14. SERVERUL SAMBA

14.1. Concepte si terminologie Windows

14.1.1. Protocoale

In lumea Windows, principalul protocol folosit pentru partajare de resurse este SMB (Server Message Block). Acesta este un protocol client-server de nivel aplicatie care ofera servicii de autentificare, file/printer sharing si comunicare inter-proces.

De-a lungul timpului, SMB a functionat peste diverse protocoale de nivel sesiune sau transport. La origini, SMB folosea NetBIOS pentru a-si livra serviciile; ulterior Microsoft a realizat o implementare de SMB care functioneaza direct peste TCP/IP.

NetBIOS este un API – o simpla interfata pentru protocoalele de nivel aplicatie – care ofera trei servicii:

- *name service* – un serviciu care administreaza numele statiilor din retea si al serviciilor oferite de catre acestea
- *datagram service* – un serviciu de transfer de date connectionless (best effort)
- *session service* – un serviciu de transfer de date orientat pe conexiune, care ofera garantia receptiei in ordinea corecta a informatiilor transmise

Pentru a-si oferi serviciile, NetBIOS trebuie sa se foloseasca de protocoale de nivel transport. Istoric vorbind au existat diverse implementari:

- NBF (NetBIOS Frames Protocol) – NetBIOS peste IEEE 802.2
- NBX (NetBIOS over IPX/SPX) – implementarea Novell a API-ului NetBIOS, ce se foloseste de stiva de protocoale promovata la acea vreme de Novell
- NBT (NetBIOS over TCP/IP) – implementarea NetBIOS peste TCP/IP , cea mai raspandita dintre ele, cu urmatoarele particularitati:
  - serviciul de nume functioneaza pe portul 137 TCP si UDP
  - serviciul de datagrame ruleaza pe portul 138 UDP
  - serviciul de sesiuni ruleaza pe portul 139 TCP

Desi utilizat pe scara larga in lumea Microsoft, NetBIOS prezinta unele probleme/limitari care au facut ca, odata cu Windows 2000, Microsoft sa lanseze o noua versiune de SMB ce functiona direct peste TCP/IP, folosind portul TCP 445. Versiunea a fost intitulata CIFS (Common Internet Filesystem) si utiliza DNS pentru a asigura serviciile de nume oferite anterior de NetBIOS. CIFS a beneficiat si de alte adaugiri importante, cum ar fi suportul pentru symbolic/hard link-uri si dimensiuni mult mai mari de fisiere.

Odata cu Windows Vista a fost introdus SMB2, rafinat ulterior in Windows 7, iar Windows 8 a consacrat aparitia lui SMB 3.0.

14.1.2. Moduri de organizare a retelei in lumea Microsoft

Inca de dinaintea aparitiei retelelor, administrarea de sistem a ridicat problema automatizarii: spre exemplu, un angajat era necesar sa se poata loga pe mai multe statii din cadrul firmei, si deci administratorul trebuia sa-i creeze cont pe toate statiile implicate. Cu cat numarul de statii crestea, cu atat cantitatea de munca repetitiva devenea mai importanta si trebuia eficientizata.

Aparitia retelelor si deci a conectivitatii intre statii a adus si o posibilitate eleganta de rezolvare a acestei probleme: administrarea centralizata. Au fost gandite solutii care stocau conturi, privilegii etc intr-o singura locatie din retea, accesibila diferitelor statii prin protocoale specifice; o statie care dorea sa autentifice un

utilizator nu mai facea acest lucru folosindu-se de baze de date proprii, stocate local (cum este cazul clasicei combinatii passwd/shadow) ci accesa prin retea baza de date centrala. Primele implementari ale unor astfel de sisteme au fost prezente in sistemele de operare Unix, extinzandu-se apoi in sistemele de operare ulterioare.

Trebuie mentionat ca, desi explicatiile anterioare au pornit de la problema autentificarii, exista multe alte aspecte/operatii administrative care beneficiaza de pe urma centralizarii. Administrarea presupune nu doar creare de conturi si schimbare de parole, ci si securizare, gestionare de privilegii, politici de acces, solutii eficiente de backup/restaurare etc.

Desigur ca cele doua abordari (statii individuale vs administrare centralizata) exista in continuare, deoarece fiecare corespunde cate unui scenariu intalnit in viata reala:

- intr-o retea mica, unde overhead-ul administrativ este minim, se pot folosi statii individuale, administrarea de sistem efectuandu-se pe fiecare statie in parte, independent de celelalte. Intr-o astfel de situatie nu se justifica implementarea unei solutii complexe de administrare centralizata
- in momentul in care dimensiunea retelei creste, overhead-ul administrativ devine important si atunci timpul/resursele consumate in implementarea unei solutii centralizate vor fi compensate de cele scutite ulterior

Desigur, Windows nu a facut exceptie de la problematica descrisa. Din punct de vedere Microsoft exista doua moduri de organizare a retelei:

- **workgroup** – reteaua este formata din statii administrate individual. Fiecare statie are capabilitatea de a juca rolul de client sau de server dupa cum este cazul, astfel incat orice statie sa poata accesa resurse partajate de alta daca este nevoie
- **domeniu** – reprezinta raspunsul Microsoft la nevoia de administrare centralizata. Un domeniu reprezinta un grup de statii care “asculta” de unul sau mai multe servere centrale, denumite *domain controller*, care depoziteaza informatiile de autentificare si nu numai. Istoric vorbind au existat doua implementari majore:
  - *domeniile NT* – prezente in variantele Windows NT. Un astfel de domeniu beneficia de un PDC (Primary Domain Controller) si zero sau mai multe BDC (Backup Domain Controller), relatia dintre ele fiind asemanatoare cu cea dintre serverele autoritative ale unei zone DNS: PDC-ul era cel pe care se puteau aduce modificari, iar BDC-urile isi mentineau baza de date sincronizata cu cea a PDC-ului, putand astfel servi si ele cereri de autentificare, degrevand PDC-ul
  - *domeniile Active Directory* – odata cu Windows 2000 a fost gandita o noua combinatie de tehnologii pentru realizarea unui domeniu, bazata pe LDAP si Kerberos. Dispare conceptul de primary/backup domain controller, domain controllerele unui domeniu putand toate accepta modificari si existand o modalitate flexibila de a le atribui rolurile necesare in cadrul domeniulu

14.2. Prezentare Samba

14.2.1. Descriere si capabilitati

Samba reprezinta o implementare open source a protocolului SMB si a tehnologiilor/protocoalelor asociate (NBT, DCE/RPC etc). Iata rolurile pe care le poate juca Samba intr-o retea Windows:

- client pentru masini Windows standalone – implementarea de SMB din Samba permite aplicatiilor Unix/Linux sa acceseze resurse partajate pe masini Windows (fisiere, imprimante etc)
- standalone server – Samba poate fi configurat sa partajeze resurse in retea; acestea pot fi accesate de catre clientii Windows ca niste share-uri obisnuite
- domain member server – Samba partajeaza resurse pentru clientii Windows din retea, insa autentificarea este una centralizata, serverul Samba fiind parte a unui domeniu Microsoft. Roluri posibile:
  - NT4 Style Domain Server
  - Active Directory Domain Server

- domain controller – Samba reprezinta unul dintre serverele ce gestioneaza un domeniu Microsoft. Roluri posibile:
  - Primary Domain Controller (PDC) - NT4
  - Backup Domain Controller (BDC) - NT4
  - Active Directory domain dontroller (incepand cu Samba 4)

14.2.2. Fisa serverului

Executabile principale:
- **smbd** – serverul SMB
- **nmbd** – ofera servicii de nume NetBIOS

Fisier de configurare: **smb.conf**, aflat de obicei in */etc/samba*
Validarea lui smb.conf: **testparm** (cu optiunea -v sunt aratate valorile tuturor parametrilor, inclusiv cei care raman pe default!)
Pornirea serverului SMB in foreground, cu output detaliat pe ecran: **smbd -FSd 1**
Reload configurare: **smbcontrol reload-config**
Alte utilitare:
- **nmblookup** – interogare NetBIOS (*nmblookup numeNetbios* intoarce IP-ul corespunzator numelui)
- **smbclient** – client de tip FTP pentru accesarea informatiilor de pe un share SMB
- **smbcontrol** – permite comunicarea cu daemonii *smbd* si *nmbd*
- **smbpasswd** – utilitar de schimbare a parolelor utilizatorilor SMB
- **pdbedit** – management de useri virtuali samba
- **smbstatus** – listeaza conexiunile active cu serverul
- **smbtree** – listare servere SMB detectate in retea si servicii oferite de acestea
- **smbget** – utilitar asemanator cu wget pentru download de fisiere din share-uri Windows

Nota: man samba *indica toate utilitarele componente Samba impreuna cu rolul fiecaruia.*

14.3. Configurarea serverului Samba

14.3.1. Formatul fisierului de configurare

Fisierul de configurare Samba este format din directive de configurare grupate in sectiuni. O directiva este de forma *nume directiva = valoare*. Numele directivei poate cuprinde unul sau mai multe cuvinte separate prin spatii (spre deosebire de alte servere, nu se mai evita spatiul prin folosirea de – sau _). Numele de directive nu sunt case sensitive, in schimb valorile pot fi in unele situatii (spre exemplu, cazul unei valori ce constituie o cale in sistemul de fisiere).

Valorile directivelor pot fi valori simple sau liste. In cel de-al doilea caz, elementele listei pot fi separate prin spatiu, virgula sau TAB:

```
# urmatoarele directive sunt echivalente
valid users = ana dan vlad
valid users = ana,dan,vlad
valid users = ana   dan    vlad
```

Nota: *toate directivele au valori implicite, astfel incat configurarea de server presupune editarea unui minim de directive. Pentru a determina valorile default ale directivelor se poate utiliza comanda* ***testparm -v****.*

Directivele sunt incluse in *sectiuni*. O sectiune reprezinta un grup de directive ce debuteaza cu titlul sectiunii cuprins intre paranteze patrate, ca in exemplul de mai jos:

```
[sectiune1]
directiva 1
directiva 2
…
[sectiune2]
directiva3
directiva4
…
```

Fiecare sectiune se intinde de la titlul său pana la urmatoarea sectiune sau pana la finalul fisierului, oricare dintre acestea survine mai intai.

Cu foarte putine exceptii, fiecare sectiune defineste un serviciu oferit de catre serverul Samba. In cele mai dese cazuri sectiunile reprezinta share-uri (directoare partajate de Samba si accesibile clientilor Windows din retea); exista insa si sectiuni cu rol aparte. Din acest punct de vedere impartim sectiunile in doua categorii:

- speciale – reprezinta sectiuni cu rol prestabilit; nu se pot defini share-uri cu aceste nume. Samba defineste urmatoarele sectiuni speciale:
  - **[global]** – directivele cuprinse in aceasta sectiune reprezinta setari globale de server. Valorile definite in contextul global sunt mostenite automat si in sectiunile per-serviciu, unde pot fi suprascrise in caz de nevoie
  - **[printers]** – serviciu folosit pentru partajarea de imprimante in retea
  - **[homes]** – sectiune utilizata atunci cand se doreste partajarea automatizata a directoarelor personale ale utilizatorilor de sistem de pe serverul Samba
- definite de administrator – este cazul share-urilor obisnuite (vezi mai jos capitolul dedicat definirii de share-uri)

Nu orice directiva de configurare poate fi folosita in orice context; unele pot fi utilizate numai in contextul global (sectiunea [global]), celelalte pot fi utilizate si per share. Daca studiem manpage-ul lui *smb.conf* vom constata ca fiecare directiva este marcata G (de la global) sau S (de la share); directivele S pot fi folosite in general si in contextul global, unde vor juca rolul de default pentru toate share-urile definite. Atunci cand o directiva este folosita atat global cat si intr-un share, valoarea din share o suprascrie pe cea globala pentru acel share.

14.3.2. Configurare generala

Configurarea generala a serverului presupune stabilirea catorva proprietati ale sale care nu depind de lista de servicii pe care acesta le ofera:

- **workgroup** – stabileste numele workgroup-ului sau domeniului din care face parte serverul Samba
- **netbios name** – reprezinta numele NetBIOS al statiei, cel cu care ea se prezinta in retea
- **server string** – reprezinta o descriere a serverului, afisata clientilor atunci cand il acceseaza

```
workgroup = home
netbios name = sambatest
server string = server samba de test
```

14.3.3. Definirea unui share

Definirea unui share se realizeaza creand in *smb.conf* o noua sectiune cu urmatoarele particularitati:

- titlul sectiunii va reprezenta chiar numele share-ului, asa cum il vor vedea clientii

- continutul share-ului va proveni dintr-un director din sistemul de fisiere Linux; calea catre acesta se specifica cu ajutorul directivei **path**
- share-ul poate avea atasat un sir de caractere ce ii este afisat si clientului, cu ajutorul directivei **comment**
- share-ul poate fi definit ca read-only sau read-write cu ajutorul directivelor **read only** sau **writable**. Cele doua directive sunt una opusul celeilalte: *read only = no* echivaleaza cu *writable = yes*
- share-ul poate fi usor activat sau dezactivat utilizand directiva **available** (valori posibile *yes* sau *no*)
- share-ul poate fi vizibil sau nu clientilor in My Network Places sau echivalentul sau, folosind directiva **browsable** cu valoarea *yes* sau *no*

```
[poze]
path = /STUFF/pictures
comment = poze concedii
read only = yes # alternativ puteam scrie writable = no
browsable = yes
# pentru dezactivare rapida: available = no
```

14.3.4. Stabilirea rolului serverului si a modului de functionare a autentificarii

In SMB exista doua modalitati de a autentifica accesul la o resursa:

- *acces share-level* – share-ul in cauza are atasata o parola. Clientul trebuie sa dispuna de parola corecta pentru a accesa resursa in cauza. Acest mod de functionare era propriu sistemelor de operare mai vechi, standalone (Windows 95/98/ME) unde nu era necesar login pentru accesarea sistemului de operare si, in consecinta, clientul Windows nu dispunea de un username
- *acces user-level* – inainte de accesarea share-ului, clientul trebuie sa deschida o sesiune cu serverul iar in acest scop trebuie sa se autentifice folosind username si parola. Serverul autentifica clientul folosind fie baze de date proprii (cazul unui server standalone) fie baza de date centrala a unui domeniu (in cazul apartenentei la domeniu)

Exista doua directive de configurare care stabilesc – impreuna sau separat – modul in care Samba isi autentifica clientii: **security** si **server role**. Traditional, primul dintre ei era cel folosit pentru a defini modul de autentificare; in versiunile mai recente de Samba a fost introdus cel de-al doilea pentru a simplifica configurarea serverului si a o face mai intuitiva (in fond, modul de autentificare deriva din scenariul in care functioneaza serverul Samba).

Directiva **server role** poate lua una dintre urmatoarele valori:

- *AUTO* – in acest caz rolul serverului este dedus prin consultarea valorii directivei *security*
- *STANDALONE* – Samba functioneaza ca server de sine statator, autentificandu-si “personal” clientii. Aceasta este valoarea implicita atunci cand valoarea directivei *security* nu este precizata
- *MEMBER SERVER* – Samba functioneaza ca server component al unui domeniu, delegand autentificarea catre domain controllerele acelui domeniu
- *CLASSIC PRIMARY DOMAIN CONTROLLER* – Samba va juca rolul de PDC pentru un domeniu de tip NT4
- *NETBIOS BACKUP DOMAIN CONTROLLER* – Samba va juca rolul de BDC pentru un domeniu de tip NT4
- *ACTIVE DIRECTORY DOMAIN CONTROLLER* – Samba va indeplini rolul de domain controller pentru un domeniu de tip Active Directory

Directiva **security** poate lua valori care cad sub incidenta a doua categorii:

- curente, active:
  - *auto* – valoarea este dedusa din *server role*

- *user* – clientul trebuie sa se autentifice folosind username/parola. Aceasta este valoarea implicita atunci cand *server role* lipseste
- *domain* – autentificare la un PDC/BDC NT4 in calitate de member server intr-un domeniu NT4
- *ads* – autentificare la un domain controller in calitate de member server intr-un domeniu AD

• vechi, nu se mai folosesc:
- *share* – pentru implementarea accesului share-level descris mai sus
- *server* – folosit in trecut, pe vremea cand Samba nu avea capabilitatea de a face parte dintr-un domeniu nici macar ca member server si era fortat sa delege autentificarea catre unul dintre serverele domeniului

```
# server de sine statator (valoare implicita)
server role = standalone
```

14.3.5. Configurarea unui server standalone

14.3.5.1. Share-uri cu acces public

14.3.5.1.1. Principii

Un share/serviciu Samba cu acces public este unul care poate fi accesat de catre clienti indiferent de combinatia user/parola furnizata de acestia. Acest lucru se realizeaza la nivel de share cu ajutorul directivei **guest ok** (sinonima cu **public**) ce poate avea valorile *yes* sau *no*. Atunci cand un client acceseaza un asemenea share, el va face acest lucru ca *guest* – o identitate Windows speciala, creata pentru accesarea resurselor fara a prezenta user/parola corecte. Contul de guest trebuie sa aiba in spate o identitate Linux; implicit aceasta este reprezentata de contul *nobody*, dar ea poate fi modificata folosind directiva **guest account.**

Scenariul ridica insa o problema. Sa ne imaginam ca dorim sa configuram un server Samba de sine statator cu doua share-uri/servicii – unul cu acces public, altul accesibil numai clientilor autentificati corect. Vom seta *server role = standalone*, ceea ce va avea ca implicatie imediata *security=user*. Consecinta este ca orice client va trebui sa se autentifice la serverul Samba folosind username si parola, INAINTE de a preciza care este share-ul pe care doreste sa il acceseze. Inseamna ca Samba nu are idee, in momentul autentificarii clientului, daca este cazul sa-l autentifice drastic (obligandu-l sa prezinte o combinatie user/parola corecta, conform share-ului autentificat) sau sa-i permita orice username, in combinatie cu orice parola (cazul share-ului cu acces public). In aceste conditii, unui client care prezinta un username inexistent pe serverul Samba i s-ar interzice din start accesul la server.

Pentru a rezolva problema, Samba ii ofera administratorului directiva **map to guest** care ii permite sa mapeze login-urile esuate pe contul de guest. In acest fel, nu mai este nevoie nici macar ca username-ul prezentat de client sa existe pe serverul Samba – in urma tentativei de autentificare, clientul va capata automat acces ca *guest* si apoi va incerca sa acceseze share-ul dorit cu aceasta identitate.

14.3.5.1.2. Configurarea unui share read-only cu acces public

Pentru a defini un share public read-only este necesara/utila folosirea urmatoarelor directive:
- **guest ok** pentru a permite accesul ca guest. Valori posibile: yes sau no
- **map to guest** pentru a mapa login-urile invalide pe contul de guest. Principalele valori de interes:
  - *Never* – niciun client a carui autentificare esueaza nu este mapat pe guest
  - *Bad User* – clientii care prezinta username-uri inexistente pe serverul Samba vor fi considerati automat guest. *Pentru username-urile valide este nevoie in continuare de parola corecta!*
  - *Bad Password* – clientii care furnizeaza o parola gresita sunt considerati automat guest – fara ca clientul sa fie instiintat de acest lucru! (problematic, si deci de evitat)

- **guest account** pentru a preciza identitatea de sistem Linux corespunzatoare accesului ca guest
- **guest only** pentru a permite numai accesul ca guest la un share, excluzand eventualii alti useri autentificati corect. Valori posibile: yes sau no.
- **read only** (antonimul lui **writable**, care la randul sau este sinonim cu **writeable**) – pentru a interzice modificarea informatiei din share

```
[myhouse]
guest ok = yes
guest account = ftp
map to guest = Bad User
guest only = yes
read only = yes
```

14.3.5.1.3. Share-uri read-write. Control ownership & permisiuni

Am putea fi tentati sa credem ca a defini un share read-write se reduce la a inlocui *read only = yes* cu *read only = no* sau cu *writable = yes*. Insa odata ce clientul poate crea/sterge/modifica informatie din share, apar aspecte noi ce trebuie luate in considerare:

- cu ce identitate Linux sunt accesate fisierele din directorul share-ului? De aici rezulta atat permisiunile pe care clientul le va avea asupra fisierelor existente, cat si ownership-ul fisierelor nou-create
- ce permisiuni vor avea fisierele create de catre client?

Identitatea de sistem Linux cu care sunt accesate resursele este stabilita astfel:

- in cazul unei autentificari reusite – asadar clientul a furnizat o combinatie user/parola corecta – implicit username-ul Linux este chiar cel furnizat la autentificare (se va vedea ca fiecare cont Samba are in spate un user de sistem Linux)
- daca username-ul furnizat nu exista in baza de date Samba si a fost utilizata corect directiva *map to guest*, fisierele share-ului vor fi accesate cu contul Linux corespunzator accesului ca guest (stabilit cu directiva *guest account*)
- in oricare dintre cazurile de mai sus, este posibila schimbarea userului/grupului de sistem Linux cu care se acceseaza resursele din share, folosind directivele **force user** si **force group**, care primesc ca valoare username-ul, respectiv grupul cu care vor fi accesate fisierele share-ului. Asta inseamna ca, indiferent de username-ul cu care se logheaza clientii pe serverul Samba, accesul la fisierele acelui share se realizeza cu UID/GID stabilite de directivele *force**

Nota: *directivele* force user *si* force group *actioneaza DUPA autentificare, deci este necesar ca clientul sa se autentifice corect in prealabil!*

In privinta permisiunilor implicite la crearea de fisiere, Samba defineste doua directive ce limiteaza setul de permisiuni ale unui fisier sau director nou creat: **create mask** si **directory mask**. Acestea actioneaza ca o masca de biti aplicata permisiunilor implicite – in sensul in care bitii pusi pe 0 in masca vor determina absenta permisiunii de pe pozitia corespunzatoare. *Atentie! Acest mod de manifestare este diferit de cel al umask-ului din linia de comanda, care utiliza biti 1 pentru permisiunile ce se doreau eliminate!*

Fie sectiunea de configurare de mai jos:

```
[writehere]
force user = user1
force grup = www-data
create mask = 0640
directory mask = 0770
```

Ca urmare, fisierelor nou-create le vor lipsi integral permisiunile pentru other si dreptul de write pentru grup, iar directoarelor le vor lipsi permisiunile pentru other. Fisierele si directoarele nou-create vor avea ownerul *user1* si groupul *www-data* indiferent de username-ul cu care s-a autentificat clientul.

14.3.5.2. Accesul autentificat

14.3.5.2.1. Concepte

Asa cum in Linux/Unix un cont de utilizator este caracterizat printr-un UID iar un grup printr-un GID, in lumea Windows fiecare user/grup/domeniu/workgroup/etc are un SID (Security ID). SID-urile de user sunt de forma urmatoare:

S-1-5-**21-7375663-6890924511-1272660413**-*2944159*

SID-ul este compus din diverse elemente, dintre care mai sus au fost evidentiate doua:

- SID-ul statiei/domeniului in care este definit userul (cel reprezentat in bold)
- RID-ul (Relative Identifier) – identificatorul userului (cel reprezentat italic). Este echivalent ca rol cu UID-ul din Linux

Informatiile legate de conturi – si implicit SID-urile – sunt stocate intr-o baza de date denumita SAM (Security Account Manager).

Pentru a interactiona cu clientii Windows, Samba mentine propria baza de date cu useri. Argumentele ce impun existenta acesteia (in loc sa fie folosite bazele de date de sistem, passwd si shadow) sunt urmatoarele:

- orice operatie in sistemul de operare Linux are nevoie de un UID si deci trebuie stabilita intr-un fel corespondenta SID-UID, lucru care nu se poate memora in passwd, caci acesta are format impus
- setul de proprietati ale unui user Windows difera de cel al unui user Linux
- Windows foloseste alti algoritmi de hashing decat Linux; in momentul in care, sa spunem, un client Windows ar trimite hash-ul MD4 al parolei catre serverul Samba, acesta nu ar putea sa-l valideze deoarece in Linux este stocat hash-ul MD5 sau SHA al parolei, iar algoritmul de hashing este ireversibil – Samba nu poate obtine, pe baza hash-ului, parola originala pentru a-i calcula hash-ul MD4 si a-l confrunta cu cel primit de la client

Baza de date SAM mentinuta de Samba nu este una cu useri virtuali in adevaratul sens al cuvantului; **fiecarui user din SAM ii corespunde unul de sistem Linux!** In acest fel se stabileste (indirect) corespondenta SID-UID.

14.3.5.2.2. Back-end-uri de autentificare Samba

Exista diverse solutii de stocare a bazei de date SAM mentinuta de Samba, depinzand de rolul serverului si de numarul de conturi de utilizator implicate. Avem urmatoarele posibilitati:

- back-end-uri curente, in uz:
  - **tdbsam** – TDB (trivial database) reprezinta un format de baza de date dezvoltat de echipa Samba. Informatiile sunt stocate in fisiere cu extensia *.tdb*. Ce are special formatul TDB fata de alte solutii de stocare de informatie in fisiere binare este ca suporta modificare concurenta, ceea ce elimina o potentiala gatuitura cand mai multe procese acceseaza simultan aceeasi baza de date. Back-end-ul *tdbsam* este recomandat pentru workgroup-uri sau pentru scenarii de maxim 250 de useri (desi in functie de caz poate functiona si cu mai multi)
  - **ldapsam** – solutia scalabila, redundanta, recomandabila pentru scenarii enterprise
- back-end-uri vechi, care nu se mai folosesc:

- **smbpasswd** – reprezinta o baza de date stocata intr-un fisier text, asemanatoare ca format cu passwd. *Atentie! A nu se confunda back-end-ul/fisierul smbpasswd, cu utilitarul smbpasswd folosit pentru schimbarea parolei utilizatorilor!*
- **plaintext** – o solutie veche ce stoca in clar informatiile de autentificare

14.3.5.2.3. Managementul userilor Samba

Samba pune la dispozitia administratorului cateva utilitare pentru administrarea informatiilor de autentificare din baza sa de date SAM:

- **smbpasswd** – este utilitarul clasic, folosit de pe vremea cand baza de date SAM era memorata in fisierul smbpasswd. El permite adaugare/stergere/activare/dezactivare conturi de useri/masini si schimbare de parole. Daca la origini el modifica direct fisierul smbpasswd, in ziua de astazi utilitarul se conecteaza la daemonul *smbd* si ii solicita acestuia operatiile dorite, ceea ce il decupleaza de backend-ul concret folosit. Se intentioneaza ca, pe viitor, functionalitatea acestui utilitar sa fie migrata in comanda *net*
- **pdbedit** – utilitar nou care, in plus fata de *smbpasswd*, permite editarea politicilor de login ale userului (numar de incercari de login esuate, politica de schimbare periodica a parolei etc)
- **net** – utilitar general de administrare de server Samba, in diversele aspecte pe care aceasta le presupune. Sintaxa sa de utilizare este gandita sa se suprapuna peste cea a utilitarului omonim din Windows.

Adaugarea unui nou utilizator in SAM presupune doua operatii:

- crearea unui cont de sistem Linux, cu acelasi username; in lipsa acestuia, adaugarea in SAM va esua!
- adaugarea userului in SAM folosind unul dintre utilitare:

```
# adaugare in sistem
root@samba# useradd test
# adaugare in SAM – varianta pdbedit
root@samba# pdbedit -a -u test
# alternativa – varianta smbpasswd
root@samba# smbpasswd -a test
```

Iata modul de realizare a altor cateva operatii ce implica conturi de utilizator:

- stergerea unui user

```
# varianta pdbedit
root@samba# pdbedit -x -u test
#varianta smbpasswd
root@samba# smbpasswd -x test
```

- schimbarea parolei unui utilizator existent:

```
# varianta pdbedit
root@samba# pdbedit -a -u test
# varianta smbpasswd
root@samba# smbpasswd test
```

- listarea informatiilor unui user existent in baza de date:

```
# varianta pdbedit
root@samba# pdbedit -Lvu test
Unix username:        test
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-2446688148-2151841699-259383257-1005
```

```
Primary Group SID:    S-1-5-21-2446688148-2151841699-259383257-513
Full Name:            ionut
Home Directory:       \\black\ionut
HomeDir Drive:
Logon Script:
Profile Path:         \\black\ionut\profile
Domain:               BLACK
Account desc:
Workstations:
Munged dial:
Logon time:           0
Logoff time:          Mi, 06 feb 2036 17:06:39 EET
Kickoff time:         Mi, 06 feb 2036 17:06:39 EET
Password last set:    Jo, 25 sep 2014 12:01:00 EEST
Password can change:  Jo, 25 sep 2014 12:01:00 EEST
Password must change: never
Last bad password   : 0
Bad password count  : 0
Logon hours         : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

14.3.5.2.4. Alias-uri de useri

Atunci cand un client se autentifica la serverul Samba, maparea userului Windows pe unul de sistem Linux poate fi realizata in urmatoarele moduri:

- direct: implicit va fi utilizat contul de sistem Linux atasat userului Windows in baza de date Samba
- indirect: Samba dispune de directiva **username map** cu ajutorul careia se pot defini manual corespondente user Windows-user Linux. Directiva primeste ca valoare calea catre un fisier text in care fiecare linie este de forma *user_unix = user_windows_1 user_windows_2 …* Iata un exemplu:

```
# in fisierul smb.conf
[global]
username map = /etc/samba/useri.map

# in fisierul useri.map
root = administrator manager
```

Atentie! In cazul unui server standalone (care nu este parte a unui domeniu, si deci isi autentifica singur clientii), substitutia de username este efectuata inaintea autentificarii! Asadar configurarea de mai sus va avea doua efecte notabile:

- userii *administrator* si *manager* nu trebuie sa existe in SAM sau in passwd
- la autentificare, cei doi useri vor trebui sa foloseasca parola userului root (este vorba de userul root din SAM, nu de cel de sistem!)

14.3.6. Controlul accesului la resurse/servicii

Samba ofera un set bogat de directive ce-i permit administratorului sa controleze accesul la resursele oferite de server. Iata cele mai importante categorii:

- restrictionarea setului de interfete pe care asculta serverul: directivele **interfaces** si **bind interfaces only**. Interfetele permise pot fi specificate sub forma de nume (cu sau fara wildcard-uri) sau de adresa, insotita eventual de netmask:

```
[global]
interfaces = eth*  lo  10.0.0.100/24  192.168.0.100/255.255.255.0
bind interfaces only = yes
```

- restrictionarea/acordarea per user/grup a accesului la un serviciu/share: directivele **valid users**, **invalid users** si **admin users**. Daca un user se regaseste in ambele liste (useri valizi si invalizi) accesul ii va fi interzis. Userii din lista de *admin users* vor efectua toate operatiile din share cu drepturi de root!

```
[share1]
valid users = @managers, user1 # cele prefixate cu @ sunt grupuri
invalid users = user2, @limited
admin users = administrator # userii din aceasta lista au drepturi depline asupra resurselor
```

- gestionarea dreptului de scriere intr-un share: directivele **read only**/**writable** (sinonime), **read list** si **write list**. Directiva *read only* activeaza sau dezactiveaza dreptul de scriere la nivel de intreg share; userii din *read list* vor avea numai permisiune de citire, indiferent de valoarea directivei read only; userii din *write list* vor avea permisiune de write indiferent de valoarea directivei read only

```
[share1]
read only = yes
write list = user1 @managers
```

Nota: *directiva* write list *nu poate acorda permisiuni de scriere decat daca permisiunile din sistemul de fisiere Linux accepta modificarea acelor fisiere! Write list nu poate contrazice permisiunile Linux.*

- controlul accesului la serverul Samba in functie de adresa IP a clientului: directivele **hosts allow** si **hosts deny**. Pot fi specificate global sau per-share, efectul fiind si el global sau per-share, dupa caz. Allow are prioritate fata de deny; in cazul in care adresa unui client se regaseste in ambele liste, i se va permite accesul

```
hosts allow = 127.0.0.1 192.168.2.0/24 192.168.3.0/24
hosts deny = 0.0.0.0/0 # in loc de 0.0.0.0/0 se poate scrie ALL
```

Nota: *sa nu uitam ca utilitarul smbpasswd comunica cu daemonul smbd chiar si la schimbarea parolei unui user local, si intr-un astfel de caz el are nevoie sa acceseze serverul Samba de pe localhost! Accesul de pe localhost este implicit permis, insa trebuie sa avem grija sa nu-l interzicem explicit in hosts deny.*

14.4. Diagnostic

14.4.1. Rularea serverului in foreground, cu loguri detaliate in terminal

Executabilul *smbd* dispune de urmatoarele optiuni utile:

- **-F** – solicita ruarea in foreground
- **-S** – determina afisarea logurilor in terminal
- **-d** *nivel* – stabileste nivelul de debug si deci gradul de detaliu al informatiilor afisate pe ecran. Nicelul de debug se incadreaza intre 1 (nimin) si 10 (maxim)

```
root@samba# smbd -FSd4
smbd version 4.1.6-Ubuntu started.
Copyright Andrew Tridgell and the Samba Team 1992-2013
uid=0 gid=0 euid=0 egid=0
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Registered MSG_REQ_POOL_USAGE
Registered MSG_REQ_DMALLOC_MARK and LOG_CHANGED
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Processing section "[share1]"
Processing section "[share2]"
added interface eth0 ip=192.168.1.100 bcast=192.168.1.255 netmask=255.255.255.0
```

```
standard input is not a socket, assuming -D option
Failed to fetch record!
waiting for connections
```

14.4.2. Utilitare client Linux

Amintim cateva utilitare Samba ce ajuta in verificarea corectei functionalitati a serverului si in diagnosticarea sa in caz de nevoie:

- **smbstatus** – arata conexiunile active pe serverul Samba

```
root@samba# smbstatus

Samba version 4.1.6-Ubuntu
PID     Username      Group         Machine
-------------------------------------------------------------------
13410    user2        admins         192.168.1.101 (ipv4:192.168.1.101:1077)

Service      pid     machine       Connected at
-------------------------------------------------------
IPC$         13410   192.168.1.101  Thu Sep 25 13:10:10 2014

No locked files
```

- **smbclient** – un utilitar de tip client cu utilizare foarte asemanatoare cu *ftp*. Cu ajutorul sau administratorul se poate conecta la un server SMB si poate initia transferuri de fisiere folosind comenzile clasice ale utilitarului ftp (*cd, lcd, ls, get, mget, put, mput, quit*)
- **net** – permite accesul la un set impresionant de operatii. Comanda actioneaza prin conectarea la un server SMB, asadar este necesara specificarea adresei serverului (daca difera de localhost) folosind optiunea -S, si a unui user care are permisiuni administrative folosind optiunea -U. Iata cateva exemple de operatii utile:

```
# listare share-uri de pe serverul specificat
root@samba# net rpc share -S 10.0.0.100 -Uadministrator
Enter administrator's password:
share1
share2
print$
IPC$
HP-Photosmart-3100

# listare useri definiti in SAM (cu parametri suplimentari se pot sterge/adauga/redenumi useri)
root@samba# net rpc user -S 10.0.0.100 -Uadministrator
Enter administrator's password:
root
user1
user2

# listare imprimante partajate pe serverul sepcificat
root@samba# net rpc printer -S 10.0.0.100 -Uadministrator
Enter root's password:
listing printers
printer 1: HP-Photosmart-3100, shared as: HP-Photosmart-3100
```

14.4.3. Utilitare client Windows

Din Windows, un share definit pe alt server (fie el Windows sau Samba) poate fi accesat din Windows Explorer, folosind in bara de adresa *numeserver* sau *IPserver*. Trebuie tinut cont insa de doua aspecte:

- Windows va incerca implicit sa se autentifice cu userul curent logat, fara a incerca intai sa afiseze fereastra de introducere de username/parola
- Windows cache-uieste (retine temporar si refoloseste) informatiile de login deja introduse, ceea ce complica re-conectarea la un acelasi share cu un user diferit

Atunci cand se testeaza un server Samba si se verifica diferite share-uri sau conturi de utilizator de pe aceeasi masina client Windows, avem la dispozitie doua solutii pentru a nu fi incomodati de mecanismul de caching Windows:

- putem lucra efectiv cu useri diferiti – fie delogand un user si logandu-l pe celalalt, fie folosind facilitatea de comutare de user prezenta in Windows
- putem restarta serviciul Workstation de pe masina client atunci cand dorim inchiderea conexiunilor active cu alte servere
- putem folosi utilitarul *net* (cel de Windows!) in command prompt pentru a sterge conexiunile active ale clientului cu serverul dorit sau cu toate serverele la care acesta este conectat:

```
# vizualizare conexiuni active
C:\Documents and Settings\Administrator> net use
New connections will be remembered.

Status       Local     Remote                    Network
-------------------------------------------------------------------------------
OK                     \\black\IPC$              Microsoft Windows Network
The command completed successfully.

# inchiderea sesiunilor active
C:\Documents and Settings\Administrator> net use * /del
You have these remote connections:
                   \\black\IPC$
Continuing will cancel the connections.

Do you want to continue this operation? (Y/N) [N]: y
The command completed successfully.
```

14.5. BIBLIOGRAFIE

- Samba HOWTO collection: http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/
- Samba 3 by example: http://www.samba.org/samba/docs/man/Samba-Guide/
- Samba wiki: https://wiki.samba.org/index.php/Main_Page
- Directive smb.conf: http://www.samba.org/samba/docs/man/manpages/smb.conf.5.html
- Comanda net: http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/NetCommand.html
- Ubuntu Server Guide - Samba: https://help.ubuntu.com/14.04/serverguide/samba.html
- Ubuntu Wiki - Samba Community page: https://help.ubuntu.com/community/Samba