-ului din shell-urile Linux.

```
Umask      027:027            # stabileste permisiunile pentru fisierele nou-create de clienti
```

- **protejarea serverului impotriva unor incercari de atacuri**. Una dintre categoriile de atacuri des intalnite este Denial of Service (DoS), care presupune degradarea sau intreruperea serviciului oferit prin epuizarea resurselor serverului (procesor, memorie, spatiu pe HDD etc). Este necesara stabilirea de limite:

```
MaxIdleTime             5      # nr de minute de inactivitate dupa care este inchisa conexiunea
MaxClientsNumber 20     # nr total de clienti
MaxClientsPerIP  10     # nr de clienti cu acelasi IP (ex: aflati in spatele unui NAT)
MaxDiskUsage            99     # exprimat in procente din spatiul partitiei respective
```

- **minimizarea pagubelor in cazul unui atac reusit**, care a preluat controlul asupra unuia dintre procesele serverului. Presupune rularea serverului cu privilegii minime si blocarea fiecarui proces al serverului intr-un director, astfel incat un atacator care preia controlul asupra unuia dintre procesele serverului sa nu aiba acces la restul sistemului de fisiere si nici privilegiile necesare pentru a produce pagube suplimentare (ex: sa afecteze alte servicii ce ruleaza pe aceeasi statie sau sa modifice fisiere de configurare)

```
ChrootEveryone   yes    # blocarea tuturor utilizatorilor in directorul lor personal
```

Nota: *dupa cum se va vedea ulterior, in cazul lucrului cu useri virtuali se poate stabili la nivel de fiecare user in parte daca el va fi sau nu blocat in directorul sau personal.*

13.2.6. Stabilirea modelului de autentificare folosit

Serverul FTP poate functiona in urmatoarele moduri:

- **exclusiv autentificat**. Orice client trebuie sa furnizeze o combinatie user-parola valida pentru a accesa resursele serverului; in urma unei autentificari reusite, clientul este plasat in directorul personal corespunzator username-ului (identitatii) cu care s-a autentificat
- **exclusiv anonim**. Un astfel de server permite oricarui client accesul la resursele sale. Clientii vor fi plasati automat in directorul personal al unui anumit user configurat de catre administrator ca fiind cel corespunzator accesului anonim (vezi sectiunea urmatoare)
- **anonim+autentificat**. Clientii care furnizeaza un username valid si parola corecta vor fi plasati in directorul personal al acelui user; clientii care doresc acces anonim vor furniza username-ul *anonymous* insotit de o parola oarecare si vor fi plasati in directorul personal al userului ce corespunde accesului anonim

Stabilirea modului de lucru al serverului se realizeaza cu ajutorul a doua directive, ambele cu valori posibile *yes* sau *no*:

- ***AnonymousOnly***. Pus pe *yes* permite numai accesul anonim, pus pe *no* accepta si userii autentificati corect
- ***NoAnonymous***. Pus pe *yes* interzice accesul anonim, pus pe *no* il accepta

Iata mai jos valorile acestor directive in functie de modul de lucru al serverului:

Mod de lucru server	Valoare directiva NoAnonymous	Valoare directiva AnonymousOnly
Exclusiv autentificat	yes	no
Exclusiv anonim	no	yes
Anonim+autentificat	no	no

Nota: *cea de-a parta combinatie ramasa (yes/yes) nu este posibila deoarece creeaza o contradictie.*

13.2.7. Configurarea accesului anonim

Dupa cum s-a explicat anterior, serverul poate permite accesul anonim in doua moduri de functionare: in cel exclusiv anonim si in cel anonim+autentificat. In ambele cazuri principiile configurarii accesului anonim sunt aceleasi, diferind doar valorile directivelor *AnonymousOnly* si *NoAnonymous*.

Crearea unui server ce permite acces anonim este utila, de exemplu, atunci cand dorim doar sa distribuim fisiere catre un numar mare de clienti. Din motive de securitate, pentru servere FTP publice este dezactivat upload-ul si modificarea fisierelor, insa pe servere mai specializate (ex: un server FTP intern al unei companii) ele pot fi permise.

In pure-ftpd, fisierele disponibile clientilor anonimi sunt plasate in directorul personal al userului **ftp**. Pentru fiecare client care se conecteaza, in sistemul de operare al statiei server este pornit un nou proces pure-ftpd care va rula cu UID/GID contului *ftp*, iar clientul va fi plasat automat in directorul personal al acestui user. In consecinta, trebuie sa avem grija ca acest user, impreuna cu directorul sau personal, sa existe inainte de a porni serverul. In cazul instalarii serverului din pachete, ambele sunt create automat la instalare.

Iata cateva directive de configurare ce gestioneaza accesul anonim atunci cand acesta este permis:

```
AnonymousCanCreateDirs no         # au clientii dreptul de a crea directoare?
AnonymousCantUpload yes           # au clientii dreptul de a uploada fisiere?
AnonymousBandwidth  100           # viteza de transfer maxima per client, masurata in KB/s
```

13.2.8. Accesul autentificat

13.2.8.1. Concepte

Protocolul FTP permite serverului sa verifice identitatea clientilor inainte de a le permite accesul in sistem. Pentru aceasta este necesar ca serverul sa dispuna de una sau mai multe baze de date cu utilizatori si parole – asa-numitele *back-end*-uri de autentificare. Pot fi folosite in acest scop:

- fisierele de sistem Linux/Unix (*passwd* si *shadow*) – in acest caz se spune ca lucram cu “useri de sistem”. A crea un nou user FTP se reduce la a crea un nou user de sistem cu comanda *useradd*
- alte surse de date decat fisierele de sistem. In acest caz, spunem ca lucram cu „useri virtuali”. Variante:
  - fisiere text, definite de catre administrator
  - servere de baze de date SQL (MySQL, PostgreSQL)
  - servere LDAP

Se pot folosi simultan mai multe backend-uri de autentificare; ele sunt incercate in ordinea in care apar in fisierul de configurare. De exemplu, daca s-a configurat intai SQL si apoi Unix, se va incerca intai autentificare SQL; daca ea reuseste, nu mai este consultat alt backend. Daca ea esueaza, exista doua situatii posibile:

- username-ul nu exista in baza de date SQL. In acest caz va fi incercat si backend-ul urmator (*passwd*/*shadow*)
- username-ul exista in baza de date SQL, dar parola este gresita. In aceasta situatie nu mai este incercat alt backend de autetificare, intregul proces considerandu-se esuat

Asadar, cand un username nu este gasit intr-un backend, se trece la backend-ul urmator. Daca in schimb userul este gasit dar parola este incorecta, se considera autentificare esuata si nu se mai incearca alte backenduri.

13.2.8.2. Particularitati Debian

Dupa cum s-a vazut si anterior, in distributiile din familia Debian exista o abordare particulara a configurarii: cea a crearii cate unui fisier pentru fiecare directiva modificata. In aceste conditii, daca se utilizeaza mai multe back-end-uri de autentificare, administratorul nu ar mai putea stabili ordinea utilizarii lor. De aceea a fost facut urmatorul artificiu: directorul */etc/pure-ftpd/auth* contine cate un symlink pentru fiecare backend activat. Numele symlink-ului este de forma **NNbackend** (ex: **65unix**), unde numarul de la inceput determina ordinea in care sunt consultate backendurile in momentul autentificarii. Fiecare astfel de fisier symlink trimite catre directiva corespunzatoare acelui backend din directorul */etc/pure-ftpd/conf*.

Exemplu: pentru a activa autentificarea din fisierele de sistem, avem de efectuat doua operatii:

- creare/editarea fisierului */etc/pure-ftpd/conf/UnixAuthentication*, scriind in acesta cuvantul *yes*
- crearea unui symlink in */etc/pure-ftpd/auth* (daca nu exista deja!) care sa pointeze catre */etc/pure-ftpd/conf/UnixAuthentication*. Numele fisierului symlink va fi de forma *45unix*. In cazul in care folosim mai multe backenduri de autentificare, vom alege pentru fiecare numarul de asa natura incat sa rezulte ordinea dorita

Atentie! Absenta symlink-ului din directorul auth atrage dupa sine nefunctionarea acelui backend, chiar daca el a fost activat corect folosind fisierele din subdirectorul /etc/pure-ftpd/conf!

13.2.8.3. Autentificarea folosind conturi de sistem

In acest scenariu, bazele de date sunt cele de sistem: */etc/passwd* si */etc/shadow*. Manipularea conturilor de utilizator se face cu comenzile clasice: *useradd*, *userdel*, *usermod*, *passwd*. Ca urmare a login-ului se porneste un proces pure-ftpd care acceseaza/creeaza fisiere cu UID-ul si GID-ul din */etc/passwd* ale acelui user, iar clientul este plasat automat in directorul sau personal.

Activarea autentificarii din bazele de date de sistem de realizeaza in *pure-ftpd* cu directiva:

```
UnixAuthentication      yes
```

In plus, in distributiile din familia Debian trebuie sa nu uitam sa cream si symlink-ul corespunzator in subdirectorul */etc/pure-ftpd/auth*.

13.2.8.4. Autentificare folosind useri virtuali

13.2.8.4.1. Conceptul de user virtual si particularitati

Useri virtuali sunt numiti cei memorati in alte baze de date decat cele de sistem (*passwd/shadow*). Acestia pot fi stocati in:

- fisiere locale. Serverul *pure-ftpd* permite memorarea userilor virtuali in fisiere binare cu extensia *.pdb* (pure database), care se manipuleaza prin intermediul utilitarului **pure-pw** (vezi mai jos)
- servere SQL (MySQL, postgresql) sau LDAP

Lucrul cu useri virtuali este util deoarece baza de date de sistem */etc/passwd* are un numar limitat, fix, de campuri - si anume cele necesare login-ului in sistemul de operare. In cazul FTP insa, un user poate avea o multitudine de caracteristici suplimentare, care nu pot fi incluse in passwd: limita de viteza la upload/download, adrese IP de pe care i se permite conectarea, interval orar in care are accesul permis etc. Bazele de date cu useri virtuali permit prezenta tuturor acestor caracteristici. In plus, editarea unei baze de date cu useri virtuali este posibil sa nu solicite privilegii de root, asa cum se intampla in cazul lui *passwd/shadow*.

Pentru fiecare user care se autentifica, *pure-ftpd* efectueaza urmatoarele operatii:

- verifica daca userul se regaseste in backend-urile de autentificare configurate de administrator si daca parola furnizata este cea corecta
- in caz afirmativ, porneste un nou proces server, care ruleaza UID si GID ale userului virtual proaspat autentificat. Aceste UID/GID sunt cele folosite pentru a accesa fisiere in sistem
- noul proces server plaseaza clientul in directorul său personal

Operatiile enumerate mai sus presupun ca serverul FTP sa cunoasca UID, GID, username, parola si home directory pentru fiecare user virtual in parte. Daca, in cazul lucrului cu useri de sistem, aceste informatii erau furnizate de catre */etc/passwd/* si */etc/shadow*, repezentand informatii standard asociate conturilor de sistem, in cazul userilor virtuali administratorul serverului trebuie sa se asigure ca bazele de date folosite pentru autentificare contin toate aceste detalii, si in plus ca directoarele personale ale utilizatorilor exista si au permisiunile corecte.

Nota: *pure-ftpd dispune de optiunea CreateHomeDir care degreveaza administratorul de aceasta ultima obligatie.*

Atentie! Procesele pure-ftpd pornite la logarea userilor virtuali ruleaza cu UID/GID specificat in baza de date virtuala – chiar daca acestea nu exista in /etc/passwd! Exista doua consecinte:

- fisierele create de acesti useri vor avea ca owner si group chiar aceasta pereche UID/GID; comanda *ls -l* aplicata unui astfel de fisier va afisa numai ID-urile, deoarece ele nu au username/groupname corespondent in */etc/passwd* si */etc/group*:

```
$ ls -l /var/ftp/user1/eu.jpg
-rw-r--r-- 1 4002 4000 34096 2015-03-22 12:33 eu.jpg
```

- directoarele personale ale userilor virtuali vor avea ca owner chiar UID-ul userului virtual, astfel incat acesta sa aiba drepturile necesare in directoarele lor. Comanda *chmod* permite stabilirea ownerului si in forma numerica, chiar daca el nu exista in */etc/passwd*:

```
root@server# mkdir /var/ftp/user1
root@server# chown 10001 /var/ftp/user1
root@server# ls -ld /var/ftp/user1
drwxr-xr-x 2 4002 4000 4096 2015-03-22 12:33 /var/ftp/user1
```

Pentru a lucra cu useri virtuali este necesara configurarea urmatoarelor aspecte:

- stabilirea back-end-urilor de autentificare folosite de server, in ordinea corecta
- popularea bazelor de date cu informatiile utilizatorilor
- crearea directoarelor personale ale utilizatorilor, cu ownership si permisiuni corecte

Prezentam in continuare modul de configurare a serverului pentru a folosi useri virtuali memorati intr-un fisier local *.pdb*.

13.2.8.4.2. Useri virtuali memorati in fisiere .pdb locale

Bazele de date in format *.pdb* se administreaza cu ajutorul utilitarului **pure-pw**. Acesta are urmatoarea sintaxa generala:

```
pure-pw comanda argumente optiuni
```

Iata principalele comenzi suportate de *pure-pw*:

- **useradd**, **userdel**, **usermod** - folosite pentru adaugare/stergere/modificare useri virtuali
- **passwd** - schimba parola unui user virtual existent
- **show** - listeaza setul complet de caracteristici ale unui user virtual
- **mkdb** - genereaza fisierul *.pdb* pe baza celui in format text (vezi mai jos)

Lucrul cu *pure-pw* presupune doua etape:

1. crearea/modificarea unui fisier text ce contine baza de date cu toate detaliile userilor. Toate comenzile *pure-pw* actioneaza asupra acestui fisier
2. pe baza fisierului text se genereaza fisierul binar cu extensia *.pdb*, care este cel folosit de fapt de catre server la autentificare

Locatia si numele ambelor fisiere sunt la alegerea administratorului, dupa cum se va vedea in continuare.

Atentie! Ori de cate ori produceti modificari in fisierul text, nu uitati sa re-compilati baza de date, generand noua versiune a fisierului .pdb!

Iata modul de realizare a operatiilor de administrare a userilor virtuali memorati in fisiere .pdb:

- adaugarea userilor virtuali in baza de date text:

```
pure-pw useradd user1 -f /etc/pure-ftpd/pure-ftpd.passwd -u 10004 -g 20000 -d /var/ftp/user1
```

Al doilea argument (cel de dupa operatia dorita, useradd) reprezinta username-ul. Optiunile pasate lui *pure-pw* au aceleasi semnificatii ca in cazul utilitarului de sistem useradd: -u specifica UID, -g specifica GID, -d directorul personal. Optiunea -f stabileste calea catre baza de date text, cea in care va fi adaugat userul. La executarea comenzii va fi solicitata interactiv parola utilizatorului creat. In fisier, inregistrarea rezultanta va fi de forma urmatoare:

```
user1:$1$7OxAq9O0$ClbbAytSlQ7ao8.UXEDMI1:10004:20000::/var/ftp/user1/./::::::::::::
```

Atentie la urmatoarele aspecte:

- directorul personal nu este creat automat la adaugarea userului! El trebuie creat de catre administrator, stabilind ownership-ul si permisiunile corecte, conform celor explicate in 13.2.8.4.1
- chiar daca UID si GID se pot suprapune cu unele existente in bazele de date de sistem, acest lucru este contraindicat. Dupa cum se stie, un user este identificat in sistem dupa UID-ul sau; daca un user

virtual are un UID identic cu al unuia din *passwd*, procesul ce deserveste clientul FTP va avea aceleasi privilegii ca userul de sistem corespunzator!

- optiunea -d face ca userul sa fie blocat in directorul său personal; a se observa secventa /./ din baza de date text, care indica directorul in care userul este tinut captiv. Alternativ poate fi folosita optiunea -D pentru specificarea directorului personal, caz in care userul este plasat in directorul personal la login insa il poate parasi ulterior

- stergerea unui user din baza de date text:

```
pure-pw userdel user1 -f /etc/pure-ftpd/pure-ftpd.passwd
```

- schimbarea parolei unui user:

```
pure-pw passwd user1 -f /etc/pure-ftpd/pure-ftpd.passwd
```

La executarea comenzii va fi solicitata noua parola a userului in cauza.

- generarea fisierului *.pdb* pe baza fisierului text populat anterior:

```
pure-pw mkdb /etc/pure-ftpd/pureftpd.pdb -f /etc/pure-ftpd/pure-ftpd.passwd
```

- vizualizarea caracteristicilor unui user:

```
root@server:/etc/pure-ftpd# pure-pw show user1 -f /etc/pure-ftpd/pure-ftpd.passwd

Login              : user1
Password           : $1$mrCo39/0$TbwwB7.9mnxDvLoLYumQi.
UID                : 10001 (-)
GID                : 10001 (-)
Directory          : /tmp/./
Full name          :
Download bandwidth : 0 Kb (unlimited)
Upload   bandwidth : 0 Kb (unlimited)
Max files          : 0 (unlimited)
Max size           : 0 Mb (unlimited)
Ratio              : 0:0 (unlimited:unlimited)
Allowed local  IPs :
Denied  local  IPs :
Allowed client IPs :
Denied  client IPs :
Time restrictions  : 0000-0000 (unlimited)
Max sim sessions   : 0 (unlimited)
```

Odata baza de date *.pdb* generata, este necesara configurarea serverului *pure-ftpd* sa o foloseasca, dupa cum urmeaza:

- directiva de configurare **PureDB** trebuie sa aiba ca valoare calea catre fisierul *.pdb* creat:

```
PureDB /etc/pure-ftpd/pureftpd.pdb      # calea catre fisierul PDB folosit pt autentificare
```

- in cazul unei distributii din familia Debian trebuie creat symlink-ul corespunzator in directorul */etc/pure-ftpd/auth*

13.3. Clienti FTP si utilizarea lor pentru diagnosticarea serverului

Exista o multitudine de clienti FTP, care cad sub incidenta a doua mari (dar previzibile) categorii:

- *clienti grafici*, utilizabili atunci cand masina client beneficiaza de interfata grafica. Amintim aici **filezilla** si **gFTP** - care sunt clienti FTP dedicati - dar si browserele internet care au capabilitati de login si download de fisiere FTP (suportul de upload fiind insa limitat in cazul unora dintre ele)
- *clienti pentru linia de comanda*. Amintim **ftp** (clientul clasic), **ncftp**, **lftp** etc. dar si facilitati built-in ale unor programe celebre, cum ar fi *mc* (Midnight Commander)

Clientii FTP utilizabili din terminal/consola sunt unelte interactive care ii permit userului/administratorului sa "dialogheze" cu un server FTP. Ei primesc ca argument la apelare numele sau adresa IP a serverului FTP dorit si ofera un prompt in care utilizatorul poate scrie comenzi:

```
student@server$ ftp ftp.linux.ro
> aici scriem comenzi interactiv
```

Iata cateva comenzi *ftp* utile:

- **ls -** listeaza continutul directorului curent de pe server
- **get *numefisier*** - descarca fisierul cu numele specificat din directorul curent de pe server, salvandu-l in directorul curent de pe client
- **mget *sablon_nume*** - multiple get. Analog cu get, insa permite descarcarea de fisiere multiple prin utilizarea de wilcard-uri (ex: *mget *.jpg*)
- **put *numefisier*** - uploadeaza fisierul cu numele specificat din directorul curent de pe client in directorul curent de pe server
- **mput *sablon_nume*** - analog cu *mget* dar pentru upload
- **cd *cale*** - schimba directorul curent de pe server
- **lcd *cale*** - schimba directorul curent de pe client. Implicit directorul curent al clientului este cel din care s-a rulat comanda *ftp*
- **quit** - paraseste sesiunea FTP

Iata un exemplu:

```
student@server:~$ ftp ftp.lug.ro
Connected to ftp.ines.lug.ro.
220 (vsFTPd 2.3.2)
Name (ftp.lug.ro:student): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd ubuntu-releases/15.10
250 Directory successfully changed.
ftp> ls
[...output prea bogat pentru a fi reprodus aici...]
ftp> get MD5SUMS
local: MD5SUMS remote: MD5SUMS
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for MD5SUMS (256 bytes).
226 Transfer complete.
256 bytes received in 0.02 secs (15.0 kB/s)
ftp> quit
221 Goodbye.
```

13.4. BIBLIOGRAFIE

- Protocolul FTP: https://en.wikipedia.org/wiki/File_Transfer_Protocol
- Documentatia oficiala pure-ftpd: https://www.pureftpd.org/project/pure-ftpd/doc
  - compilare si configurare: https://download.pureftpd.org/pub/pure-ftpd/doc/README
  - folosirea unui unic fisier de configurare: https://download.pureftpd.org/pub/pure-ftpd/doc/README.Configuration-File
  - lucrul cu useri virtuali: https://download.pureftpd.org/pub/pure-ftpd/doc/README.Virtual-Users
- utilizarea clientului *ftp*: http://www.tldp.org/HOWTO/FTP-3.html

13.5. ANEXA 1 - exemplu de implementare cu useri virtuali memorati intr-o baza de date SQL

In cazul in care dorim sa memoram conturile de utilizator intr-o baza de date MySQL, operatiile necesare sunt urmatoarele:

– crearea unei baze de date si a unei tabele ce va memora informatiille utilizatorilor. Iata un exemplu de structura de tabela SQL:

```
CREATE TABLE useri (
  User VARCHAR(16) BINARY NOT NULL PRIMARY KEY,
  Password VARCHAR(64) BINARY NOT NULL,
  Uid INT(11) NOT NULL default '-1',
  Gid INT(11) NOT NULL default '-1',
  Dir VARCHAR(128) BINARY NOT NULL,
);
```

– popularea respectivei baze de date, folosind linie de comanda SQL sau clienti grafici (ex: mysql-query-browser, phpmyadmin etc)

– configurarea serverului pureftpd sa foloseasca MySQL ca back-end de autentificare:

```
# calea catre fisierul de configurare aditional cu configurarile de interfatare cu MySQL
MySQLConfigFile   /etc/pure-ftpd/pureftpd-mysql.conf
NoAnonymous yes          # optional – numai daca dorim acces exclusiv autentificat
UserBandwidth     200    # optional, in KB/s
```

– scrierea fisierului de configurare aditional, in care se specifica detaliile de conectare la baza de date, formatul parolelor si interogarile necesare pentru obtinerea informatiilor utilizatorilor:

```
MYSQLServer     localhost
MYSQLPort       3306
#MYSQLSocket     /var/lib/mysql/mysql.sock  # daca serverul ruleaza pe aceeasi masina
MYSQLUser       pureftpd
MYSQLPassword   PurePass
MYSQLDatabase   pureftpd
MYSQLCrypt      cleartext
MYSQLGetPW      SELECT Password FROM users WHERE User="\L"
MYSQLGetUID     SELECT Uid FROM users WHERE User="\L"
MYSQLGetGID     SELECT Gid FROM users WHERE User="\L"
MYSQLGetDir     SELECT Dir FROM users WHERE User="\L"
```

ATENTIE!

– Nu este permisa conectarea la serverul MySQL folosind useri fara parola! (campul MySQLPassword gol)

- Campul de username din tabela SQL trebuie sa aiba valori unice! (ca masura de siguranta, e bine sa fie declarat fie primary key, fie index UNIQUE). Daca query-ul de parola returneaza mai mult de 1 inregistrare, autentificarea esueaza.