9

Hibakeresés a Sambában

A Samba rendkívül hibatűrő programcsomag. Miután mindent beállítottunk a kívánságunk szerint, talán meg is feledkezünk arról, hogy egyáltalán fut. Hibák többnyire csak a telepítés során vagy akkor fordulnak elő, amikor valamilyen új összetevővel egészítjük ki a kiszolgálót. Szerencsére egész sor eszköz áll a rendelkezésünkre az ilyen problémák diagnosztizálására. Arra semmiképpen sem vállalkozhatunk, hogy minden lehetséges hibára részletes megoldást adjunk, de az e fejezetben leírtak elegendő támpontot adhatnak a megfelelő kiinduláshoz.

A fejezet első részében a hibakereséshez rendelkezésünkre álló eszközökkel, a második részében a részletes tudnivalókkal ismerkedünk meg. Végül a harmadik rész azokról az erőforrásokról szól, amelyekre speciális problémák megoldásához lehet szükségünk.

Az eszközkészlet

A Unixra bizonyos szempontból úgy is tekinthetünk, mint egy sor alkalmazás és eszköz gyűjteményére. Az utóbbiak egy részét kifejezetten hibakereséshez használhatjuk. Mint sok más esetben, itt is igaz, hogy ugyanazt a feladatot többféleképpen is megoldhatjuk. Amikor a Sambával kapcsolatos hibákat keressük, először a következőkből induljunk ki:

1. Samba naplófájlok
2. Hibafa
3. Unix segédprogramok
4. Samba tesztprogramok
5. Dokumentáció és a FAQ (gyakran ismételt kérdések)
6. Kutatható archívumok
7. Samba hírcsoportok

Lássuk külön-külön, miként használhatjuk ezeket.

Samba naplófájlok

Hiba esetén elsőként mindig a naplófájlokat nézzük át. A Samba naplófájljai jelentősen segítik az olyan problémák diagnosztizálását, amellyel a kezdő és a középhaladó Samba rendszergazdák találkozhatnak. A naplóvezetést illetően a Samba nagyon rugalmas. A kiszolgáló konfigurálásától függően a naplófájlok nagyon kevés és nagyos sok információt is megörökíthetnek. Változókat használva gépenként, megosztásokként vagy ezek kombinációjaként szét is választhatjuk a naplófájlokat.

Alapbeállítás szerint a Samba naplóbejegyzéseit a *samba_könyvtár/var/smbd.log* és a *samba_könyvtár/var/nmbd.log* fájlok tartalmazzák, ahol a *samba_könyvtár* azt a könyvtárat jelenti, ahová a Sambát telepítettük (ez rendszerint a */usr/local/samba*). Amint a 4. fejezetben említettük, az *smb.conf* konfigurációs fájl `log file` beállításában a fájlok helyét és nevét is megváltoztathatjuk. A beállításban a 2. fejezetben ismertetett bármelyik változót használhatjuk, így külön-külön naplófájlokat hozhatunk létre minden egyes kapcsolódó felhasználó számára, ha az *smb.conf* fájl `[global]` szakaszába felvesszük az alábbi sort:

```
log file = %m.log
```

Másik megoldásként a parancssorban az `-l` kapcsoló megadásával is specifikálhatjuk a naplófájl könyvtárát, például a következő módon:

```
smbd -l /usr/local/var/samba
```

Azt is megtehetjük, hogy megosztásonként választjuk szét a naplófájlokat; ennek különösen akkor vehetjük hasznát, ha a hiba okozójaként egy adott megosztásra gyanakszunk. A konfigurációs fájl `[global]` szakaszába ekkor az `%S` változót használva a következő bejegyzést vegyük fel:

```
log file = %S.log
```

Naplózási szintek

A Samba naplózási szintjeit az *smb.conf* fájl globális hatókörű `log level` vagy az ezzel egyenértékű `debug level` beállításában írhatjuk elő. A naplózási szint értéke egy 0 (nincs naplózás) és 10 (legrészletesebb napló) közötti értéket vehet fel. Az esetek többségében a 3-as szint már kielégítő adatokat szolgáltat. Tegyük fel például, hogy egy Windows ügyfél tallóz egy könyvtárat a Samba kiszolgálón. Ha csak kevés adatot akarunk összegyűjteni, akkor a `log level = 1` beállítást használhatjuk, ami arra utasítja a Sambát, hogy csak felületes információkat naplózzon. Az alábbi bejegyzés például csak a kapcsolat létrejöttéről ad tájékoztatást:

```
105/25/98 22:02:11 server (192.168.220.100) connect to service public
as user pcguest (uid=503,gid=100) (pid 3377)
```

A magasabb naplózási szintek részletesebb információkkal szolgálnak. A 3-as szint által szolgáltatott információk általában elegendőek, és még a Samba rendszergazdáknak sincs többre szükségük. A magasabb szinteknek megfelelő információkat többnyire csak a fejlesztők használják, akik értik is ezeknek a hosszasan sorjázó rejtélyes üzeneteknek a jelentését.

Az alábbiak arra mutatnak példát, hogy ugyanazon művelet elvégzésekor mi kerül be a naplófájlba a 2-es, és mi a 3-as naplózási szint választásakor. Ne aggódjunk amiatt, ha netán nem értenénk egy SMB kapcsolat minden részletét, a célunk mindössze az, hogy bemutassuk, milyen típusú információk kerülnek naplózásra különböző szintek használatakor:

```
/* Level 2 */
Got SIGHUP
Processing section "[homes]"
Processing section "[public]"
Processing section "[temp]"
Allowed connection from 192.168.220.100 (192.168.220.100) to IPC$
Allowed connection from 192.168.220.100 (192.168.220.100) to IPC/

/* Level 3 */
05/25/98 22:15:09 Transaction 63 of length 67
switch message SMBtconX (pid 3377)
Allowed connection from 192.168.220.100 (192.168.220.100) to IPC$
ACCEPTED: guest account and guest ok
found free connection number 105
Connect path is /tmp
chdir to /tmp
chdir to /
05/25/98 22:15:09 server (192.168.220.100) connect to service IPC$ as
user pcguest (uid=503,gid=100) (pid 3377)
05/25/98 22:15:09 tconX service=ipc$ user=pcguest cnum=105
05/25/98 22:15:09 Transaction 64 of length 99
switch message SMBtrans (pid 3377)
chdir to /tmp
trans <\PIPE\LANMAN> data=0 params=19 setup=0
Got API command 0 of form <WrLeh> <B13BWz>
(tdscnt=0,tpscnt=19,mdrcnt=4096,mprcnt=8)
Doing RNetShareEnum
RNetShareEnum gave 4 entries of 4 (1 4096 126 4096)
05/25/98 22:15:11 Transaction 65 of length 99
switch message SMBtrans (pid 3377)
chdir to /
chdir to /tmp
trans <\PIPE\LANMAN> data=0 params=19 setup=0
Got API command 0 of form <WrLeh> <B13BWz>
(tdscnt=0,tpscnt=19,mdrcnt=4096,mprcnt=8)
Doing RNetShareEnum
RNetShareEnum gave 4 entries of 4 (1 4096 126 4096)
05/25/98 22:15:11 Transaction 66 of length 95
switch message SMBtrans2 (pid 3377)
chdir to /
chdir to /pcdisk/public
call_trans2findfirst: dirtype = 0, maxentries = 6,
close_after_first=0, close_if_end = 0 requires_resume_key = 0 level =
260, max_data_bytes = 2432
unix_clean_name [./DESKTOP.INI]
unix_clean_name [desktop.ini]
```

```
unix_clean_name [./]
creating new dirptr 1 for path ./, expect_close = 1
05/25/98 22:15:11 Transaction 67 of length 53
switch message SMBgetatr (pid 3377)
chdir to /
[...]
```

Az első csomag beérkezését követően elvágtuk a listát, mert a további része még több oldalt is elfoglalna a könyvben. Arra azonban már ennyiből is következtethetünk, hogy a 3-as szint fölötti szinteket beállítva megabájtok sokaságával zsúfolhatunk tele értékes lemezterületeket. A 3-as naplózási szint információi alapján már pontosan nyomon követhetjük, hogy mi történik a kiszolgálón, és az esetek többségében ezekből egyértelműen kideríthető a hiba oka.

Még egy figyelmeztetés: a magasabb naplózási szintek (3-as vagy e felettiek) jelentős mértékben lelassíthatják a kiszolgáló működését. Ne feledjük, hogy minden egyes naplóbejegyzés egy írási műveletet jelent (ami már önmagában is lassú művelet), és a 2-esnél magasabb szintek rengeteg bejegyzést generálnak. Ezért azt javasoljuk, hogy a 3-as szintet is csak akkor használjuk, ha ténylegesen ki akarjuk deríteni egy hiba okát.

A naplózás be- és kikapcsolása

A naplózás be- és kikapcsolásához adjuk meg a megfelelő naplózási szintet az *smb.conf* fájl `[global]` szakaszában. Ezt követően vagy újraindítjuk a Sambát, vagy arra kényszerítjük az aktuális démont, hogy dolgozza fel újra a konfigurációs fájl. Azt is megtehetjük, hogy az *smbd* processzt a futása közben utasítjuk a naplózási szint módosítására: a SIGUSR1 jel kiküldésével eggyel növelhetjük, a SIGUSR2 jel kiküldésével pedig eggyel csökkenthetjük a naplózás szintjét.

```
# Naplózási szint növelése 1 szinttel
kill -SIGUSR1 1234

# Naplózási szint csökkentése 1 szinttel
kill -SIGUSR2 1234
```

Naplózás ügyfélgépenként vagy felhasználónként

A hibakeresés egyik hatékony – és más felhasználókat nem zavaró – módja az, hogy az *smb.conf* fájl `[global]` szakaszában különböző naplózási szinteket rendelünk a különböző gépekhez. Ezt a korábban már bemutatott stratégia alapján tehetjük meg:

```
[global]
    log level = 0
    log file = /usr/local/samba/lib/log.%m
    include = /usr/local/samba/lib/smb.conf.%m
```

Ezekkel a beállításokkal arra utasítjuk a Sambát, hogy minden egyes kapcsolódó ügyfélhez egyedi konfigurációt és naplófájlokat használjon. Ekkor csak az a teendőnk, hogy elkészítünk egy adott ügyfél gépéhez egy olyan *smb.conf* fájlt, amely a `log level = 3`

beállítást tartalmazza (a többi ügyfélre az alapértelmezés szerinti naplózási szint [0] lesz érvényes), és az így létrejövő naplófájlt tanulmányozva keressük a hiba okát.

Az is előfordulhat, hogy csak egy adott felhasználóval kapcsolatos hibára gyanakszunk, és ez a hiba gépről-gépre követi a felhasználót. Ekkor megtehetjük, hogy csak az ehhez a felhasználóhoz kapcsolódó naplófájlokat készítjük el, ha az *smb.conf* fájlba felvesszük az alábbi sorokat:

```
[global]
    log level = 0
    log file = /usr/local/samba/lib/log.%u
    include = /usr/local/samba/lib/smb.conf.%u
```

Ezt követően készítsünk az illető felhasználó számára egy egyedi *smb.conf* fájlt (ami például a */usr/local/samba/lib/smb.conf.tim* lehet), és vegyük fel bele a `log level = 3` beállítást. Ekkor csak az ehhez a felhasználóhoz kapcsolódó részletes naplóbejegyzéseket kell tanulmányoznunk.

A Samba tesztprogramjai

A Samba tesztprogramjait, amelyek segítségével a kiszolgáló jelentős része vizsgálható, a disztribúciós csomag */docs/textdocs* könyvtárában található különböző fájlok írják le, kezdve a *DIAGNOSIS.TXT* fájllal. Az ebben a fejezetben bemutatásra kerülő hibafa a Samba fejlesztőcsoportja által javasolt alaptesztek valamivel részletesebb változata, de csak a telepítéssel és az újrakonfigurálással kapcsolatos diagnosztikai műveleteket tartalmazza, akárcsak a *DIAGNOSIS.TXT* fájl. A */docs* könyvtárban és az alkönyvtáraiban lévő többi fájl speciális problémákkal foglalkozik (például Windows NT ügyfelekkel), és olyan hibákkal kapcsolatban adnak útmutatásokat, amelyekre ebben a könyvben nem térünk ki. Ha a hibafa nem lenne elegendő, akkor olvassuk el a *DIAGNOSIS.TXT* és a többi ehhez hasonló fájl szövegét.

Unix segédprogramok

Esetenként az is eredményre vezethet, ha nem a Samba csomag valamelyik eszközével vizsgáljuk, mi is történik a kiszolgáló belsejében. A Unixban két olyan diagnosztikai eszköz is van, amelyekkel hibákat kereshetünk a Samba kiszolgálóban: a *trace* és a *tcpdump*.

A trace eszköz

A *trace* eszköz a különböző operációs rendszerekben különböző neveken bukkan fel. A Linuxban *strace*, a Solarisban *truss*, míg az SGI rendszerben *padc* és *par* a neve. Lényegében mindegyiküknek azonos a feladata: bemutatja, hogyan zajlik le az operációs rendszerben egy függvényhívás. Ezzel nyomon követhetjük egy program, így például a Samba kiszolgáló végrehajtásának menetét, és gyakran már ebből is kiszúrhatjuk azt a függvényhívást, amelyik a problémát okozza.

A *trace* eszköz segítségével az olyan hibák okát is kideríthetjük, amelyek abból származnak, hogy rossz helyen vannak a dinamikusan csatolt függvénytárak. Ilyen hibák akkor fordulhatnak elő, ha a Samba előre lefordított összetevőit töltjük le valahonnan.

A hibát okozó függvényhívás általában a kimenet végén látható, közvetlenül a program befejeződése előtt.

Az alábbiakban a Linux operációs rendszeren végrehajtott `strace` program kimenete látható. Ez a kimenet egy nagyobb fájlnak a részlete, amely a Samba kiszolgáló egyik könyvtárának megnyitásakor generálódott. A fájl mindegyik sora egy-egy rendszerfüggvény neve, ami után a paraméterek és a visszatérési érték áll. Ha hiba keletkezett egy függvény meghívásakor, akkor a sor végén a hibaérték (például `ENOENT`) és egy magyarázat olvasható. A paraméterek típusáról és a hibaértékek jelentéséről az illető operációs rendszer megfelelő *trace* programjának kézikönyvében olvashatunk.

```
chdir("/pcdisk/public")                 = 0
stat("mini/desktop.ini", 0xbffff7ec)    = -1 ENOENT (No such file or
directory)
stat("mini", {st_mode=S_IFDIR|0755, st_size=1024, ...}) = 0
stat("mini/desktop.ini", 0xbffff7ec)    = -1 ENOENT (No such file or
directory)
open("mini", O_RDONLY)                  = 5
fcntl(5, F_SETFD, FD_CLOEXEC)           = 0
fstat(5, {st_mode=S_IFDIR|0755, st_size=1024, ...}) = 0
lseek(5, 0, SEEK_CUR)                   = 0
SYS_141(0x5, 0xbfffdbbc, 0xedc, 0xbfffdbbc, 0x80ba708) = 196
lseek(5, 0, SEEK_CUR)                   = 1024
SYS_141(0x5, 0xbfffdbbc, 0xedc, 0xbfffdbbc, 0x80ba708) = 0
close(5)                                = 0
stat("mini/desktop.ini", 0xbffff86c)    = -1 ENOENT (No such file or
directory)
write(3, "\0\0\0#\377SMB\10\1\0\2\0\200\1\0"..., 39) = 39
SYS_142(0xff, 0xbffffc3c, 0, 0, 0xbffffc08) = 1
read(3, "\0\0\0?", 4)                   = 4
read(3, "\377SMBu\0\0\0\0\0\0\0\0\0\0\0"..., 63) = 63
time(NULL)                              = 896143871
```

A fenti példában különböző `stat` hívások sikertelenek, mert nem találják az általuk elvárt fájlokat. Nem kell szakértőnek lennünk ahhoz, hogy rájöjjünk arra, hiányzik a *desktop.ini* fájl az illető könyvtárból. Nagyon sok, látszólag bonyolult probléma okára egyszerűen rájöhetünk, ha azt látjuk, hogy ismétlődő hibákat jelez a *trace* program.

A tcpdump eszköz

A *tcpdump* program – írója Van Jacobson, Craig Leres és Steven McCanne, a kiegészítését Andrew Tridgell végezte – lehetővé teszi a hálózati forgalom valós idejű figyelemmel kísérését. A programhoz különböző kiviteli formátumok állnak rendelkezésre, amelyek segítségével úgy szűrhetjük meg a kimenetét, hogy csak egy adott típusú forgalmat figyeljünk meg. A *tcpdump* program segítségével az ügyfél és a kiszolgáló közötti teljes adatforgalmat vizsgálhatjuk, beleértve az SMB és az NMB üzenetszórásos (broadcast) üzeneteket is. A program a hibákat elsősorban az OSI modell hálózati szintjén keresi, ennek ellenére általános képet kaphatunk a kiszolgáló és az ügyfél tevékenységéről.

Az alábbiakban egy példát látunk a *tcpdump* naplózott futásáról. A példában az ügyfél egy könyvtárlistát kér, amire a kiszolgáló a válaszában megadja a `homes`, a `public`, az `IPC$` és a `temp` könyvtár nevét (a jobb szélre néhány magyarázó megjegyzést írtunk be):

```
$tcpdump -v -s 255 -i eth0 port not telnet
SMB PACKET: SMBtrans (REQUEST)              Csomag kérése
SMB Command   =  0x25                       A kérés ls vagy dir volt
[000] 01 00 00 10 ....

>>> NBT Packet                              SMB csomag külső kerete
NBT Session Packet
Flags=0x0
Length=226
[kihagyott sorok]

SMB PACKET: SMBtrans (REPLY)                A kérésre adott válasz kezdete
SMB Command   =  0x25                       A parancs ls vagy dir volt
Error class   =  0x0
Error code    =  0                          Nincs hiba
Flags1        =  0x80
Flags2        =  0x1
Tree ID       =  105
Proc ID       =  6075
UID           =  100
MID           =  30337
Word Count    =  10
TotParamCnt=8
TotDataCnt=163
Res1=0
ParamCnt=8
ParamOff=55
Res2=0
DataCnt=163
DataOff=63
Res3=0
Lsetup=0
Param Data: (8 bytes)
[000] 00 00 00 00 05 00 05 00              ........

Data Data: (135 bytes)                    Az aktuális könyvtár tartalma:
[000] 68 6F 6D 65 73 00 00 00  00 00 00 00 00 00 00 00  homes... ........
[010] 64 00 00 00 70 75 62 6C  69 63 00 00 00 00 00 00  d...publ ic......
[020] 00 00 00 00 75 00 00 00  74 65 6D 70 00 00 00 00  ....u... temp....
[030] 00 00 00 00 00 00 00 00  76 00 00 00 49 50 43 24  ........ v...IPC$
[040] 00 00 00 00 00 00 00 00  00 00 03 00 77 00 00 00  ........ ....w...
[050] 64 6F 6E 68 61 6D 00 00  00 00 00 00 00 00 00 00  donham.. ........
```

```
[060] 92 00 00 00 48 6F 6D 65  20 44 69 72 65 63 74 6F  ....Home  Directo
[070] 72 69 65 73 00 00 00 49  50 43 20 53 65 72 76 69  ries...I PC Servi
[080] 63 65 20 28 53 61 6D                              ce (Sam
```

A parancsban `-v` kapcsoló jelentése *verbose* (részletes információk), az `-s 255` azt jelzi, hogy az egyes csomagok első 255 bájtját kérjük (az alapértelmezés szerinti 68 bájt helyett), az `-i eth0` kapcsoló azt közli a *tcpdump* paranccsal, hogy melyik portra figyeljen (egy Ethernet portra), végül a `port not telnet` kapcsoló letiltja a telnetes forgalom figyelését, mivel telnettel távolról jelentkezhetünk be a kiszolgálóra. A *tcpdump* programnak ezeken felül még egyéb kapcsolói is vannak, amelyek segítségével úgy szűrhetjük meg a forgalmat, hogy csak az legyen látható, amire kíváncsiak vagyunk. Aki dolgozott már *snoop* vagy *etherdump* programmal, annak ismerősök lehetnek ezek a szűrők.

A módosított *tcpdump* program a Samba FTP kiszolgálójáról, az *ftp://samba.anu.edu.au/pub/samba/tcpdump-smb* címről tölthető le. Más változatai nem támogatják az SMB protokollt – ha nem látnánk az előző példában bemutatott kimenetet, akkor a program SMB-t támogató verzióját kell használnunk.

A hibafa

A hibafa a Samba telepítésekor és konfigurálásakor előforduló hibák diagnosztizálására és kijavítására használható eljárás, amely a Samba disztribúció hibakereső és diagnosztizáló dokumentációjának kibővített változata.

Mielőtt még a Samba szoftvercsomag bármelyik részét vizsgálhatnánk, ismernünk kell az alábbi adatokat:

- az ügyfél IP címe (példánkban a 192.168.220.105 címet használjuk);
- a kiszolgáló IP címe (példánkban a 192.168.220.100 címet használjuk);
- a hálózat alhálózati maszkja (ez tipikusan a 255.255.255.0 cím);
- az összes gép ugyanabban az alhálózatban van-e (a példánkban igen).

Az egyszerűbb érthetőség érdekében a példánkban a kiszolgálónak *server.example.com*, az ügyfélgépnek pedig *client.example.com* lesz a neve.

A hibafa használata

A hibakeresést az itt leírt sorrendben végezzük, anélkül, hogy előre ugranánk a könyv későbbi részeire. Ha sikeresen befejeződik egy teszt, megadjuk annak a fejezetrésznek a címét, ahová már nyugodtan továbbléphetünk.

Hibakeresés az IP alacsony szintjén

Az első tesztsorozatban a Samba futtatásához szükséges alacsony szintű szolgáltatásokat vizsgáljuk. E tesztek futtatásával meggyőződhetünk arról, hogy

- működik az IP-szoftver;
- működik az Ethernet kártya;
- működik az alapvető névszolgáltatás.

A további tesztekkel a TCP-szoftvert, a Samba *smdb* és *nmbd* démonjait, a host alapú hozzáférésvezérlést, a hitelesítést, a felhasználói hozzáférésvezérlést, a fájlszolgáltatásokat és a tallózást fogjuk vizsgálni. Ezeket a teszteket nagyon részletesen írjuk le, hogy mind a műszaki beállítottságú, mind a tapasztalt hálózati és rendszergazdák számára érthetők legyenek.

Hálózati szoftver tesztelése a ping paranccsal

Elsőként mind a kiszolgálónál, mind az ügyfélnél a `ping 127.0.0.1` parancsot kell kiadnunk. A megadott cím az ún. *visszahurkolási* (loopback) cím, és a parancs azt teszteli, hogy egyáltalán támogatja-e a gép a hálózati munkát. Unix rendszerben a parancs alakja `ping 127.0.0.1` a statisztikai beállítással, és néhány sor után megszakíthatjuk a futását. Sun munkaállomásokon a parancs alakja általában `/usr/etc/ping -s 127.0.0.1`, míg Linux rendszerben egyszerűen csak `ping 127.0.0.1`. Windows ügyfeleknél a DOS parancssorba a `ping 127.0.0.1` parancsot kell beírnunk, és a program négy sor kiírása után magától leáll.

Az alábbiakban egy Linux kiszolgálón mutatjuk be a parancs kimenetét:

```
server% ping 127.0.0.1
PING localhost: 56 data bytes 64 bytes from localhost (127.0.0.1):
icmp-seq=0. time=1. ms 64 bytes from localhost (127.0.0.1):
icmp-seq=1. time=0. ms 64 bytes from localhost (127.0.0.1):
icmp-seq=2. time=1. ms ^C
----127.0.0.1 PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss round-trip
(ms) min/avg/max = 0/0/1
```

Ha válaszként a „ping: no answer from…” vagy a „100% packet loss” üzenetet kapjuk, akkor a gépen egyáltalán nincs telepítve hálózati összetevő. A `127.0.0.1` a belső visszahurkolási cím, és semmi köze ahhoz, hogy a számítógép kapcsolódik-e fizikailag egy hálózathoz vagy sem. Ha nem fut le a teszt, akkor komoly helyi problémára gyanakodhatunk. Lehetséges, hogy a TCP/IP nincs telepítve vagy rosszul van konfigurálva. Ha Unix kiszolgálón lép fel ez a hiba, akkor nézzünk utána az operációs rendszer dokumentációjában, ha viszont egy Windows ügyfélnél tapasztaljuk ezt, akkor a 3. fejezetben olvassuk el, hogyan telepíthetjük a TCP/IP protokollt.

Hálózati rendszergazdáknak ajánlható olvasmány Craig Hunt *TCP/IP Network Administration*, valamint Craig Hunt és Robert Bruce Thompson *Windows NT TCP/IP Network Administration* című könyve (mindkettő kiadója az O'Reilly).

Helyi névszolgáltatás tesztelése a ping paranccsal

Próbálkozzunk most a `ping localhost` paranccsal a Samba kiszolgálón. A `localhost` a 127.0.0.1 visszahurkolási cím gazdaneve, és a névfeloldónak erre a névre kell feloldania a címet. Miután beírtuk a `ping localhost` parancsot, a kimenetnek ehhez hasonlónak kell lennie:

```
server%  ping localhost
PING localhost: 56 data bytes  64 bytes from localhost (127.0.0.1):
icmp-seq=0. time=0. ms  64 bytes from localhost (127.0.0.1):
icmp-seq=1. time=0. ms  64 bytes from localhost (127.0.0.1):
icmp-seq=2. time=0. ms  ^C
```

Ha sikeresen lefutott a parancs, akkor végezzük el ugyanezt a tesztet az ügyfélnél is. Ellenkező esetben:

- Ha az „unknown host: localhost" üzenetet kapjuk, akkor valamilyen probléma van a localhost gazdanév érvényes IP címmé való átalakításával. (Lehet, hogy csak az a baj, hogy hiányzik egy bejegyzés egy helyi *hosts* fájlból.) Lapozzunk át a fejezet *„Hibakeresés a névszolgáltatásban"* című részére.
- Ha a „ping: no answer" vagy a „100% packet loss" üzenetet kapjuk, de a ping 127.0.0.1 parancs sikeresen lefutott, akkor a névfeloldó elvégzi a név feloldását, csak nem a megfelelő címmé. Ellenőrizzük azt a fájlt vagy adatbázist (ez Unix rendszerben tipikusan az */etc/hosts*), amelyet a névszolgáltató a névfeloldáshoz használ, hogy lássuk, rendben vannak-e benne a bejegyzések.

Hálózati kártya tesztelése a ping paranccsal

Ellenőrizzük most a kiszolgáló hálózati IP címét magából a kiszolgálóból. Ennek ugyanazt az eredményt kell szolgáltatnia, mintha a 127.0.0.1 címre adtuk volna ki a ping parancsot:

```
server% ping 192.168.220.100
PING 192.168.220.100: 56 data bytes 64 bytes from 192.168.220.100
(192.168.220.100):
icmp-seq=0. time=1. ms 64 bytes from 192.168.220.100
(192.168.220.100):
icmp-seq=1. time=0. ms 64 bytes from 192.168.220.100
(192.168.220.100):
icmp-seq=2. time=1. ms ^C
----192.168.220.100 PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss round-trip
(ms) min/avg/max = 0/0/1
```

Ha ez működik a kiszolgálón, akkor ismételjük meg az ügyfélnél is. Ellenkező esetben:

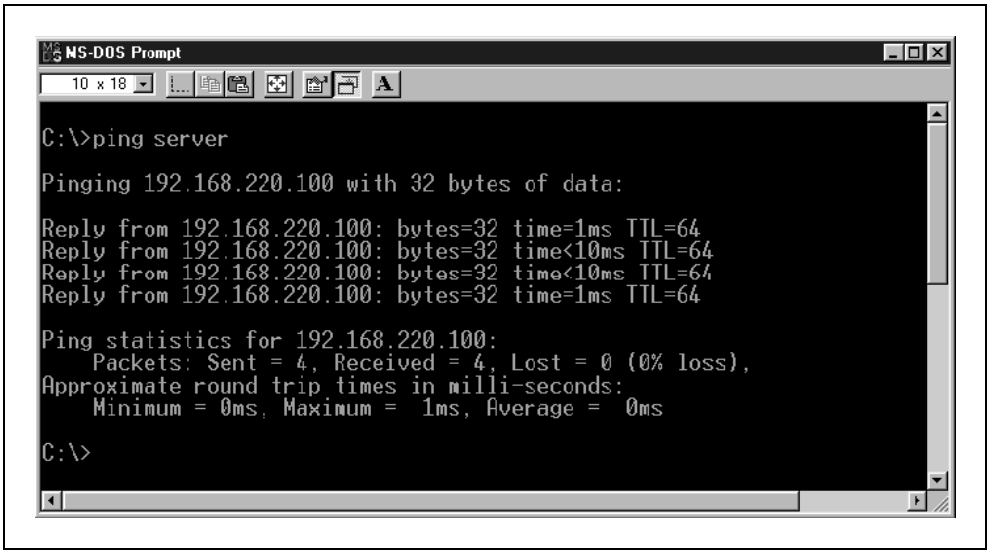
- Ha `ping` *`hálózati_ip`* parancs akár a kiszolgálón, akár az ügyfélnél nem futott le rendben, de a ping 127.0.0.1 sikeresen végrehajtódott, akkor az illető számítógépben lévő Ethernet kártyával kapcsolatos TCP/IP hibával van dolgunk. Lapozzuk fel a hálózati kártya és a gazdagép operációs rendszerének dokumentációját, és nézzük utána, hogy megfelelően konfiguráltuk-e a kártyát. Azzal azonban legyünk tisztában, hogy egyes operációs rendszereknél akkor is rendben lefuthat a *ping* parancs, ha a gép nem kapcsolódik a hálózathoz, ezért ez a vizsgálat nem deríthet mindig fényt az összes hálózati problémára.

Kapcsolódások tesztelése a ping paranccsal

Adjuk ki most a ping parancsot úgy, hogy a kiszolgáló IP címe helyett nevét írjuk be a parancssorba. A parancsot először a kiszolgálón, majd az ügyfélnél futtassuk le. Ez a hálózati kártya működésének általános tesztje:

```
server% ping server
PING server.example.com:0
     56 data bytes 64 bytes from server.example.com (192.168.220.100):
icmp-seq=0. time=1. ms 64 bytes from server.example.com
(192.168.220.100):
icmp-seq=1. time=0. ms 64 bytes from server.example.com
(192.168.220.100):
icmp-seq=2. time=1. ms ^C
----server.example.com PING Statistics----
3 packets transmitted, 3 packets received, 0% packet loss round-trip
(ms) min/avg/max = 0/0/1
```

Microsoft Windows rendszerben a kiszolgálóra kiadott *ping* parancs a 9.1. ábrán láthatóhoz hasonló eredménnyel fut le.



9.1. ábra. Windows ügyfélnél a Samba kiszolgálóra kiadott ping parancs kimenete

Ha a parancs sikeresen lefutott, akkor ebből öt tény következik:

1. A gazdanevet (vagyis a „server”-t) megtalálta a helyi névkiszolgáló.
2. A gazdanév kiegészült a teljes nevére *(server.example.com)*.
3. A kiszolgáló visszaküldte a gazdagép IP címét (192.168.220.100).
4. Az ügyfél négy, egyenként 56 bájtos UDP/IP csomagot küldött a Samba kiszolgálóra.
5. A Samba kiszolgáló mind a négy csomagra válaszolt.

Ha sikertelen a teszt, akkor különböző hibák lehetnek a hálózatban:

- Ha a „ping: no answer" vagy „100% packet loss" hibaüzenetet kapjuk, akkor vagy a kiszolgáló, vagy az ügyfél nem kapcsolódik a hálózathoz, vagy az egyikük címe hibás. Ellenőrizzük az egyes gépeken a `ping` parancs által visszaadott címeket, és győződjünk meg arról, hogy ezek megegyeznek az eredetileg megadott címekkel.
  Lehetséges, hogy hibásan adtunk meg egy címet. Adjuk ki az `arp -a` parancsot, hogy lássuk, kapunk-e bejegyzést a másik gépről. E parancs `arp` neve az Addres Resolution Protocol (címfeloldó protokoll) kezdőbetűiből álló rövidítés. Az `arp -a` parancs a helyi gép által ismert összes címet felsorolja. Próbálkozzunk a következőkkel:
  - Ha a „192.168.220.100 at (incomplete)" vagy ehhez hasonló üzenetet kapunk, akkor a gép számára ismeretlen a 192.168.220.100 IP címhez tartozó Ethernet cím. Ez a kapcsolódás teljes hiányát jelenti, és valószínűleg a TCP/IP protokoll legalsó szintjén, az Ethernet szintjén van a probléma. Az ilyen hibákkal a *TCP/IP Network Administration* című könyv (O'Reilly) 5. és 6. fejezete foglalkozik.
  - Ha a „server (192.168.220.100) at 8:0:20:12:7c:94" vagy ehhez hasonló üzenetet kapunk, akkor a kiszolgáló esetenként elérhető, vagy másik gép válaszol a nevében. Annyit azonban már tudhatunk, hogy a *ping* parancs rendben lefuthat: valamilyen időszakos hálózati vagy címfeloldási problémával lehet dolgunk.
  - Ha az *arp* parancs által visszaküldött Ethernet cím nem egyezik meg a várt címmel, akkor keressük meg és kézzel javítsuk ki a címet.
- Ha a ping parancs magukon az egyes gépeken rendben lefut, de két különböző gép között már nem, akkor az őket összekötő hálózatban lehet a hiba.
- Ha a „ping: network unreachable" vagy az „ICMP Host Unreachable" üzenetet kapjuk, akkor a hiba több hálózatot is érint.
  Lehetőség szerint ne próbáljunk hibát keresni olyan SMB ügyfeleknél és kiszolgálókon, amelyek különböző hálózatokhoz kapcsolódnak. A hibakeresést egyetlen hálózat gépeire korlátozzuk. Az alábbi három teszt abból a feltételezésből indul ki, hogy két hálózat között keresünk hibát:
  a) Először a válaszokra vonatkozó teszteket végezzük el a korábban leírtak szerint. Ha ezekkel nem tudjuk behatárolni a hiba okát, akkor a következő hibák lehetségesek: rossz valamelyik cím, rossz az alhálózati cím, a hálózat nem működik, vagy a vizsgálatnak útját állja egy tűzfal.
  b) Ellenőrizzük mind a küldő, mind a fogadó gépeknél a címeket, hogy lássuk, nincs valamilyen nyilvánvaló hiba. Ha mind a küldő, mind a fogadó gép ugyanahhoz a hálózathoz tartozik, akkor mindkettőnek azonos alhálózati maszkkal kell rendelkeznie, és a *ping* parancsnak a helyes címet kell visszajelentenie. Ha hibás ez a cím, akkor ki kell javítani. Ha helyes a cím, akkor a programokat valamilyen hibás alhálózat zavarja össze. Lásd a fejezet későbbi, *„Hálózati maszkok"* című részét.
  c) Ha a parancsok továbbra is azt jelzik vissza, hogy elérhetetlen a hálózat, és az előző két hiba nem fordul elő, akkor az egyik hálózat nem érhető el a másikból. Az ilyen hiba elhárítása a hálózat rendszergazdájának a dolga.
- Ha az „ICMP Administratively Prohibited" üzenetet kapjuk, akkor vagy egy tűzfalba, vagy egy hibásan konfigurált forgalomirányítóba (router) ütköztünk. Ebben az esetben a hálózat biztonságáért felelős személlyel kell felvennünk a kapcsolatot.

- Ha az „ICMP Host redirect” üzenetet kapjuk, és a *ping* parancs a csomagok kiküldését jelentette, akkor nincs nagy baj: mindössze annyi történt, hogy a kérésünk a hálózat valamely másik részére átirányításra került.
- Ha a kérésünk átirányításra került, és a *ping* parancs nem jelez választ, akkor a hálózat egyik helye sem válaszol a kérésünkre. Tekintsük úgy ezt az üzenetet, mintha elérhetetlen lenne a hálózat, és vizsgáljuk meg a címeket és az alhálózatokat.
- Ha az „ICMP Host Unreachable from gateway *átjáró_neve*” üzenetet kapjuk, akkor a *ping* parancs csomagjai másik hálózatra kerültek átirányításra, de a másik gép nem válaszol, és a forgalomirányító a maga részéről hibát jelez. Tekintsük ezt az üzenetet is úgy, mintha elérhetetlen lenne a hálózat, és vizsgáljuk meg a címeket és az alhálózatokat.
- Ha a „ping: unknown host *gazdanév*” üzenetet kapjuk, akkor ismeretlen a gépünk neve. Ebből névszolgáltatási problémára következtethetünk, ami nem befolyásolja a localhost vizsgálatot. Vessünk egy pillantást a fejezet későbbi, *„Hibakeresés a névszolgáltatásban”* című részébe.
- Ha a *ping* parancs esetenként sikeresen, máskor hibásan fut le, akkor vagy valamilyen időszakos hiba van a gépek között, vagy túlterhelt a hálózat. Futtassuk többször a parancsot, hogy lássuk, meghaladja-e a 3 százalékot a hibás csomagok száma. Amennyiben igen, akkor vegyük fel a kapcsolatot a hálózati rendszergazdával. Ha viszont csak néhány csomag hibás, vagy tudjuk, hogy valamilyen nagyméretű program futása terheli le a hálózatot, akkor legyünk türelemmel: a *ping* programot úgy tervezték meg, hogy esetenként elveszíthet csomagokat.
- Ha a ping parancsot történetesen egy *client.example.com* gépre adtuk ki, és válaszul az „smtsvr.antares.net is alive” vagy ehhez hasonló üzenetet kapunk, akkor vagy valaki másnak a címét használjuk, vagy a géphez több név és cím tartozik. Ha rossz a cím, akkor ebben a névszolgáltató a hibás; javítsuk ki a névszolgáltató adatbázisában a címet, hogy az a megfelelő gépre mutasson. Erről bővebben a fejezet későbbi, *„Hibakeresés a névszolgáltatásban”* című részében olvashatunk.
  A kiszolgálók gyakran *többlakiak*: egynél több hálózathoz is kapcsolódhatnak, és mindegyik hálózatban más-más nevet használnak. Ha egy ilyen többlaki kiszolgálótól váratlan néven kapunk választ, akkor vizsgáljuk meg, hogy a címe szerepel-e a hálózatunkban (lásd a fejezet *„Hálózati maszkok”* című későbbi részét). Amennyiben szerepel, akkor – mind a teljesítmény, mind a megbízhatóság érdekében – ezt a címet, és ne a másik hálózatban lévő címét használjuk.
  A kiszolgálóknak egyetlen Ethernet címhez is több nevük lehet, főleg akkor, ha webkiszolgálók. Bármennyire furcsa is, ez nem okoz problémát. Célszerű azonban a hivatalos (és állandó) nevét használnunk; az álnevek változhatnak.
- Ha minden rendben működik, de a jelentett IP cím a 127.0.0.1, akkor névszolgáltatási hibával van dolgunk. Ez általában akkor fordul elő, ha egy operációs rendszer telepítő programja olyan */etc/hosts* sort hoz létre, mint amilyen például a `127.0.0.1 localhost` *gazdanév-tartománynév*. A localhost sornak `127.0.0.1 localhost` vagy `127.0.0.1 localhost loghost` alakúnak kell lennie. Javítsuk ki, nehogy hibát okozzon abban a folyamatban, amelyben a gépek egyeztetik, melyikük legyen a főtallózólista vezetője és melyikük a főtallózó. Az ilyen problémák más (kétértelmű) hibákat is okozhatnak a későbbi tesztek során.

Ha a kiszolgálón lefutott a teszt, akkor ismételjük meg az ügyfélnél is.

Hibakeresés a TCP-ben

Miután a *ping* parancs segítségével teszteltük az IP-t, az UDP-t és a névszolgáltatást, itt az ideje, hogy a TCP-t is teszteljük. Amíg a *ping* és a tallózás az ICMP (Internet Control Message Protocol) és az UDP (User Datagram Protocol) protokollokat használja, addig a fájl- és nyomtatószolgáltatások a kapcsolódásokat a TCP (Transmission Control Protocol) protokoll szerint hozzák létre. Mindegyik protokoll alapja az alsóbb szintű IP protokoll, és ezek mindegyike a névszolgáltatástól függ. A TCP-t legkényelmesebben egy FTP (fájlátviteli protokoll) program segítségével tesztelhetjük (*természetesen csak akkor, ha fut FTP kiszolgáló az adott számítógépen - a lektor megjegyzése*).

A TCP tesztelése FTP-vel

Hozzunk létre FTP-s kapcsolatot először a kiszolgálóról magához a kiszolgálóhoz, majd az ügyféltől a kiszolgálóhoz:

```
server%  ftp server
Connected to server.example.com.
220  server.example.com  FTP  server  (Version  6.2/OpenBSD/Linux-0.10)
ready.
  Name (server:davecb):
331 Password required for davecb.
Password:
230 User davecb logged in.
  ftp> quit
221 Goodbye.
```

Ha ez sikerült, akkor ugorjunk át a fejezet *„Hibakeresés a kiszolgáló démonjaiban”* című részére. Ellenkező esetben:

- Ha a „server: unknown host” üzenetet kapjuk, akkor nem sikerült a névfeloldás. Menjünk vissza a *„Helyi névszolgáltatás tesztelése a ping paranccsal”* című fejezetrészre, és futtassuk újra az ott bemutatott teszteket, hogy lássuk, miért hibás a helyi névfeloldás.
- Ha az „ftp: connect: Connection refused” üzenetet kapjuk, akkor a gép nem futtat egy FTP démont. Ez meglehetősen szokatlan lenne Unix kiszolgálókon. Megpróbálkozhatunk még azzal, hogy FTP helyett telnettel kapcsolódunk a géphez; ennek hasonlóak az üzenetei, és a telnet is TCP protokollt használ.
- Ha hosszú szünet következett, majd az „ftp: connect: Connection timed out” üzenetet kapjuk, akkor a gép nem érhető el. Térjünk vissza a fejezet *„Kapcsolódások tesztelése a ping paranccsal” című* fejezetrészre.
- Ha az „530 Logon Incorrect” üzenetet kapjuk, akkor a kapcsolódás sikeres volt, csak éppen más probléma merült fel. Feltehetően hibás felhasználónevet vagy jelszót adtunk meg. Próbáljuk újra, és győződjünk meg arról, hogy a Unix kiszolgálóról a felhasználónevünket használtuk, és helyesen írtuk be a jelszavunkat.

Hibakeresés a kiszolgáló démonjaiban

Miután meggyőződtünk arról, hogy helyesen működik a TCP hálózat, következő lépésként azt vizsgáljuk, hogy futnak-e a démonok a kiszolgálón. Ehhez három különálló tesztet kell futtatnunk, mert önmagában egyik sem garantálja ezek helyes működését.

Akkor lehetünk biztosak a démonok futásában, ha tudjuk, hogy:

1. A démonok elindultak.
2. A démonokat az operációs rendszer regisztrálja vagy hozzáköti egy TCP/IP-hez.
3. A démonok figyelő állapotban vannak.

Mielőtt elkezdenénk

Először vizsgáljuk meg a naplókat. Ha elindítottuk a démonokat, tartalmazniuk kell az „smbd version *valamilyen_szám* started" üzenetet. Amennyiben nem így lenne, újra kell indítanunk a Samba démonjait.

Ha a démon azt jelenti magáról, hogy elindult, akkor keressünk egy „bind failed on port 139 socket_addr=0 (Address already in use)" bejegyzést. Ez azt jelenti, hogy egy másik démon már el lett indítva a 139-es porton (*smbd*). Az *nmbd* is hasonló hibát jelezne, ha nem tudna kötődni a 137-es porthoz. Ekkor vagy az történt, hogy kétszer indítottuk el a démonokat, vagy az *inetd* próbált meg elindítani egy démont. Ez utóbbira azonnal kitérünk.

Démon processzeinek keresése a ps paranccsal

Következő lépésként meg kell néznünk, hogy elindultak-e a démonok. Adjuk ki a kiszolgálón a `ps` parancsot a géptípusra vonatkozó `long` paraméterrel (ez általában a `ps ax` vagy a `ps -ef`), hogy lássuk, fut-e az *smbd* vagy az *nmbd* valamelyike. Ekkor az alábbihoz hasonló jelenhet meg:

```
server% ps ax
  PID TTY STAT TIME   COMMAND
  1   ?   S    0:03   init [2]
  2   ?   SW   0:00   (kflushd)
(...processzek sorai...)
  234 ?   S    0:14   nmbd -D3
  237 ?   S    0:11   smbd -D3
(...további sorok, köztük további smbd sorok...)
```

A fenti példából látható, hogy mind az *smbd*, mind az *nmbd* démon önálló démonként elindult (a `-D` utal erre) 3-as naplózási szinttel.

Portokhoz kötött démonok keresése

Következő lépésként a démonoknak regisztráltatniuk kell magukat az operációs rendszerrel, hogy hozzáférhessenek a TCP/IP portokhoz. A `netstat` parancs ad tájékoztatást ennek a megtörténtéről. Futtassuk a kiszolgálón a `netstat -a` parancsot, és keressük meg a `netbios`, a `137` vagy a `139` tartalmú sorokat:

```
server% netstat -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address       Foreign Address      (state)
udp   0      0       *.netbios-          *.*
tcp   0      0       *.netbios-          *.*
LISTEN
tcp   8370   8760    server.netbios-     client.1439
ESTABLISHED
```

vagy:

```
server% netstat -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address       Foreign Address       (state)
udp   0      0       *.137               *.*
tcp   0      0       *.139               *.*
LISTEN
tcp   8370   8760    server.139          client.1439
ESTABLISHED
```

Sok hasonló sor között kell lennie legalább egy olyan UDP sornak, amely tartalmazza a `*.netbios-` vagy a `*.137` szövegelemet. Ez jelzi azt, hogy az *nmbd* démon regisztrálva van, és (remélhetőleg) várja, hogy válaszolhasson a kérésekre. Emellett kell lennie legalább egy olyan TCP sornak, amelyben megtalálható a `*.netbios-` vagy a `*.139` szövegelem, és ez is figyelő (LISTEN) állapotú. Ez azt jelenti, hogy az *smbd* démon is fut, és figyeli a kapcsolódásokat.

Más TCP sorok is lehetnek a kimenetben, amelyek az *smbd* más ügyfelekkel fennálló kapcsolatait jelzik (mindegyik ügyfélhez egy-egy sor). Ezek a sorok általában ESTABLISHED (= létrejött) állapotot jeleznek. Ha vannak ilyen állapotú *smbd* sorok, akkor biztosak lehetünk abban, hogy fut az *smbd* démon. Ha csak egyetlen sornak LISTEN az állapota, akkor még nem lehetünk biztosak ebben. Ha mindkét sor hiányzik, akkor az egyik démonnak nem sikerült elindulnia – ekkor ellenőrizzük a naplófájlokat, és lapozzunk vissza a 2. fejezethez.

Ha minden egyes ügyfélhez tartozik egy sor, akkor az vagy a Samba démonjától, vagy az *inetd* szuperdémonból származhat. Könnyen elképzelhető, hogy az *inetd* indító fájl olyan sorokat tartalmaz, amelyek elindítják a Samba démonjait anélkül, hogy tudnánk erről. Ilyen sorok például akkor kerülhetnek be a fájlba, ha a Sambát egy Linux disztribúció részeként telepítettük. Ha a démonokat az *inetd* indítja el, akkor magunk nem indíthatjuk már ezeket. Ha megtennénk, akkor általában olyan üzenetek kerülnének a naplófájlokba, mint a „bind failed on port 139 socket_addr=0 (Address already in use)”.

Ellenőrizzük az */etc/inetd.conf* fájl tartalmát; hacsak nem kifejezetten az a szándékunk, hogy innen induljanak a démonok, akkor a fájl semmiféle `netbios-ns` (udp port 137) vagy `netbios-ssn` (tcp port 139) szövegelemet nem tartalmazhat. Az *inetd* különböző szolgáltatásokat nyújtó démon, amelynek a viselkedését az */etc/inetd.conf* fájl bejegyzései határozzák meg. Ha a rendszerünkben az *inetd* szolgáltatja a Samba valamelyik démonját, akkor a fájlban az alábbiakhoz hasonló sorok lehetnek:

```
netbios-ssn stream tcp nowait root /usr/local/samba/bin/smbd smbd
netbios-ns dgram udp wait root /usr/local/samba/bin/nmbd nmbd
```

Az smbd ellenőrzése telnettel

Kissé humoros, de legegyszerűbben mégis úgy tesztelhetjük az *smbd* démon működését, hogy egy jelentés nélküli üzenetet küldünk ki rá, és megfigyeljük, visszautasítja-e. Próbálkozzunk a következővel:

```
echo hello | telnet localhost 139
```

Ez a parancs egy hibás, de nem káros üzenetet küld az *smbd* démonra. Az üzenetben fontos a `hello` szó. Ne próbálkozzunk más üzeneteket küldeni a telnettel a portra; előfordulhat, hogy „kiakasztjuk" a processzt. A `hello` üzenet általában nem okoz problémát.

```
server% echo "hello" | telnet localhost 139
Trying
Trying 192.168.220.100 ...
Connected to localhost. Escape character is '^]'.
Connection closed by foreign host.
```

Ha egy „Connected" majd ezt követően egy „Connection closed" üzenetet kapunk, akkor a teszt sikeresen lefutott. A porton létezik egy figyelő állapotú *smbd* démon, amely visszautasítja a hibás kapcsolódási kérelmeket. Ha viszont a „telnet: connect: Connection refused" üzenetet kapjuk vissza, akkor feltehetően nem kapcsolódik démon a porthoz. Ellenőrizzük a naplófájlokat, és lapozzunk vissza a 2. fejezethez.

Az *nmbd* démont sajnos nem tesztelhetjük ilyen könnyen. Ha mind a `telnet`, mind a `netstat` teszt azt mutatja, hogy fut egy *smbd* démon, akkor jó esélyünk van arra is, hogy a `netstat` az *nmbd* futását is helyesen jelentette.

A démonok tesztelése a testparm programmal

Ha már tudjuk, hogy futnak a démonok, mindig futtassuk le a `testparm` programot, remélve, hogy a következő bejegyzéseket kapjuk:

```
server% testparm
Load smb config files from /opt/samba/lib/smb.conf
Processing section "[homes]"
Processing section "[printers]" ...
Processing section "[tmp]"
Loaded services file OK. ...
```

A `testparm` program normál esetben sorra jelenti, hogy feldolgozza a különböző szakaszokat, majd miután végzett velük, a „Loaded services file OK" üzenettel jelenti, hogy mindent rendben talált. Ha nem ez lenne a helyzet, akkor az alábbi üzenetek valamelyikét küldi – amelyet egyébként a naplófájlba is feljegyez:

„Allow/Deny connection from account (n) to service”

A *testparm* parancs akkor küldi ezt az üzenetet, ha érvényes/érvénytelen felhasználói beállításokat vettünk fel az *smb.conf* fájlba. Meg kell bizonyosodnunk arról, hogy rajta vagyunk az érvényes felhasználók listáján, a root, a bin stb. pedig az érvénytelen felhasználók listájában szerepelnek. Ha nem így lenne, akkor magunk nem tudnánk kapcsolódni, míg mások, akiknek nem volna szabad kapcsolódniuk, képesek lennének erre.

„Warning: You have some share names that are longer than eight chars"

Ez az üzenet Windows for Workgroups vagy régebbi rendszereket használó ügyfeleknek szól. Az ilyen ügyfelek nem tudnak kapcsolódni olyan megosztásokhoz, amelyeknek nyolcnál több karakterből áll a nevük. Az ilyen megosztásra irányuló kapcsolódáskor a program túlcsordulási hibát jelez, ami kissé megtévesztő – úgy hangzik, mintha memóriatúlcsordulás következett volna be.

„Warning: [name] service MUST be printable!”

Egy nyomtatómegosztásból kimaradt a `printable = yes` beállítás.

„No path in service name using [name]”

Egy fájlmegosztás nem tudja, hogy melyik könyvtárat kell szolgáltatnia a felhasználó számára, vagy egy nyomtatómegosztás nem tudja, hogy melyik könyvtárat kell használnia a várakozó nyomtatási feladatok tárolásához. Ha nem adtunk meg elérési utat, akkor a szolgáltatás a */tmp* elérési úttal próbálkozik, ami viszont nem biztos, hogy megegyezik a szándékunkkal.

„Note: Servicename is flagged unavailable”

Csak egy figyelmeztetés arra vonatkozóan, hogy egy megosztásba az `available = no` (nem elérhető) beállítást vettük fel.

„Can’t find include file [name]”

Egy `include` beállításhoz nem létező konfigurációs fájlt rendeltünk. Ha a fájlt feltétel nélkül akartuk beilleszteni, akkor emögött valószínűleg súlyos hiba húzódik meg: a megosztásnak nem a szándékaink szerinti a konfigurációja. Ha a beillesztést valamilyen változótól, például a `%a` (architektúra) változótól tettük függővé, akkor meg kell állapítanunk, hogy például egy hiányzó Windows for Workgroups konfigurációs fájl okozza-e a problémát. Gyakran nem ez a hiba oka.

„Can’t copy service name, unable to copy to itself”

Az *smb.conf* fájl egyik szakaszát önmagára akartuk másolni.

„Unable to copy service-source not found: [name]”

Arra figyelmeztet, hogy a `copy =` beállításhoz nem rendeltünk szakaszt, vagy hibásan írtuk be a szakasz nevét.

„Ignoring unknown parameter name"
: Általában egy elavult, hibásan beírt vagy nem támogatott beállításra hívja fel a figyelmet.

„Global parameter name found in service section"
: Arra figyelmeztet, hogy csak globális hatókörrel rendelkező paramétert adtunk meg egy egyedi megosztásban. A Samba az ilyen paramétert figyelmen kívül hagyja.

Miután lefuttattuk a `testparm` tesztet, ismételjük meg (pontosan) három paraméterrel: az *smb.conf* fájl nevével, az ügyfelünk nevével és az IP címével:

```
testparm samba_könyvtár/lib/smb.conf client 192.168.220.105
```

Ez még egy tesztet futtat le, amely megvizsgálja a gazdanevet és -címet a `host allow` és a `host deny` beállításokra vonatkozóan, és az „Allow/Deny connection from account *fióknév* to service" üzenetet küldheti az illető ügyfélgéppel kapcsolatban. Az üzenet azt jelzi, hogy érvényes/érvénytelen gazdabeállítások vannak az *smb.conf* fájlban, amelyek nem engedélyezik az ügyfélgépnek a szolgáltatás elérését. A `testparm /usr/local/lib/kísérleti.conf` teszt lefuttatása is hasznos módja egy *kísérleti smb.conf* fájl tesztelésének, mielőtt még a valós szerepében használnánk a konfigurációs fájlt.

Hibakeresés az SMB kapcsolatokban

Miután már tudjuk, hogy elindultak a démonok, meg kell győződnünk arról, hogy helyesen is futnak. A vizsgálatot a *samba_directory/lib* könyvtárban lévő *smb.conf* fájllal kezdjük el.

A minimális smb.conf fájl

A most következő tesztek során feltételezzük, hogy van egy tesztelésre alkalmas `[temp]` megosztásunk, továbbá legalább egy fiókunk. Ezek az alábbi *smb.conf* fájllal valósíthatók meg:

```
[global]
    workgroup = EXAMPLE
    security = user
    browsable = yes
    local master = yes
[homes]
    guest ok = no
    browseble = no
[temp]
    path = /tmp
    public = yes
```

Figyelmeztetés: A `[temp]` megosztásban szereplő `public = yes` beállítás csak tesztelési célokat szolgál. Feltehetően nem az a szándékunk, hogy bármilyen, fiókkal nem rendelkező személy tárolhassa a dolgait a Samba kiszolgálónkon; miután elvégeztük a tesztet, töröljük a beállítást, vagy megjegyzéssé minősítsük át.

Tesztelés helyileg az smbclient programmal

Elsőként vizsgáljuk azt, hogy képes-e a kiszolgáló a saját szolgáltatásainak (megosztásainak) felsorolására. Futtassuk az `smbclient` parancsot az `-L localhost` kapcsolóval, hogy önmagához kapcsolódjon, és az `-U%` kapcsolóval, ami a vendégfelhasználót jelenti. Ekkor a következőt kell látnunk:

```
server% smbclient -L localhost -U%
Server time is Wed May 27 17:57:40 1998 Timezone is UTC-4.0
Server=[localhost]
User=[davecb]
Workgroup=[EXAMPLE]
Domain=[EXAMPLE]
    Sharename      Type      Comment
    ---------      -----     -------------------------
    temp           Disk
    IPC$           IPC       IPC Service (Samba 1.9.18)
    homes          Disk      Home directories
This machine does not have a browse list
```

Ha ezt a kimenetet kaptuk, akkor térjünk rá a *„Kapcsolódások tesztelése az smbclient programmal”* című következő fejezetrészre. Ellenkező esetben, ha valamilyen hiba történt, vizsgáljuk a következőket:

- Ha a „Get_hostbyname: unknown host localhost” üzenetet kapjuk, akkor vagy hibásan írtuk be a nevet, vagy valamilyen más probléma fordult elő (aminek a *„Helyi névszolgáltatás tesztelése a ping paranccsal”* című fejezetrészben nézhetünk utána). Az utóbbi esetben lapozzuk fel a *„Hibakeresés a névszolgáltatásban”* című fejezetrészt.
- Ha a „Connect error: Connection refused” üzenetet kapjuk, akkor a kiszolgálót ugyan sikerült megtalálni, de nem fut rajta az *nmbd* démon. Lapozzunk vissza a *„Hibakeresés a kiszolgáló démonjaiban”* című fejezetrészre, és teszteljük újra a démonokat.
- Ha a „Your server software is being unfriendly” üzenetet kapjuk, akkor a kapcsolatfelvételt kérő első csomagra érthetetlen válasz érkezett a kiszolgálóról. Lehetséges, hogy összeomlott a kiszolgáló, vagy hibásan lett elindítva. A hiba okának kiderítéséhez nézzük át a naplófájlokat, és keressük a következőket:
  - Hibás parancssori paraméterek lettek megadva az *smbd* parancshoz; lásd az *smbd* kézikönyvét.
  - Az *smb.conf* fájlban olyan hiba van, amely megakadályozza az *smbd* indulását. Mindig vizsgáljuk át a változtatásokat, ahogyan ezt a *„A démonok tesztelése a testparm programmal”* fejezetrészben tettük.
  - Hiányoznak azok a könyvtárak, melyek a Samba a napló- és a zárolási fájljait tárolják.
  - Már van egy kiszolgáló a porton (a 139-es port az *smbd*, a 137-es port az *nmbd* démoné), ami megakadályozza az indulását.

- Ha a démonokat nem önállóan, hanem az *inetd* szuperdémonból indítottuk, akkor a kézikönyvekben nézzünk utána, hogy nincsenek-e hibák az */etc/inetd.conf* és az */etc/services* bejegyzéseiben.
- Ha felszólítást kapunk a jelszó megadására, akkor hibásan van beállítva a vendégfiók. A `%U` változó azt közli az *smbclient* programmal, hogy „null bejelentkezést" végezzen, ami megköveteli, hogy létezzen vendégfiók, de nem igényli, hogy bármilyen privilégium tartozzon hozzá.
- Ha az „SMBtconX failed. ERRSRV-ERRaccess" üzenetet kapjuk, akkor nincs engedélyünk a kiszolgáló eléréséhez. Ez általában azt jelenti, hogy vagy olyan `valid hosts` beállítást használtunk, amely nem foglalja magában a kiszolgálót, vagy olyan `invalid hosts` beállítást használtunk, amely viszont magában foglalja a kiszolgálót. Adjuk ki újra a `testparm smb.conf` *`saját_gazdanév saját_ip_cím`* parancsot (lásd *„A démonok tesztelése a testparm paranccsal"* című fejezetrészt), és javítsuk ki a nem szándékos tiltásokat.

Kapcsolódások tesztelése az smbclient programmal

Adjuk ki az `smbclient \\`*`server`*`\temp` parancsot, amely kapcsolatot hoz létre a kiszolgálónk */temp* megosztásával, hogy lássuk, tudunk-e kapcsolódni egy fájlmegosztáshoz. A következő választ kell kapnunk:

```
server% smbclient '\\server\temp'
Server time is Tue May  5 09:49:32 1998 Timezone is UTC-4.0 Password:
smb: \> quit
```

- Ha válaszként a „Get_Hostbyname: Unknown host name", „Connect error: Connection refused" vagy a „Your server software is being unfriendly" üzenetek valamelyikét kapjuk, akkor a diagnózis felállításához lapozzunk vissza a *„Tesztelés helyileg az smbclient programmal"* című fejezetrészre.
- Ha a „servertemp: Not enough '\' characters in service" üzenetet kapjuk, akkor valószínűleg nem tettük idézőjelek közé a címet, ezért a Unix átugrotta a fordított törtvonal (\) karaktereket. A parancs az alábbi alakban is beírható:

```
smbclient \\\\server\\temp
```

vagy

```
smbclient //server/temp
```

Ezt követően a `Password` promptra adjuk meg a unixos fiókunk jelszavát. Ha ez után egy `smb\>` promptot kapunk, akkor a parancs sikeresen lefutott. Lépjünk ki a `quit` beírásával, majd térjünk rá a *„Kapcsolódások tesztelése a NET USE paranccsal"* című fejezetrészre. Ha viszont az „SMBtconX failed. ERRSRV-ERRinvnetname" üzenetet kapjuk, akkor a probléma az alábbiak valamelyike lehet:

- Hibás megosztásnév: lehet, hogy elírtuk a megosztás nevét vagy túl hosszú a név, lehet, hogy vegyesen használtunk nagy- és kisbetűket, de az is lehet, hogy nem áll rendelkezésre a megosztás. Futtassuk le újra a *testparm* programot (lásd *„A démonok tesztelése a testparm programmal"* című fejezetrészt).

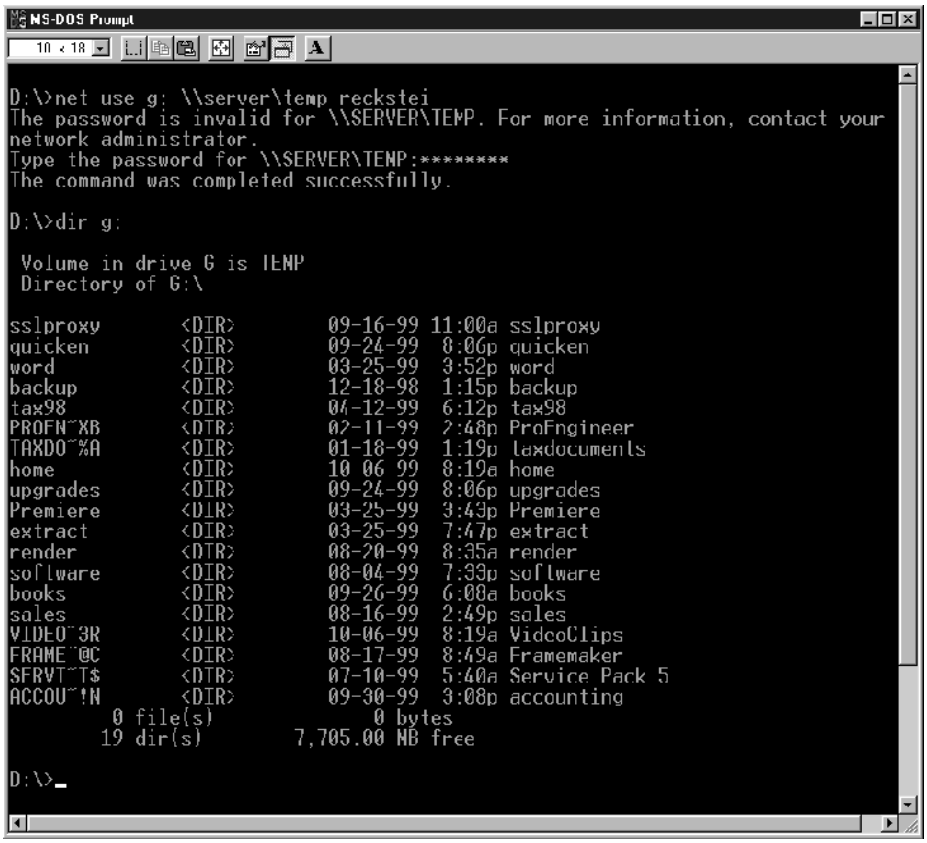
- `security = share` beállítás, amelybe esetleg fel kellett volna venni az *smbclient* parancshoz az *`-U saját_fiók`* paramétert, vagy tudni kellett volna egy *temp* nevű unixos fiók jelszavát.
- Hibás felhasználónév.
- Hibás jelszó.
- Olyan `invalid users` vagy `valid users` beállítás az *smb.conf* fájlban, amely nem engedi, hogy kapcsolódjunk a fiókunkhoz. Teszteljük újra a `testparm smb.conf` *`saját_gazdanév saját_ip_cím`* parancs futtatásával (lásd *„A démonok tesztelése a testparm paranccsal”* című fejezetrészt).
- Olyan `valid hosts` beállítás, amely nem foglalja magában a kiszolgálót, vagy olyan `invalid hosts` beállítás, amely tartalmazza a kiszolgálót. Ebben az esetben is futtassuk újra a *testparm* programot.
- Hitelesítési probléma, ha például shadow jelszavakat vagy PAM (Password Authentication Module) modulokat használ a kiszolgáló, viszont a Samba úgy lett lefordítva, hogy ne használja ezeket a jellemzőket. Ez ugyan ritka eset, de előfordulhat, ha egy SunOS 4 Samba bináris (nem használ shadow jelszavakat) újrafordítás nélkül fut egy Solaris rendszeren (ez használ shadow jelszavakat).
- Szerepel az `encrypted passwords = yes` beállítás a konfigurációs fájlban, de az *smbpasswd* fájl nem tartalmaz jelszót a fiókunkhoz.
- Szerepel egy null jelszóbejegyzés akár a Unix */etc/passwd* fájljában, akár az *smbpasswd* fájlban.
- A `[temp]` megosztáshoz kapcsolódunk, de nem szerepel a `guest ok = yes` beállítás az *smb.conf* fájl `[temp]` szakaszában.
- A `[temp]` megosztáshoz kapcsolódunk, mielőtt még a home könyvtárunkhoz kapcsolódtunk volna, és a vendégfiók hibásan van konfigurálva. Ha képesek vagyunk kapcsolódni a home könyvtárhoz, majd a `[temp]` megosztáshoz, akkor ez a hiba. A 2. fejezetben részletesen olvashattunk arról, miként készíthetünk el egy egyszerű Samba konfigurációs fájlt.
  A hibásan megadott vendégfiók azt követően is megakadályozhat bennünket a nyomtatásban és a tallózásban, hogy bejelentkeztünk a home könyvtárunkba.

Ennek a hibának még egy másik oka is lehet, amelynek viszont semmi köze a jelszavakhoz: az *smb.conf* fájlban a `path =` line beállítás valamilyen nem létező helyre mutat. Ezt a hibát nem jelzi ki a *testparm* program, és a Samba ügyfelek többsége is hibás felhasználói fióknak tekinti. A hibát kézzel kell kijavítani.

Miután sikerült kapcsolódnunk a `[temp]` megosztáshoz, ismételjük meg a tesztet, ezúttal a home könyvtárba való bejelentkezéssel (vagyis képezzük le a *server|davecb* hálózati meghajtót), és nézzük, előfordul-e hiba. Ha a futásához bármit meg kellett változtatnunk, akkor újra futtassuk le a tesztet a `[temp]` megosztásra.

Kapcsolódások tesztelése a NET USE paranccsal

DOS vagy Windows ügyfélnél futtassuk a net use `* \\server\temp` parancsot, hogy lássuk, tud-e kapcsolódni a kiszolgálóhoz. Felszólítást kell kapnunk a jelszó beírására, majd a „The command was completed successfully” üzenetet kell kapnunk, ha sikerült a kapcsolatfelvétel (lásd a 9.2. ábrát).



9.2. ábra. A NET USE parancs kimenete

Ha sikeresen lefutott ez a parancs, akkor hajtsuk végre a *„Kapcsolódások tesztelése a Windows Intézőjében”* című fejezetrészben leírt lépéseket. Ellenkező esetben:

- Ha a „The specified shared directory cannot be found” vagy a „Cannot locate specified share name” üzenetet kapjuk, akkor vagy hibásan írtuk be a könyvtár nevét, vagy nem létezik az *smb.conf* fájlban. Az üzenet arra is figyelmeztethet, hogy vegyesen használtunk kis- és nagybetűket a névben, szóközöket tartalmaz vagy nyolc karakternél hosszabb a név.
- Ha a „The computer name specified in the network path cannot be located” vagy a „Cannot locate specified computer” üzenetet kapjuk, akkor lehetséges, hogy rosszul írtuk be a könyvtár nevét, nem működik a névszolgáltató, valamilyen hálózati probléma lépett fel, vagy a `hosts deny` = beállításban szerepel a gazdanév.
  - Ha a hiba oka nem a hibás írásmód, akkor vissza kell lapoznunk legalább a *„Kapcsolódások tesztelése az smbclient programmal”* című fejezetrészhez, hogy kiderítsük, miért nem jött létre a kapcsolat.

  - Ha az *smbclient* program működik, akkor a névszolgáltatással lehet probléma, és előre kell ugranunk a *„Kiszolgáló tesztelése az nmblookup programmal”* című fejezetrészre, hogy megvizsgáljuk, megtalálható-e a program segítségével mind az ügyfél, mind a kiszolgáló.
- Ha a „The password is invalid for `\server\username`” üzenetet kapjuk, akkor az ügyfélnél helyileg tárolt másolat nem egyezik meg a kiszolgálón lévővel. Felszólítást kapunk a cseréjére.

A Windows 95 és 98 ügyfelek fenntartanak egy helyi jelszófájlt, ami azonban nem más, mint annak a jelszónak a tárolt másolata, amit a Samba vagy egy NT kiszolgálónak elküld hitelesítés céljából. Az üzenet felszólítása erre vonatkozik. Windows gépre továbbra is bejelentkezhetünk jelszó nélkül (NT-re viszont nem).

Ha a jelszó megadása ellenére nem fut le a teszt, akkor a megadott jelszó nem felel meg a kiszolgálón tárolt jelszóval, létezik egy `valid users` vagy `invalid users` lista, amelyik megtagadja az engedélyt, közbeavatkozik a NetBEUI protokoll, vagy a titkosított jelszóval van valamilyen probléma (lásd a következő bekezdést).

- Ha az ügyfélgépen az NT 4.0, az NT 3.5 a 3-as szervizcsomaggal, a Windows 95 a 3-as szervizcsomaggal vagy a Windows 98 valamelyike fut, és az Internet Explorer 4.0 tallózót tartalmazza a rendszer, akkor a jelszavak alapértelmezés szerint Microsoft-titkosításúak (lásd a 6. fejezet *„Jelszavak”* című részét). Ha újabban telepítettünk egy nagyobb Microsoft alkalmazást, vagy frissítettünk egy régebbit, akkor bekapcsolhatjuk a titkosított jelszavak használatát.

Mivel az Internet Explorer késznek mutatkozott arra, hogy SMB kapcsolatokon keresztül elérhetővé tegyen olyan URL címeket, mint a *fájl://valamilyen_gazda/valamilyen_fájl*, az ügyfelek egészen a Windows 95 2-es szervizcsomagjáig bezárólag nyugodtan elküldhették normál szövegként a jelszavaikat az SMB kiszolgálóra bárhol az interneten. Ez azonban nem bizonyult helyesnek, ezért a Microsoft gyorsan áttért arra, hogy csak titkosított jelszavakat engedett meg az SMB protokollban. Az azóta kibocsátott valamennyi terméke tartalmazza ezt a javítást. A titkosított jelszavakra egyébként nincs szükség, hacsak nem egy tűzfal nélküli hálózaton Internet Explorer 4.0-t használunk. Titkosítás nélküli jelszavak használatát csak a saját hálózatunkon belül engedjük meg.

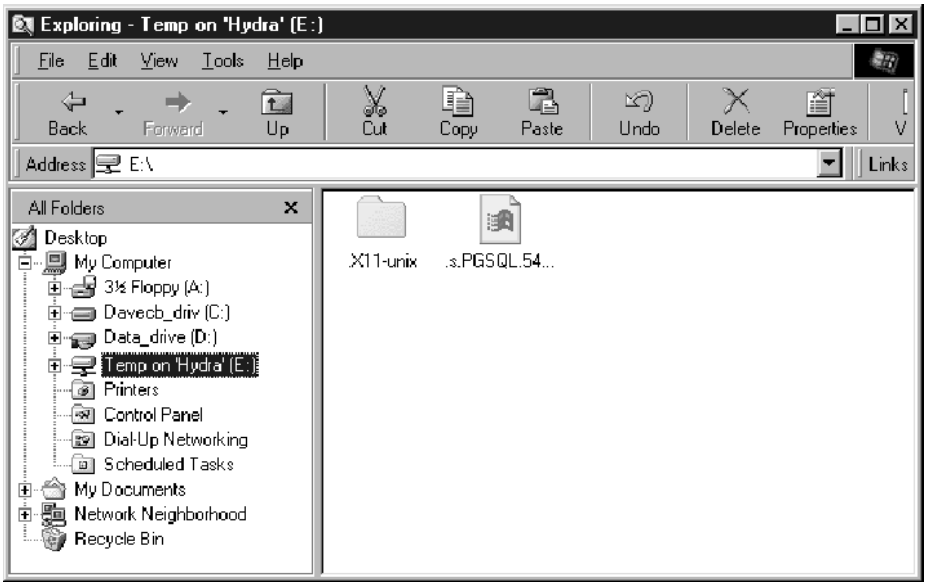
- Ha a Unixban vegyesen fordulnak elő a kis- és nagybetűk a jelszóban, akkor az ügyfél feltehetően vagy csak kisbetűket, vagy csak nagybetűket használva küldte el a jelszavát. Ha ilyenre változtatjuk a jelszavunkat, akkor ez volt a probléma. Sajnos az ügyfelek a legrégebbiek kivételével a nagybetűs jelszavakat támogatják, ezért a Samba először a csupa nagybetűsre, majd a csupa kisbetűsre alakított jelszavakkal próbálkozik. Ha vegyesen szeretnénk használni a nagy- és a kisbetűket a jelszavakban, akkor a probléma megkerüléséhez használjuk a 6. fejezetben ismertetett `password level` beállítást.

- Probléma lehet a `valid users` beállítással, amint ezt az *smbclient* programmal is tesztelték (lásd a *„Kapcsolódások tesztelése az smbclient programmal”* című fejezetrészt).
- Lehetséges, hogy a NetBEUI protokoll telepítve van az ügyfél gépére. Ebből gyakran időtúllépések és más hibák származhatnak, és a múltban problémát okoztak a jelszavak elfogadásában.

A Microsoft technológiában a „kötés” azt jelenti, hogy egy szoftver hozzá van kapcsolva egy másik szoftverhez. A Microsoft SMB ügyfél hozzá van kötve a TCP/IP protokollhoz, amint ezt a Hálózat TCP/IP tulajdonságai párbeszédablak Kötések lapján is láthatjuk. Ezzel szemben a TCP/IP itt egy Ethernet kártyához van hozzákötve. Ez nem ugyanazt jelenti, mint egy SMB démonnak egy TCP/IP porthoz való kötése.

Kapcsolódások tesztelése a Windows Intézőjében

Indítsuk el a Windows vagy az NT Intézőjét, az Eszközök menüből adjuk ki a Hálózati meghajtó csatlakoztatása parancsot, és jelöljük ki a `\\server\temp` tételt, hogy lássuk, létrehozható-e az Intézőből a kapcsolat a */tmp* könyvtárhoz. Ekkor a 9.3. ábrán láthatóhoz hasonló párbeszédablaknak kell megnyílnia. Ha így történt, akkor létrejött a kapcsolat, és átugorhatunk a *„Hibakeresés a tallózásban”* című fejezetrészre.



9.3. ábra. Kapcsolódás a /tmp könyvtárhoz a Windows Intézőjéből

Rövid figyelmeztetés: A Windows és az NT Intézője elsősorban nem diagnosztikai eszköz: figyelmeztet ugyan a hibára, de az okát nem árulja el. Ha problémába ütközünk, akkor inkább a NET USE parancsot használjuk, ami ennél sokkal bővebb tájékoztatást nyújt.

- Ha a „The password for this connection that is in your password file is no longer correct" üzenetet kapjuk, akkor a hibának a következő okai lehetnek:
  - A helyileg tárolt másolat nem egyezik meg a kiszolgálón lévővel.
  - Nem adtunk meg felhasználónevet és jelszót, amikor bejelentkeztünk az ügyfélre. Az Intéző általában a null felhasználónevet és jelszót küldi el még akkor is, ha megadtunk egy jelszót.
  - Hibásan írtuk be a jelszót.
  - Létezik egy `valid users` vagy `invalid users` lista, amelyik megtagadja az engedélyt.
  - Az ügyfélgépen az NT 4.0, az NT 3.5 a 3-as szervizcsomaggal, a Windows 95 a 3-as szervizcsomaggal vagy a Windows 98 valamelyike fut, és az Internet Explorer 4.0 tallózót tartalmazza a rendszer. Ezek mindegyike titkosított jelszót igényel.
  - A jelszóban vegyesen fordulnak elő a kis- és a nagybetűk, az ügyfél viszont vagy csupa kisbetűs, vagy csupa nagybetűs jelszót küldött.
- Ha a „The network name is either incorrect, or a network to which you do not have full access" vagy a „Cannot locate specified computer" üzenetet kapjuk, akkor az alábbi hibák lehetségesek:
  - hibásan beírt név;
  - hibásan működő szolgáltatás;
  - hibás megosztás;
  - rosszul megadott `path` sor;
  - magunkat kizáró `hosts deny` sor.
- Ha a „You must supply a password to make this connection" üzenetet kapjuk, akkor az ügyfélnél lévő jelszó nincs szinkronizálva a kiszolgálón lévő jelszóval, vagy ez az első alkalom, hogy az ügyféltől ezzel a jelszóval jelentkezünk be, és az ügyfél helyileg még nem tárolta el a jelszót.
- Ha a „Cannot locate specified share name" üzenetet kapjuk, akkor vagy rosszul írtuk be a megosztás nevét, vagy szintaktikai hibát követtünk el a megosztás specifikálásakor, de az is lehet, hogy a megosztás neve nyolc karakternél hosszabb, vagy szóközöket tartalmaz, vagy vegyesen fordulnak elő benne kis- és nagybetűk.

Ha sikerült megbízhatóan kapcsolódni a `[temp]` könyvtárhoz, akkor próbáljuk meg újra, ezúttal a home könyvtárt használva. Ha bármit változtatnunk kellett ahhoz, hogy létrehozhassuk a kapcsolatot a home könyvtárhoz, akkor újra futtassuk le a tesztet a `[temp]` könyvtárra, majd fordítva, amint ezt a *„Kapcsolódások tesztelése a NET USE paranccsal"* című fejezetrészben tettük. Ha az Intézőben nem sikerül valami, akkor – mint mindig – térjünk vissza ehhez a fejezetrészhez, és ott folytassuk a hibakeresést.

Hibakeresés a tallózásban

Végre eljutottunk a tallózáshoz. Ezt a témakört nem azért hagytuk a végére, mert ez a legnehezebb, hanem azért, mert ez a szolgáltatás egyrészt nem kötelező, másrészt pedig részben olyan protokolltól függ, ami nem garantálja a csomagok továbbítását. A tallózás-

nál nehéz diagnózist készíteni, ha nem tudnánk előre, hogy a többi szolgáltatás már hibátlanul fut.

A tallózás egyáltalán nem kötelező: ez mindössze csak az egyik módja annak, hogy megkeressük a hálózaton a kiszolgálókat és a rajtuk keresztül elérhető megosztásokat. A Unix például egészen jól boldogul tallózás nélkül is. A tallózás még azt is feltételezi, hogy az összes gép egyetlen helyi hálózatban van, ahol engedélyezett az üzenetszórás (broadcast).

A tallózás a gépeket először a nem megbízható UDP protokollon keresztül azonosítja, majd egy megbízható TCP/IP protokollon keresztül hozza létre azt a kapcsolatot, amely felsorolja az illető gépen elérhető megosztásokat.

A tallózás tesztelése az smbclient programmal

Kezdjük a megbízható kapcsolat tesztelésével. A kiszolgálóról próbáljuk meg felsorolni a saját megosztásait úgy, hogy az *smbclient* programot az `-L` *kiszolgálónév* kapcsolóval hívjuk meg. Eredményül ehhez hasonlót kell kapnunk:

```
server% smbclient -L server
Added interface ip=192.168.220.100 bcast=192.168.236.255
    nmask=255.255.255.0 Server
time is Tue Apr 28 09:57:28 1998 Timezone is UTC-4.0
Password:
Domain=[EXAMPLE]
OS=[Unix]
Server=[Samba 1.9.18]
Server=[server]
User=[davecb]
Workgroup=[EXAMPLE]
Domain=[EXAMPLE]
    Sharename      Type      Comment
    ---------      ----      -------
    cdrom          Disk      CD-ROM
    cl             Printer   Color Printer 1
    davecb         Disk      Home Directories

This machine has a browse list:
    Server         Comment
    ----------     -------
    SERVER          Samba 1.9.18

This machine has a workgroup list:
    Workgroup      Master
    ---------      -------
    EXAMPLE        SERVER
```

- Ha a parancs kimenete nem tartalmazza a *Sharename* listát, akkor a kiszolgáló nem engedélyezi a megosztásai tallózását. Ez azonban nem fordulhat elő, ha a Windows Intézőjében vagy a NET USE paranccsal már teszteltük akármelyik megosztást. Ha még

nem futtattuk volna le az `smbclient -L localhost -U%` parancsot (lásd a *„A minimális smb.conf fájl”* című alfejezetet), akkor tegyük meg most. Elég egy hibásan megadott vendégfiók, és már nem lesznek láthatók a megosztások. Ellenőrizzük az *smb.conf* fájlt is, hogy lássuk, nem tartalmaz-e egy valahol egy `browsable = no` beállítást. Induláshoz javasolható egy minimális *smb.conf* fájl használata (lásd a 4. fejezet elejét). A tallózást a `browsable = yes` beállítással engedélyeznünk kell ahhoz, hogy legalább a `[temp]` megosztást láthassuk.

- Ha nem látjuk a tallózólistát, akkor a kiszolgáló semmiféle adatot sem árul el a hálózatban lévő gépekről. Legalább egy gépnek képesnek kell lennie a tallózólisták támogatására. Ha azt szeretnénk, hogy a Samba legyen a helyi főtallózó, akkor vegyük fel a `local master = yes` beállítást az *smb.conf* fájlba.
- Ha megjelenik a tallózólista, de a listában nem szerepel a */tmp* bejegyzés, akkor feltehetően az *smb.conf* fájlban van valamilyen hiba. Lapozzunk vissza *„A démonok tesztelése a testparm programmal”* című fejezetrészhez.
- Ha a munkacsoportok listájában nem szerepel a saját munkacsoportunk neve, akkor feltehetően hibásan van beállítva a munkacsoportunk az *smb.conf* fájlban.
- Ha egyáltalán nem jelenik meg a munkacsoportok listája, akkor ellenőrizzük, nem hiányzik-e az *smb.conf* fájlból a `workgroup = EXAMPLE` beállítás.
- Ha semmi sem jelenik meg, akkor újra próbálkozzunk az `-I` *`ip_cím`* `-n` *`netbios_név`* `-W` *`munkacsoport`* `-d3` paramétersorral úgy, hogy nagybetűkkel írjuk be a NetBIOS és a munkacsoport nevét (a `d3` a 3-as naplózási szintet állítja be).

Ha még most sem kapnánk semmit, akkor a korábbi tesztelések során valamit elrontottunk. Lépjünk vissza legalább *„A TCP tesztelése FTP-vel”* vagy a *„Kapcsolódások tesztelése a ping paranccsal”* című fejezetrészig. Más esetekben:

- Ha az „SMBtconX failed. ERRSRV-ERRaccess” üzenetet kapjuk, akkor nincs engedélyünk a kiszolgálóhoz való hozzáféréshez. Ez általában azt jelenti, hogy vagy olyan `valid hosts` beállítást használtunk, amely nem foglalja magában a kiszolgálót, vagy olyan `invalid hosts` beállítást használtunk, amely viszont magában foglalja a kiszolgálót.
- Ha a „Bad password” üzenetet kapjuk, akkor feltehetően az alábbiak valamelyike áll fenn:
  - hibásan megadott `hosts allow` vagy `hosts deny` beállítás;
  - hibásan megadott `invalid users` vagy `valid users` beállítás;
  - kisbetűs jelszó és OS/2 vagy Windows for Workgroups ügyfelek;
  - hiányzó vagy érvénytelen vendégfiók.

  Ellenőrizzük a vendégfiók beállítását (lásd a *„Tesztelés helyileg az smbclient programmal”* című fejezetrészt), teszteljük az *smb.conf* fájlt a `testparm smb.conf` *`saját_gazdanév saját_ip_cím`* parancs kiadásával, és változtassuk meg vagy alakítsuk megjegyzésekké a `hosts allow`, a `hosts deny`, a `valid users` és az `invalid users` beállításokat.
- Ha a „Connection refused” üzenetet kapjuk, akkor az *smbd* démon nem fut, vagy összeomlott. Ellenőrizzük a *netstat* paranccsal a démon állapotát (lásd *„A portokhoz kötött démonok keresése”* című fejezetrészt).
- Ha a „Get_Hostbyname: Unknown host name” üzenetet kapjuk, akkor gépelési hibát követtünk el, vagy nem egyezik a Unix és a NetBIOS gazdanév, vagy a névszolgáltatás-

sal van baj. Keressük a hibát a névszolgáltatásban a *„Kapcsolódások tesztelése a NET USE paranccsal”* című fejezetrészben leírtak szerint. Ha a teszt nem talált hibát, akkor névmegadási hibára gyanakodhatunk, és ugorjunk a *„Hibakeresés a NetBIOS nevekben”* című fejezetrészre.

- Ha a „Session request failed” üzenetet kapjuk, akkor a kiszolgáló visszautasította a kapcsolatfelvételi kérelmet. Ez általában valamilyen belső hibát jelent, például azt, hogy kevés memória a művelet végrehajtásához.
- Ha a „Your server software is being unfriendly” üzenetet kapjuk, akkor a kapcsolatfelvételt kérő első csomagra érthetetlen válasz érkezett a kiszolgálóról. Lehetséges, hogy összeomlott a kiszolgáló, vagy hibásan lett elindítva. Lapozzunk vissza a *„Tesztelés helyileg az smbclient programmal”* című fejezetrészhez, ahol először elemeztük ezt a problémát.
- Ha az a gyanúnk, hogy nem fut a kiszolgáló, akkor lapozzunk vissza a *„Démon processzeinek keresése a ps paranccsal”* című fejezetrészre, hogy lássuk, miért nem válaszolnak a kiszolgáló démonjai.

Kiszolgáló tesztelése az nmblookup programmal

Ez a teszt azt a „hirdetési” rendszert vizsgálja, amelyet a Windows névszolgáltatása és tallózása használ egy gép jelenlétének megállapításához. Egy gép egy ilyen üzenetszórásos (broadcasting) eljárással „hirdeti” ki, hogy jelen van a hálózatban, és kész a szolgáltatásai nyújtására. Ez a fajta hirdetés a tallózás része, amely a nem megbízható UDP protokollt használja, és csak az Ethernethez hasonló, üzenetszórást engedélyező hálózatokban valósítható meg. Az *nmblookup* program névlekérdezéseket szór szét az általunk megadott gazdanévre vonatkozóan, és a gép IP címével és a szolgáltatás nevével tér vissza, hasonlóan a DNS *nslookup* programjához. Az alábbi példában a `-B` kapcsoló (broadcast) egy adott gépre irányítja a lekérdezést.

Először vizsgáljuk meg a kiszolgálóról önmagát a kiszolgálót. Futtassuk úgy az *nmblookup* programot, hogy a `-B` kapcsoló és a kiszolgáló nevének megadásával a lekérdezést a kiszolgálóra irányítjuk, és keresendő névként adjuk meg a `_ _SAMBA_ _` szimbolikus nevet. Válaszként a következőket kell kapnunk:

```
server% nmblookup -B server _ _SAMBA_ _
Added interface ip=192.168.220.100 bcast=192.168.236.255
    nmask=255.255.255.0
Sending queries to 192.168.220.100 192.168.220.100 _ _SAMBA_ _
```

Vissza kell kapnunk a kiszolgáló IP címét, amely után a `_ _SAMBA_ _` név áll, ami azt jelenti, hogy a kiszolgáló sikeresen hirdette ki magáról, hogy jelen van, és nyújtja a `_ _SAMBA_ _` nevű szolgáltatását – következésképp a NetBIOS névszolgáltatásnak legalább egy része működik.

- Ha a „Name_query failed to find name _ _SAMBA_ _” üzenetet kapjuk, akkor lehet, hogy rossz nevet adtunk meg a `-B` kapcsolóhoz, vagy nem fut az *nmbd*. A `-B` kapcsoló egyébként broadcast címet vesz fel: mi a példában a gép nevét adtuk meg, hogy unicast címet kapjunk vissza, és megkérdeztük, hogy létezik-e rajta `_ _SAMBA_ _` nevű szolgáltatás.

- Próbálkozzunk újra a `-B` *`ip_cím`* megadásával, és ha ez sem sikerül, akkor az *nmbd* valamilyen oknál fogva nem jelenti be a nevet. Térjünk vissza *„Hibakeresés a kiszolgáló démonjaiban”* című fejezetrészhez, hogy lássuk, fut-e az *nmbd*. Ha azt látjuk, hogy fut, akkor győződjünk meg arról, hogy az *smb.conf* fájl nem tartalmazza a `browsing = no` beállítást.

Ügyfél tesztelése az nmblookup programmal

Következő lépésként vizsgáljuk meg az ügyfél IP címét a kiszolgálóról úgy, hogy az *nmblooup* parancs paramétereiként a `-B` kapcsolóhoz az ügyfél nevét, a szolgáltatás neveként pedig a `'*'` helyettesítő karaktert adjuk meg, aminek itt „bármi” a jelentése:

```
server% nmblookup -B client '*'
Sending queries to 192.168.220.105 192.168.220.105 *
Got a positive name query response from 192.168.220.105
(192.168.220.105)
```

- Ha a „Name-query failed to find name *” üzenetet kapjuk, akkor vagy elírtunk valamit a parancsban, vagy az ügyfél gépén nincs telepítve az ügyfélszoftver, nem indult el, vagy nincs hozzákötve a TCP/IP-hez. Lapozzunk vissza a 2. vagy a 3. fejezethez, és bizonyosodjunk meg arról, hogy telepítettük az ügyfélszoftvert, és a szoftver figyeli a hálózatot.

Ha bármilyen hiba történt, ismételjük meg a parancsot a következő paraméterekkel:

- Ha az `nmblookup -B` *`client_IP_cím`* parancs sikeresen lefut, de a `-B` *`client_név`* nem, akkor névszolgáltatási problémával van dolgunk az ügyfél nevét illetően. Ugorjunk át a *„Hibakeresés a névszolgáltatásban”* című fejezetrészre.
- Ha az `nmblookup -B 127.0.0.1'*'` parancs sikeresen lefut, de a `-B client-`*`IP cím`* nem, akkor hardverhibára gyanakodhatunk, és a *ping* parancs sem működhet. Lépjünk kapcsolatba a hálózat rendszerfelügyelőjével.

A hálózat tesztelése az nmblookup paranccsal

Futtassuk újra az *nmblookup* parancsot ezúttal a -d2 (2-es hibakeresési, naplózási szint) kapcsolóval és a '*' paraméterrel. Ezúttal a programok (például az *nmbd*) azt a képességét vizsgáljuk, hogy képesek-e az üzenetszórásra. Ez lényegében egy kapcsolódási teszt, amely az alapbeállítás szerinti broadcast címre vonatkozik.

A hálózatban lévő NetBIOS/TCP-IP gazdagépek egész sorának kell visszajeleznie a „got a positive name query response” üzenettel. A Samba a rövid figyelési ideje alatt nem foghatja be az összes választ, ezért nem fogjuk mindig látni a hálózathoz kapcsolódó összes SMB ügyfelet. A legtöbbjük azonban megjelenik:

```
server% nmblookup -d 2 '*'
Added interface ip=192.168.220.100 bcast=192.168.236.255
nmask=255.255.255.0 Sending queries to 192.168.236.255
Got a positive name query response from 192.168.236.191 (192.168.236.191)
Got a positive name query response from 192.168.236.228 (192.168.236.228)
Got a positive name query response from 192.168.236.75 (192.168.236.75)
```

```
Got a positive name query response from 192.168.236.79 (192.168.236.79)
Got a positive name query response from 192.168.236.206 (192.168.236.206)
Got a positive name query response from 192.168.236.207 (192.168.236.207)
Got a positive name query response from 192.168.236.217 (192.168.236.217)
Got a positive name query response from 192.168.236.72 (192.168.236.72)
192.168.220.100 *
```

- Ha viszont ezek között nem szerepel az előzőleg vizsgált ügyfél címe, akkor hibás az alapbeállítás szerinti broadcast cím. Próbálkozzunk az `nmblookup -B 255.255.255.255 -d 2 '*'` paranccsal, ami az utolsó „mentsvár" (a broadcast cím összes bitjének 1 az értéke). Ha erre kapunk válaszokat , akkor a korábban használt broadcast cím hibás volt. Az ilyen jellegű hibákról a fejezet későbbi, *„Broadcast címek"* című részében lesz szó.
- Ha a 255.255.255.255 címmel sem boldogulunk, akkor nézzünk utána, hogy a PC-nk és a kiszolgáló nem más alhálózatban van-e, amint ezt a *„Kapcsolódások tesztelése a ping paranccsal"* című fejezetrészben tettük. Ezt a tesztet ugyanolyan kiszolgálóra és ügyfélre kellene lefuttatnunk, amely ugyanannak az alhálózatnak a része, de ha ezt nem tudnánk megtenni, akkor megpróbálhatjuk a `-B` kapcsolóval megadni a távoli hálózat broadcast címét. Az ilyen cím megkereséséről ugyancsak a *„Broadcast címek"* című fejezetrészben olvashatunk. A `-B` kapcsoló akkor működik, ha az útvonalválasztó támogatja az átirányított broadcast üzeneteket. Ha nem így lenne, akkor a tesztet csak ugyanannak a hálózatnak az ügyfelével tudjuk lefolytatni.

Ügyfél tallózásának tesztelése a net view paranccsal

Az ügyfélnél egy DOS ablakban futtassuk a `net view \\server` parancsot, hogy lássuk, tudunk-e kapcsolódni a kiszolgálóhoz, és milyen szolgáltatásokat érhetünk el rajta. Válaszként a kiszolgálón elérhető megosztások nevét kell kapnunk, amint ezt a 9.4. ábrán láthatjuk.

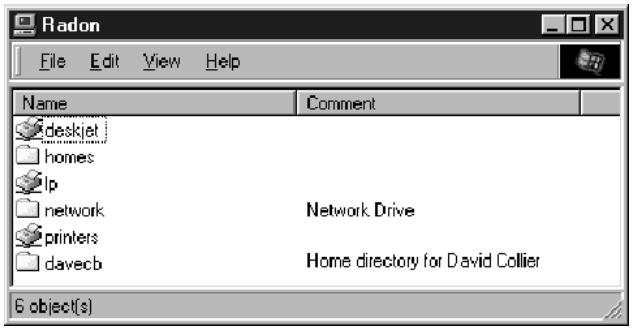
9.4. ábra. A net view parancs futása

Ha a parancs sikeresen lefutott, akkor az *„Egyéb hibalehetőségek"* című fejezetrészben leírtak szerint folytassuk a vizsgálódásainkat.

- Ha a „Network name not found” üzenetet kapjuk arra a névre, amelyet az előbb teszteltünk az *„Ügyfél tesztelése az nmblookup programmal”* című fejezetrészben, akkor magával az ügyfélszoftverrel lehet probléma. Ellenőrizzük ezt azzal, hogy az ügyfélre lefuttatjuk az *nmblookup* programot: ha ez működik, a NET VIEW viszont nem, akkor az ügyfélnél van a hiba.
- Ha nem fut le az *nmblookup* program, akkor természetesen a NetBIOS névszolgáltatóval van probléma, amint erről a *„Hibakeresés a NetBIOS nevekben”* című fejezetrészben olvashatunk.
- Ha a „You do not have the necessary access rights” vagy a „This server is not configured to list shared resources” üzenetek valamelyikét kapjuk, akkor vagy rosszul van konfigurálva a vendégfiók (lásd *„Tesztelés helyileg az smbclient programmal”*), vagy olyan `hosts allow` vagy `hosts deny` beállításokat vettünk fel, amelyek megakadályozzák, hogy kapcsolódjunk a gépünkről. Az ilyen problémákat már kellett észlelnünk az *smbclient* tesztek során (lásd *„A tallózás tesztelése az smbclient programmal”*).
- Ha a „The specified computer is not receiving requests” üzenetet kapjuk, akkor rosszul írtuk be a nevet, vagy a gép broadcast üzenetek számára nem érthető el (lásd *„A hálózat tesztelése az nmblookup programmal”*), vagy nem fut rajta az *nmbd* démon.
- Ha a „Bad password error” üzenetet kapjuk, akkor feltehetően a Microsoft-titkosítású jelszavakkal lehet probléma (ezekről és a javításaikról a 6. fejezetben volt szó).

A kiszolgáló tallózása az ügyfél oldaláról

A Windows Intézőjében próbáljuk meg csatlakoztatni a hálózati meghajtót (Eszközök menü), és tallózzuk a kiszolgálót. A Samba kiszolgálónak a helyi munkacsoport tallózólistájában kell megjelennie. Kettőt kattintva a kiszolgáló nevére, meg kell kapnunk a rajta elérhető megosztások listáját, amint a 9.5. ábrán látható.



9.5. ábra. A kiszolgálón elérhető megosztások listája

- Ha az „Invalid password” üzenetet kapjuk, és a gépen az NT 4.0, az NT 3.5 a 3-as szervizcsomaggal, a Windows 95 a 3-as szervizcsomaggal vagy a Windows 98 valamelyike fut, továbbá az Internet Explorer 4.0 tallózót tartalmazza a rendszer, akkor nagy valószínűséggel ismét titkosítási problémával kell szembenéznünk. Az összes ilyen ügyfél alapbeállítás szerint Microsoft titkosítású jelszavakat használ (lásd a 6. fejezetet).

- Ha az „Unable to browse the network” üzenetet kapjuk, akkor az alábbi hibák valamelyik léphetett fel:
  - Túlságosan hamar befejeztük a keresést, mielőtt még lezárultak volna az üzenetküldések és a frissítések; várjunk 30 másodpercig, majd próbálkozzunk újra.
  - Eddig még nem diagnosztizált hálózati probléma merült föl.
  - Nincs főtallózó. Vegyük fel az *smb.conf* fájlba a `local master = yes` beállítást.
  - Az *smb.conf* fájlban egyik megosztás sincs tallózhatóként specifikálva.
- Ha a „\\server is not accessible” üzenetet kapjuk, akkor az alábbiak lehetségesek:
  - Problémák vannak a jelszó titkosításával.
  - A gép valóban nem érhető el.
  - A gép nem támogatja a tallózást.

Egyéb hibalehetőségek

Ha már eddig eljutottunk, akkor vagy sikerült megoldani a problémát, vagy olyan hibával kerültünk szembe, amellyel eddig még nem találkoztunk. A most következő fejezetrészekben olyan hibakeresési eljárásokról lesz szó, amelyek nem magát a Sambát, hanem a Samba futtatásához szükséges infrastruktúrát igénylik.

Elmaradt bejelentkezések

Esetenként problémát okozhat, ha elfelejtünk bejelentkezni az ügyfélgépre, vagy nem a megfelelő személyként jelentkezünk be (fiók nélkül). Az előzőt egyáltalán nem lehet diagnosztizálni: a Windows nagyon barátságos, és mindenkit beenged. Helyileg! Az utóbbi esetben is csak annyi történik, hogy a Windows a köszöntője után bekér egy új fiókot. Akár az egyik, akár a másik oda vezethet, hogy ismétlődően elutasításra kerül a kapcsolatfelvétel, és vég nélkül ismétlődik a jelszó bekérése. Ha semmi más nem segít, akkor indítsuk újra a rendszert, és próbáljunk meg újra bejelentkezni.

Hibakeresés a névszolgáltatásban

Ebben a fejezetrészben rendre megvizsgáljuk azokat a névszolgáltatásokat, amelyekkel a munkánk során találkozhatunk, de csak a Sambával kapcsolatos hibákra térünk ki.

Számos jó könyv foglalkozik a különböző névszolgáltatásokkal kapcsolatos hibakereséssel: Paul Albitz és Cricket Liu szerzők *DNS and Bind* című könyve a DNS (Domain Name Service) szolgáltatást, Hal Stern *NFS and NIS* című könyve a NIS (Yellow Pages) szolgáltatást ismerteti (mindkettőnek az O'Reilly a kiadója), míg a WINS (Windows Internet Name Service) szolgáltatást, a hosts/LMHOSTS fájlokat és a NIS+ szolgáltatást elsősorban a szerzők kézikönyvében lehet tanulmányozni.

A fejezetnek ebben a részében az alábbi problémákat vizsgáljuk:
- a névszolgáltatás azonosítása;
- nem kereshető egy gazdanév;
- a gazdanév hosszú alakja (FDQN) működik, míg a rövid alakja nem;
- a név rövid alakja működik, a hosszú alakja viszont nem;
- hosszú a késleltetés a várt eredményt megelőzően.

A használt szolgáltatás azonosítása

Lássuk először, hogy mind a kiszolgáló, mind az ügyfél DNS, WINS, NIS vagy *hosts* fájlokat használ-e a megadott névhez tartozó IP cím kereséséhez. A különböző operációs rendszereknek eltérő a preferenciájuk:
- A Windows 95 és 98 rendszerű gépek először a WINS szolgáltatást, majd az *LMHOSTS* fájlokat, utána a broadcast, majd végül a DNS szolgáltatást és a *hosts* fájlokat használják.
- Az NT gépek a következő sorrendben próbálkoznak a különböző névszolgáltatásokkal: WINS, broadcast, *LMHOSTS* fájlok, *hosts* fájlok és DNS.
- A WINSOCK szabványt (mint például PC-NFS) használó Windows programok kedvezményezettségi sorrendje: *hosts* fájlok, DNS, WINS, majd broadcast. Ne tételezzük fel, hogy ha valamelyik program névkiszolgálója működik, akkor az SMB ügyfélprogram névkiszolgálója is működni fog!
- A Samba démonjai a következő sorrendben keresik a névkiszolgálókat: *LMHOSTS*, WINS, a Unixban beállított preferencia, majd broadcast.
- A Unix gazdagépei a DNS, *hosts* fájlok, NIS és NIS+ bármilyen kombinációjához, és általában bármilyen sorrendjéhez konfigurálhatók.

Ajánlható, hogy az ügyfélgépek és a Samba démonjai a WINS és a DNS szolgáltatást használják, míg a Unix kiszolgáló a DNS-t vegye igénybe. A jegyzeteinkben vagy a gépeinken megnézhetjük, hogy az illető gép éppen melyik szolgáltatáshoz van beállítva.

Az ügyfélgépeken a névszolgáltatókat a Hálózat párbeszédablakból kiindulva, a TCP/IP tulajdonságai párbeszédlapon állíthatjuk be, amint ezt a 3. fejezetben láttuk. Ellenőrizzük ezen a párbeszédlapon, hogy éppen melyik névszolgáltatás van bekapcsolva. A kiszolgálónál nézzük meg, hogy létezik-e az */etc/resolv.conf* fájl. Ha létezik, akkor az a DNS névszolgáltatást használja. Természetesen más névszolgáltatásokat is igénybe vehetünk. Szükség lehet a NIS és más szolgáltatások kombinációjának a vizsgálatára is.

Solaris és más System V Unix operációs rendszerekben keressünk egy */etc/nsswitch.conf* fájlt. Ha találunk ilyet, akkor keressünk benne olyan sort, amely a `host:` szóval kezdődik, és utána egy vagy több `files`, `bind`, `nis` vagy `nis+` bejegyzés áll. Ezek jelentik ebben a sorrendben a használatban lévő névszolgáltatásokat: a files a *hosts* fájlokat, míg a bind (a Berkeley Internet Name Daemon) a DNS-t jelenti.

Ha más az ügyfél és más a kiszolgáló, akkor első teendő ezek szinkronizálása. Az ügyfelek névkiszolgálóként csak a DNS-t, a WINS-t, a *hosts* és az *lmhosts* fájlokat használhatják, a NIS és a NIS+ szolgáltatást nem. Ezzel szemben a kiszolgálók a *hosts* fájlokat, a DNS, a NIS és a NIS+ szolgáltatásokat vehetik igénybe, a WINS-t viszont nem, még ha a Samba kiszolgáló fel is kínálja ezt. Ha nem tudjuk elérni, hogy az összes rendszer ugyanazokat a névszolgáltatásokat használja, akkor gondosan ellenőriznünk kell, hogy mind a kiszolgáló, mind az ügyfelek azonos adatokat használjanak.

A Samba 2.0 verziója (és a késői 1.9-es verziók) lehetővé teszi az `-R` kapcsoló (névfeloldási sorrend) használatát az *smbclient* program paramétereként. Ha hibát akarunk keresni a WINS névkiszolgálóban, akkor például az alábbi alakú parancsot adhatjuk ki:

```
smbclient -L server -R wins
```

A kapcsolóhoz a következő paramétereket adhatjuk meg: `hosts` (ami mindazokat a szolgáltatásokat jelenti, amelyeket a Unix gép használ, és nem csak az */etc/hosts* fájlokban felsoroltakat), `lmhosts`, `wins` és `bcast` (broadcast).

A most következő fejezetrészekben a *hosszú név* a teljes tartománynevet (FQDN, fully qualified domain name), mint például `server.example.com`, míg a *rövid név* a teljes tartománynévnek csak a gazda részét jelenti, mint például `server`.

Nem kereshetők a gazdanevek

Próbáljuk a következőket:

- DNS névkiszolgálóban:
  Futtassuk az `nslookup` *`név`* parancsot. Ha ez nem sikerül, akkor keressünk egy *resolv.conf* hibát, egy leállított DNS kiszolgálót vagy valamilyen, a rövid/hosszú nevekkel kapcsolatos problémát (lásd a következő szakaszt). Próbáljuk a következőt:
  - Az */etc/resolv.conf* fájlnak tartalmaznia kell egy vagy több névkiszolgáló sort, amelyek mindegyikében egy IP címnek kell lennie. Ezek a DNS kiszolgálók címei.
  - A megtalált kiszolgálócímekre adjuk ki a ping parancsot. Ha valamelyik címre nem működne, akkor az a gyanúsított gép. Ha egyik címre sem működnek, akkor a hálózatra gyanakodhatunk.
  - Ismételjük meg a keresést teljes tartománynévvel (vagyis például *server.example.com*), ha előzőleg a rövid névvel kerestünk, illetve fordítva, a rövid névvel, ha előzőleg a hosszú névvel kerestünk. Ha a két keresés eredménye eltér egymástól, akkor térjünk rá a következő fejezetrészre.
- Broadcast/WINS névkiszolgálóban:
  A broadcast/WINS névkiszolgáló csak rövid neveket használ, mint például `server` (például a hosszú `server.example.com` helyett). Futtassuk az `nmblookup -S` *`server`* parancsot. A parancs mindazt jelenti, amit a broadcast regisztrál az illető névhez. A példánkban a parancs a következőket jelenti vissza:

```
Looking up status of 192.168.220.100
received 10 names
        SERVER          <00> -         M <ACTIVE>
        SERVER          <03> -         M <ACTIVE>
        SERVER          <1f> -         M <ACTIVE>
        SERVER          <20> -         M <ACTIVE>
        .._ _MSBROWSE_ _.<01> - <GROUP> M <ACTIVE>
        MYGROUP         <00> - <GROUP> M <ACTIVE>
        MYGROUP         <1b> -         M <ACTIVE>
        MYGROUP         <1c> - <GROUP> M <ACTIVE>
        MYGROUP         <1d> -         M <ACTIVE>
        MYGROUP         <1e> - <GROUP> M <ACTIVE>
```

Az igényelt bejegyzés a SERVER <00>, ami a *server*-t úgy azonosítja, mint a gép NetBIOS nevét. A munkacsoport nevének is elő kell fordulnia egyszer vagy többször a kimenetben. Ha hiányoznának ezek a sorok, a broadcast/WINS nem tudná megkeresni a neveket, és a probléma további vizsgálatokat igényel.

Az előbbi listában a csúcsos zárójelek között álló számok a NetBIOS neveket munkacsoportokként, munkaállomásokként, az üzenetszolgáltató fájlhasználóiként, főtallózókként, tartomány-főtallózókként, tartományvezérlőkként és még számos egyéb szereplőként azonosítják. Közülük a <00> elsősorban ügyfélgép- és munkacsoport-neveket, a <20> pedig kiszolgálóneveket azonosít. A teljes lista a *http://support.microsoft.com/support/kb/articles/q163/4/09.asp* internetes címen érhető el.

- NIS rendszerben:
  Próbálkozzunk az `ypmatch` *`gazdanév`* `hosts` paranccsal. Ha nem fut le rendben, akkor nem működik a NIS. Az *ypwich* parancsot futtatva keressük meg a NIS kiszolgáló nevét, és a *ping* paranccsal vizsgáljuk meg, hogy elérhető-e.
- NIS+ rendszerben:
  Próbálkozzunk az `nismatch` *`gazdanév`* `hosts` paranccsal. Ha nem fut le rendben, akkor nem működik a NIS. A *niswich* parancsot futtatva keressük meg a NIS kiszolgáló nevét, és a *ping* paranccsal vizsgáljuk meg, hogy elérhető-e.
- *hosts* fájlokban:
  Vizsgáljuk meg az */etc/hosts* fájlokat az ügyfélnél (`C:\WINDOWS\HOSTS`). Mindegyik sorban egy IP-számnak és egy vagy több névnek kell állnia, melyek közül az első a valódi, a többi az álnév. Például:

```
127.0.0.1          localhost
192.168.236.1      dns.svc.example.com
192.168.200.105     client.example.com client
192.168.236.11     backup.example.com loghost
192.168.220.100     server.example.com server
192.168.236.254    router.svc.example.com
```

  Unixban a `localhost` névhez mindig a 127.0.0.1 címnek kell tartoznia, bár egy PC-n ez a név egy gazdagép álneve is lehet. Az ügyfélnél ellenőrizzük, hogy ne legyenek `#XXX` direktívák a sorok végén; ezek LAN Manager/NetBIOS direktívák, és csak az *LMHOSTS* fájlokban állhatnak (`C:\WINDOWS\LMHOSTS`).
- *LMHOSTS* fájlokban:
  Ez a fájl a LAN Manager (NetBIOS) nevek helyi forrása. A bejegyzéseinek hasonló a formátuma, mint az */etc/hosts* fájlok bejegyzéseié, de nem használhatók benne hosszú tartománynevek (mint például `server.example.com`), és a nevek után `#XXX` direktívák állhatnak. Jegyezzük meg, hogy (mintaként) van egy *lmhosts.sam* fájl a `C:\WINDOWS` könyvtárban, de ez mindaddig nem kerül feldolgozásra, amíg nem másoljuk át a `C:\WINDOWS\LMHOSTS` fájlba.

Hosszú és rövid gazdanevek

Ha egy gazdanév hosszú alakja használható, viszont a rövid alakja nem (például a `client.example.com` működik, míg a `client` nem), gondoljuk át a következőket.

- DNS:
  Ez általában azt jelenti, hogy nincs olyan alapértelmezett tartomány, amelyben kereshetők a rövid nevek. Keressünk a Samba kiszolgálón az */etc/resolv.conf* fájlban olyan `default` sort, amelyben szerepel a tartományunk, vagy keressünk olyan `search` sort, amelyben egy vagy több tartomány szerepel. Valamelyikben kell lennie olyannak, amelyik lehetővé teszi a rövid nevek használatát – az, hogy melyikben, a DNS feloldó készítőjétől és verziójától függ. Vegyünk fel egy `domain` *`saját_tartomány`* sort a *resolv.conf* fájlba, és kérdezzük meg a hálózati vagy a DNS rendszergazdát, hogy mit kell tartalmaznia a fájlnak.
- Broadcast/WINS:
  A boradcast/WINS nem támogatja a hosszú neveket, ezért itt ilyen probléma nem merülhet fel.
- NIS:
  Próbálkozzunk az `ypmatch` *`gazdanév`* `hosts` paranccsal. Ha nem találunk egyezőséget, akkor a táblázatok nem tartalmaznak rövid neveket. Kérdezzük meg a hálózat rendszergazdájától, hogy csak véletlenül hiányoznak a rövid nevek, vagy a kialakított házirend miatt maradtak ki. Egyes helyeken sohasem használnak (félreérthető) rövid neveket.
- NIS+:
  Próbálkozzunk a `nismatch` *`gazdanév`* `hosts` paranccsal, és hiba esetén járjunk el úgy, mint a NIS-nél.
- *hosts* fájlok:
  Ha a rövid név nem szerepel az */etc/hosts* fájlokban, akkor próbálkozzunk azzal, hogy álnévként vesszük fel. Lehetőség szerint ne használjuk elsődleges névként (a sorban elsőként) rövid neveket. Ha a rendszer engedi, csak álnévként használjuk ezeket.
- LMHOSTS:
  A LAN Manager nem támogatja a hosszú neveket, úgyhogy itt ilyen problémák nem merülhetnek fel.

Ha viszont a rövid nevek működnek, a hosszúak pedig nem, akkor vizsgáljuk a következőket:

- DNS:
  Ez nagyon furcsa eset – kérdezzük meg a hálózat vagy a DNS rendszergazdáját. Valószínűleg a DNS telepítésével van probléma.
- Broadcast/WINS:
  Ebben nincs semmi különös; a broadcast/WINS nem képes a hosszú nevek használatára. Próbálkozzunk a DNS névkiszolgálóval. A Microsoft azt állítja, hogy át lehet kapcsolni erre, bár a DNS nem szolgáltat olyan névtípusokat, mint a <00>.
- NIS:
  Ha az `ypmatch` paranccsal a rövid alak megtalálható, viszont a hosszú nem, akkor vegyük fel a hosszú alakot a táblázatba, legalább álnévként.
- NIS+:
  Ugyanaz, mint a NIS-nél azzal a különbséggel, hogy az `ypmatch` parancs helyett a `nismatch` paranccsal keressük a neveket.

- *hosts* fájlok:
  Vegyük fel a hosszú nevet legalább álnévként, és lehetőség szerint az elsődleges alakban. Ne zárjuk ki a DNS használatát sem.
- LMHOSTS:
  Ez érthető hiba. A LAN Manager nem képes a hosszú nevek használatára. Vegyük fontolóra a DNS-re vagy a *hosts* fájlokra való átkapcsolást.

Szokatlan késések

Ha túlságosan hosszú idő telik el a várt eredmények megérkezéséig:

- DNS:
  Teszteljük ugyanazt a nevet az *nslookup* paranccsal a lassú gépen (ügyfél vagy kiszolgáló). Ha ez a parancs is lassú lenne, akkor a DNS-sel van probléma. Ha az ügyfélnél lassú, akkor túl sok protokoll van hozzákötve az Ethernet kártyához. Szakítsuk meg a NetBEUI kötését, ami híresen lassú és a Novell kötését is, abból kiindulva, hogy nincs szükség rá. Ez különösen fontos a Windows 95-ben, amelyet nagyon lelassítanak a fölösleges protokollok.
- Broadcast/WINS:
  Teszteljük az ügyfelet az `nmblookup` paranccsal, és ha gyorsabb, akkor valószínűleg az előbb leírt protokollproblémákkal van dolgunk.
- NIS:
  Próbálkozzunk az `ypmatch` paranccsal, és jelezzük a problémát a hálózat rendszergazdájának.
- NIS+:
  Ugyanaz, mint fent, csak a `nismatch` paranccsal próbálkozzunk.
- *hosts* fájlok:
  A *hosts* fájlok, ha ésszerű határokon belül marad a méretük, mindig gyorsak. Lehetséges, hogy a DNS-nél leírt protokollproblémák okozzák a késedelmet.
- *LMHOSTS:*
  A késedelemnek nem lehet névkeresési probléma az oka, az *LMSOSTS* fájlok ugyanolyan gyorsak, mint a *hosts* fájlok.

Localhost problémák

Ha a localhost gazdának nem 127.0.0.1 a címe, akkor próbáljuk a következőket:

- DNS:
  Lehetséges, hogy nem létezik a `localhost. A 127.0.0.1` bejegyzés. Vegyük fel ezt és a fordítottját is: `1.0.0.127.IN-ADDR.ARPA PTR 127.0.0.1.`
- Broadcast/WINS:
  Nem értelmezhető.
- NIS:
  Ha `localhost` nem szerepel a táblázatban, akkor vegyük fel bele.
- NIS+:
  Ha `localhost` nem szerepel a táblázatban, akkor vegyük fel bele.
- * *host* fájlok:
  Vegyük fel a `127.0.0.1 localhost` bejegyzést.
- *LMHOSTS:*
  Nem értelmezhető.

Hibakeresés hálózati címekben

Számos problémát okoznak a hibásan továbbküldött internet címek vagy a címek hibás kiosztása. Ebben a fejezetrészben az ilyen problémákról lesz szó.

Hálózati maszkok

A hálózati maszkok (más néven alhálózati maszkok) azt mondják meg az egyes gépeknek, hogy melyek azok a címek, amelyek közvetlenül érhetők el róla (vagy a hozzá kapcsolódó helyi hálózatból), és melyek azok, amelyek csak közvetve, forgalomirányítókon (router) keresztül érhetők el. Ha rosszul van megadva a hálózati maszk, akkor kétféle hiba fordulhat elő. Az egyik az, hogy a helyi csomagok is rákerülnek a forgalomirányítóra, ami meglehetősen drága módja az időpocsékolásnak – a csomag ugyan lehet, hogy gyorsan célba ér, de az is lehet, hogy csak nagyon lassan, netán el is veszik útközben. A másik hiba az lehet, hogy egy távoli gépnek szánt csomag nem kerül rá a forgalomirányítóra, és ezért nem is jut el a célba.

A hálózati maszknak hasonló a formátuma, mint egy IP címnek – a cím hálózati részében a bitek értéke 1, a gazda részében a bitek értéke 0. A hálózati maszknak szó szerint az a szerepe, hogy a TCP/IP kódon belül letakarja a cím egy részét. Ha a maszk címe 255.255.0.0, akkor az első két bájt a cím hálózati része, a második két bájt pedig a gazda része. Leggyakoribbak a 255.255.255.0 alakú hálózati maszkok, amelyben az első három bájt a hálózati rész, az utolsó bájt pedig a gazda rész.

Tegyük fel például, hogy az IP címünk 192.168.0.10, a Samba kiszolgálóé pedig 192.168.220.100. Ha a hálózati maszkunk címe történetesen 255.255.255.0, akkor a cím első három bájtja a hálózati rész, és a negyedik bájt a gazda rész. Ebben az esetben a hálózati részek különbözőek, vagyis a két gép különböző hálózathoz tartozik:

Hálózati rész	**Gazda rész**
192 168 000	10
192 168 235	86

Ha viszont az alhálózat címe 255.255.0.0, akkor a hálózati rész csak az első két bájt. Ebben az esetben a két gép hálózati része azonos, vagyis a két gép azonos hálózathoz tartozik:

Hálózati rész	**Gazda rész**
192 168	000 10
192 168	236 86

Természetesen ha a hálózati maszk címét nem úgy adjuk meg, ahogyan azt a rendszergazda beállította, akkor hibás a hálózati maszkunk.

Broadcast címek

A broadcast (üzenetszórási) cím ugyanolyan cím, mint a többi, csak a cím gazda részének összes bitje 1-re van állítva. Ez szavakkal elmondva azt jelenti, hogy „az adott hálózaton belül az összes gép”. Az IP cím és a hálózati maszk címe alapján egyszerűen meghatároz-

hatjuk a broadcast címet: az IP cím azon részében, amely a hálózati maszk gazda része (vagyis ahol a 0-k állnak), állítsuk az összes bitet 1-re. A műveletet az alábbi táblázat szemlélteti:

	Hálózati rész	**Gazda rész**
IP cím	192 168 236	86
Hálózati maszk	255 255 255	000
Broadcast cím	192 168 236	255

A fenti példában a 192.168.236 hálózaton a broadcast cím 192.168.236.255. Létezik még egy „univerzális" broadcast cím is, a 255.255.255.255. A forgalomirányítók számára tiltott az ilyen címekre való továbbítás, de a helyi hálózatok gépei általában válaszolnak az ilyen címre szétküldött üzenetekre.

Hálózati címtartományok

Számos címtartomány le van foglalva tesztelési és nem kapcsolódó hálózatok számára – a könyvben a szerzők is ilyeneket használtak. Ha az Olvasónak még nem lenne hálózati címe, akkor nyugodtan válasszon magának ilyen címtartományba eső címet. Szabadon használhatók az A osztályú 10.*.*.* címek és a C osztályú, 192.168.1.* és 192.168.254.* közötti címtartományba eső címek. A könyvbeli példák az utóbbi tartományba eső 192.168.236.* címeket használták. Az *example.com* tartomány ugyancsak foglalt a nem kapcsolódó hálózatok, magyarázó példák és könyvek számára.

Természetesen ha kapcsolódni akarunk az internetre, akkor valós hálózati címre és tartománynévre van szükségünk, amit feltehetően az internetszolgáltatást nyújtó cégtől kaphatunk meg.

Hálózati cím keresése

Ha még nem jegyeztük volna fel az IP címünket, akkor Unix rendszerben az *ifconfig*, Windows 95 és NT rendszerben az IPCONFIG parancs kiadásával jeleníthetjük meg. (Az operációs rendszer kézikönyvében nézzünk utána, milyen kapcsolót kell esetleg megadnunk a parancssorban (a Sun például az `ifconfig -a` alakú parancsot várja). A parancs az alábbihoz hasonló kimenetet állíthatja elő:

```
server% ifconfig -a
le0: flags=63<UP,BROADCAST,NOTRAILERS,RUNNING >
      inet 192.168.236.11 netmask ffffff00 broadcast 192.168.236.255
lo0: flags=49<&lt>UP,LOOPBACK,RUNNING<&gt>
      inet 127.0.0.1 netmask ff000000
```

Az interfészek egyike a visszacsatolt interfész (a példában ez az `lo0`), a másik pedig a normál IP-interfész. A flags= bejegyzések azt jelzik, hogy fut az interfész, és az Ethernet támogatja a broadcast eljárást (a PPP nem támogatja). IP címeket még a következő helyeken is kereshetünk: */etc/hosts* fájlok, Windows *HOSTS* fájlok, Windows *LMHOSTS* fájlok, NIS, NIS+ és DNS.

Hibakeresés a NetBIOS nevekben

Történelmileg az SMB protokolloknak a NetBIOS névrendszer, más néven a LAN Manager névrendszer az alapja. Ez egy egyszerű séma volt, amely szerint minden egyes gépnek egyedi, 20 karakteres neve van, amelyet broadcast eljárással szétszór a hálózatban, hogy minden résztvevő tudomást szerezzen róla. A TCP/IP protokoll létrejöttével azonban már olyan neveket használnunk, mint például a *client.example.com*, amelyeket DNS vagy WINS névkiszolgálókon keresztül */etc/hosts* fájlok tárolnak.

A tartománynevekre, mint például a *server.example.com* névre történő szokásos leképezés a névnek csak a *server* részét használja NetBIOS névként, és ezt alakítja át nagybetűssé. Ez azonban nem mindig használható, főként akkor nem, ha a gépnek 21 karakterből áll a neve, továbbá nem mindenki használja ugyanazt a nevet NetBIOS és DNS névként. Nem szokatlan például a *corpvm1* és a *vm1.corp.com* nevek együttes használata.

Hibakeresésnél zavaró lehet, ha egy gépnek más a NetBIOS, és más a tartományneve, ezért javasolható, hogy ahol lehet, kerüljük ezt a névmegadást. A NetBIOS nevek az *smbclient* programmal nem deríthetők fel.

- Ha a Samba kiszolgálón az *smbclient* paranccsal és az `-L` kapcsolóval ki tudjuk listázni a *`rövid_név_kiszolgáló`* megosztásait, akkor a *`rövid_név`* a NetBIOS név.
- Ha a „Get_Hostbyname: Unknown host name" üzenetet kapjuk, akkor valamilyen névilleszkedési probléma van. Nézzük meg az *smb.conf* fájlban, hogy be van-e külön állítva a NetBIOS név.
- Próbálkozzunk újra a paranccsal úgy, hogy paraméterként az `-I` kapcsolót és a Samba kiszolgáló IP címét adjuk meg (vagyis `smbclient -L server -I 192.168.220.100`). Ezzel megkerüljük a név keresését, és kikényszerítjük, hogy a csomagok közvetlenül a IP címre kerüljenek. Ha ez működik, akkor biztosan névillesztéssel kapcsolatos a probléma.
- Próbálkozzunk az `-I` kapcsolóval és a kiszolgáló teljes tartománynevével (vagyis `smbclient -L server -I server.example.com`). Ez teszteli a tartománynevet, akármilyen sémát is használ a Samba kiszolgáló (például a DNS-t). Ha nem fut le rendben a parancs, akkor névszolgáltatási problémával lehet dolgunk. Miután elvégeztük a NetBIOS nevekkel kapcsolatos hibakeresést, térjünk vissza a *„Hibakeresés a névszolgáltatásban"* című fejezetrészhez.
- Próbálkozzunk az `-n` kapcsolóval (NetBIOS név) és azzal a névvel, amelyről feltételezzük, hogy létezik (például `smbclient -n server -L server-12`), de az `-I` kapcsolóval ezúttal ne lépjük át az IP címet. Ha a parancs rendben lefut, akkor az `-n` kapcsoló után megadott név a kiszolgáló aktuális NetBIOS neve. Ha a „Get-Hostbyname: Unknown host MARY" üzenetet kapjuk, akkor még nem találtuk meg a megfelelő kiszolgálót.
- Ha az eddigi próbálkozásaink egyike sem sikerült, akkor ismételjük meg a teszteket úgy, hogy a `U` *`felhasználónév`* és a `-W` *`munkacsoport`* paramétereket használjuk, ahol a felhasználónév és a munkacsoport csupa nagybetűből áll. Ezzel biztosíthatjuk, hogy ne zavarják meg a tesztet a kis- és nagybetűket vegyesen használó felhasználói vagy munkacsoport-nevek.
- Ha eddig még semmi sem sikerült, és biztosak vagyunk abban, hogy névszolgáltatással kapcsolatos a probléma, akkor keressük a hibát a *„Hibakeresés a névszolgáltatásban"* című fejezetrészben leírtak szerint, majd térjünk vissza a NetBIOS nevekhez.

Különleges erőforrások

A Sambával végzett munkánk során előfordulhat, hogy ki akarunk tekinteni a nagyvilágba, keresni akarjuk az újdonságokat és a frissítéseket, vagy egyéb segítség után szeretnénk nézni.

Dokumentációk és a gyakori kérdések

Tekintsük természetesnek a dokumentáció elolvasását – ez annyira fontos, hogy aligha kell hangsúlyozni. A Samba hatalmas méretű dokumentációs fájlokat tartalmaz, és legalább annyit tegyünk meg, hogy tallózgatunk benne – akár a */docs* fájlokban tallózva, akár a Samba *http://samba.anu.edu.au/samba/* weboldalán navigálva. Ezen a webhelyen megnézhetjük a gyakran ismételt kérdések (FAQ) legfrissebb listáját, a programban megtalált hibákat vagy a disztribúciós helyeket, továbbá hivatkozásokat találunk a Samba kézikönyveire és a speciális tudnivalókra (HOW-TOs).

Samba hírcsoportok

A Usenet hírcsoportok mindig is kiváló helynek bizonyultak, ahol bármely témában találhatunk jó tanácsokat. Az elmúlt években az itt felgyülemlett ismeretanyag felbecsülhetetlen erőforrássá vált. A különböző archiváló és kereső webhelyeknek – mint amilyen például a DejaNews (*http://www.dejanews.com*) – köszönhetően mindössze néhány egérkattintásra van szükség ahhoz, hogy értékes megoldásokat kapjunk egy-egy témával kapcsolatban.

A Samba elsődleges hírcsoportjának *comp.protocols.smb* a címe. Ha bármilyen problémánk van, ez legyen az elsőként felkeresendő hely. Akár hiszi valaki, akár nem, néha öt perc is elegendő, hogy megoldást kapjunk itt egy olyan problémára, amivel magunk esetleg órákon keresztül sem boldogulnánk.

A hírcsoportokban való kereséskor amennyire csak lehet, határoljuk be a problémát, de ne legyünk túlságosan szűkszavúak. A legjobb, ha a kapott hibaüzenet szerint végezzük a keresést. Ha a hírcsoportokban nem találunk azonnal választ a kérdésünkre, ne kérjünk azonnal segítséget, hanem gondolkozzunk el egy kicsit a problémán. Lehet, hogy a gyakran ismételt kérdések (FAQ) között vagy a Samba hatalmas méretű dokumentációs fájljaiban megtaláljuk a választ, netán a Samba valamelyik diagnosztikai eszközét használva magunk is rájövünk a megoldásra. Ha sehogyan sem boldogulunk, küldjünk egy kérést a *comp.protocols.smb* címre, és amennyire csak tudjuk, írjuk le pontosan, mikkel próbálkoztunk, és milyen eredményre jutottunk. Mellékeljük a kapott hibaüzeneteket is. Lehet, hogy napokba is beletelik, mire választ kapunk, de legyünk türelemmel, és időközben is próbálkozzunk más megoldásokkal.

Megismételjük: a segítségkérést követően is keressük magunk a megoldást. A könyv szerzői és az ismerősei folyamatosan tapasztalják, hogy miután elküldött valaki egy mindenféle rejtélyes részletekre kiterjedő, több száz sorból álló Usenet cikket, amely a föld számos kontinensét bejárta, a cikk küldője rövid egy órán belül maga is rájött a megoldásra. Erősen leegyszerűsítve kimondható: minél több emberhez jut el egy kérdés, annál egyszerűbb a megoldás. Kissé eltúlozva: ha a Unix közösség minden egyes tagját szerencséltettük már a kérdésünkkel, akkor a tanács talán következő lehet: „Kedves Barátunk, ugyan légy oly szíves, és dugd be a számítógéped villásdugóját a fali konnektorba!"

A Samba levelezőlistái

Az alábbiakban felsorolt levelezőlisták kapcsolatosak a Sambával. További információk, valamint a levelezőlistákra való be-, illetve kijelentkezéssel kapcsolatos tudnivalók a Samba *http://www.samba.org/* internetes címén találhatók.

samba-binaries@samba.org
: Ez a levelezőlista a Samba előre lefordított bináris fájljaival kapcsolatos tudnivalókat tartalmazza.

samba-bugs@samba.org
: Ez a levelezőlista a Sambában gyanított hibákat sorolja fel.

samba-ntdom@samba.org
: Ez a levelezőlista a Sambában használható tartományok (főként Windows NT) támogatásával kapcsolatos információkat tartalmazza.

samba-technical@samba.org
: Ez a levelezőlista a Samba jövőjével kapcsolatos vitafórum.

samba@samba.org
: Ez a Samba elsődleges levelezőlistája, amelyen általános kérdések és ún. HOW-TO (Mit tegyünk, hogy…) útmutatások olvashatók a Sambával kapcsolatban.

http://kt.linuxcare.com/KC/samba/
: Ez a levelezőlista szerkesztett megjegyzéseket tartalmaz a Samba fejlődéséről, hasonlóképpen, mint a Linux „Kernel Traffic" webhelye. A webhelyet olvasva folyamatosan figyelemmel kísérhetjük a Sambán végzett aktuális változtatásokat.

http://lists.samba.org/mailman/
: A Samba hivatalos levelezőlistáira mutató hivatkozások gyűjtőhelye.

Szakkönyvek

Brian L. Wong: *Configuration and Capacity Planning for Solaris Servers;* Sun/Prentice-Hall, Englewood Cliffs 1997.

Raj Jain: *The Art of Computer Systems Performance Analysis;* John Wiley and Sons, New York 1991.