Linuxos kiszolgálót mindenkinek! (7. rész)

A SuSE Linux mint kiszolgáló – kisvállalati és otthoni környezetben.

Cikksorozatunk előző részeiben népszerű internetes és hálózati szolgáltatások kiszolgálását tettük lehetővé rendszerünkön. Most az egyik leggyakrabban, bár kétség kívül nem a leglátványosabban használt szolgáltatást fogjuk megismerni, nevezetesen a DNS-t, azaz a tartománynév szolgáltatást (DNS – Domain Name Service).

Az elméletről dióhéjban

Mint azt bizonyára mindenki tudja, az internetre kapcsolt gépek mindegyike rendelkezik egy IP-címmel, amelyen keresztül az adott gép megszólítható bármelyik másik hálózatba kapcsolt gépről. Mivel az IP-címek 32 bites számok, ezért e számokból akár csak a kedvenc weboldalaink címét megjegyezni is felér egy fél telefonkönyv megtanulásával, nem beszélve az egyéb felmerülő nehézségekről.

Az előbbiekből kiindulva már a TCP/IP-hálózatok megjelenésekor felmerült az igény, hogy az egyes gépeket ne csak IP-cím szerint, hanem egy választott név alapján is meg lehessen találni. Ehhez egy olyan adatbázis szükséges, amelyben név és cím párok szerepelnek.

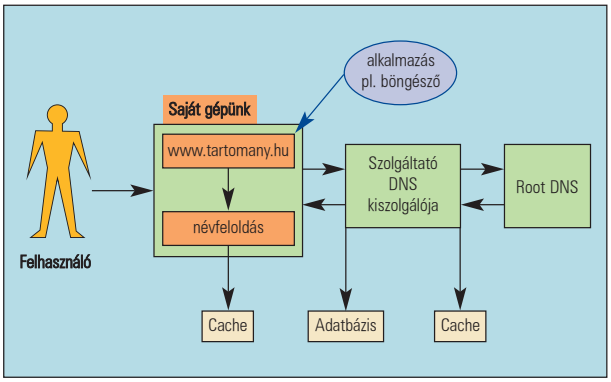
A Unix-rendszerekben kezdetben ez az adatbázis a ***hosts.txt*** állomány volt. Ebbe az állományba gyűjtötték a név és cím párokat, és ha egy gépre a neve alapján hivatkoztak, akkor a rendszer ebből az állományból kereste ki a névhez tartozó IP-címet. Ezt az adatbázist a Stanford Research Institute-nál (SRI) működő Network Information Center („the NIC") tartotta karban. Amikor új gép jelent meg a hálózatban, a választott névvel bejegyezték a ***hosts.txt*** állományba, majd hetente egyszer vagy kétszer közzétették a legújabb frissítést, hogy az új név–címpárokat elérhetővé tegyék.

Ez a rendszer tökéletesen működött néhány tucat vagy pár száz név tárolására, ellenben a jelenlegi címigények kielégítésére használhatatlan.

További gondot jelentett, hogy a ***hosts.txt*** adatbázisába bárki megkötés nélkül helyezhetett el bejegyzéseket, így megvolt annak a lehetősége, hogy kellő szabályozás híján egymással ütköző neveket helyezzenek el az állományban, ezzel téve használhatatlanná.

A ***hosts.txt*** utóda a mai napig megtalálható mind a Unix- és Linux-, mind a Windows-rendszerekben. Előbbieknél a */etc/hosts* állományt kell keresni, utóbbinál a *%SystemRoot%/system32/drivers/etc/hosts* állományt.

Paul Mockapetris 1983-ban megalkotta a Domain Name System (DNS) nevű név és cím párosítási rendszert. Az ötlet egy hierarchikus rendszeren alapul, amelyben a tartományneveket faszerkezetbe rendezetten szeretnénk tárolni. A faszerkezetbe rendezés több előnnyel is bír: jól átlátható az általa leírt szervezeti szerkezet, valamint megfelelő faszerkezet esetében a keresés nagyságrendekkel hatékonyabb, mint a hagyományos listában való keresés.



A fa gyökere az úgynevezett fő gyökér (root). Ezt az eredeti DNS-rendszerben egy ponttal jelölték. A gyökér alatt található első szint az úgynevezett legfelső szint (top level). Ez alatt pedig a másodszintű tartománynevek (second level) találhatóak.

A Top Level Domain

A legfelső szintű tartománynevek (Top Level Domain, TLD) egyedi nevek, ezeket a nemzetközi szervezetek osztják ki, illetve a használatuk körét is ők határozzák meg. A legfelső szintű tartománynevek közé tartoznak a következő háromBetűs nevek: com, org, net, edu, int, mil, gov. Ezek közül a mil és a gov különlegesek, mivel előbbit csak az Egyesült Államok hadserege használhatja, míg utóbbit csak az Egyesült Államok kormányzati szervei. A többi legfelső szintű tartománynév alá gyakorlatilag szabadon lehet neveket bejegyezni, legfeljebb működési, tevékenységi köri megkötések vonatkoznak rájuk.

A legfelső szintű tartománynevek másik nagy csoportja a kétbetűs országkódok csoportja. Ezeket a kétbetűs kódokat az ISO (International Organization for Standardization – Nemzetközi Szabványosító Szervezet) osztotta ki az országoknak.

Az előbb említett legfelső szintű tartományneveken kívül az elmúlt években újabb tartományok jelentek meg a legfelső

szinten, ilyen a .biz vagy a .info. Az összes jelenleg elérhető nem országokat jelölő legfelső szintű tartománynév és a koordináló szervezeteik elérhetősége a ➲ http://www.iana.org/gtld/gtld.htm címen található meg.

Másodszintű tartományok

A másodszintű nevek kiosztása mindig az adott legfelső szintű tartománynév fenntartójának a feladata, így a .hu alá tartozó címeket a magyarországi NIC szervezet osztja ki, mely a ➲ http://www.nic.hu címen érhető el.
Amennyiben valaki a .hu tartománynév alá szeretne nevet bejegyeztetni, akkor ennek feltételeiről és a választható, valamint a már foglalt nevekről a ➲ http://www.domain.hu címen tájékozódhat.
Másodszintű tartománynevet bárki szabadon bejegyezhet, viszont szerencsés lenne, ha a nemrég beindult „jegyezzünk be minden értelmes szót" mozgalom visszaszorulna kicsit, ugyanis – amellett, hogy lassan nem lehet értelmes, még be nem jegyzett szavakat találni – azzal, hogy lassan minden internetre kapcsolt gép saját másodszintű névvel rendelkezik, elvesznek a DNS felépítéséből származó olyan előnyök, mint a kiegyensúlyozott fában való keresés vagy a szervezeti felépítések leírása.

A DNS működéséről

A DNS működése egy elosztott adatbázison alapul, ez jelenleg a világ legnagyobb ilyen típusú rendszere. A DNS-rendszer jelenleg 13 úgynevezett fő (root) kiszolgálóból és számtalan kisebb-nagyobb DNS-kiszolgálóból áll. A 13 fő DNS különleges feladatot lát el, nevezetesen az ő feladatuk, hogy minden egyes tartományhoz ismerjék a kiszolgálógépet vagy az oda vezető utat.
Amikor egy felhasználó egy weboldal letöltését kezdeményezi a böngészőjéből, a gépe megnézi a saját DNS-gyorstárát (cache), hogy az adott címhez tartozó IP megtalálható-e benne. Amennyiben nem találta meg az IP-t, akkor megszólítja a rendszernek megadott egyik DNS-kiszolgálót. Ez a DNS-kiszolgáló szintén megpróbálja a névhez tartozó címet megtalálni, ehhez átnézi a saját adatbázisát, valamint a saját gyorstárát. Minden DNS-kiszolgáló szintén rendelkezik gyorstárral, amelybe a legutóbb használt név–címpárokat raktározza el. Amennyiben a megszólított DNS-kiszolgáló sem találja meg a név feloldását, akkor a hozzá földrajzilag legközelebb eső fő DNS-hez fordul. A fő DNS visszakeresi a névhez tartozó kiszolgálót és elküldi a címet a DNS-kiszolgálónknak, aki ezt a címet egyfelől eljuttatja hozzánk, másfelől elhelyezi a saját memóriájában későbbi hasznosítás céljából. Felmerülhet a kérdés, mi szükség van a ritkán használt címek tárolására. A válasz roppant egyszerű: a ritkán használt címet sem csak egyetlen egyszer használjuk, hanem az adott weboldal megtekintése során is többször szükségünk lehet rá.
Egy DNS-kiszolgáló több tartomány, zóna kiszolgálását is lehetővé teszi. Ilyenkor minden egyes zónához tartozik egy állomány, amely leírja az adott zónába tartozó név–cím-párokat. Ha egy gép adott zónához kéréssel fordul, a kiszolgáló megkeresi a zónaállományban a kérdéses bejegyzést, és amennyiben eredményes volt a kérés, akkor visszajuttatja az őt megszólító géphez. Ezek a kérések és válaszok minden esetben UDP-n (User Datagramm Protocol) keresztül futnak. Az UDP protokoll előnye a TCP-hez képest, hogy mivel nincs kapcsolatfelépítés, valamint nem megbízható kapcsolaton zajlik a kapcsolattartás, ezért a TCP-hez képest lényegesen kisebb adatfolyamot hoz létre, valamint sokkal rövidebb idő alatt megy végbe a lekérdezés. Természetesen hátrányként jelenik meg a csomagvesztés, de ez nem gond, legfeljebb mind a kérést, mind a választ újraküldjük.

A gyakorlat

Jelenleg a Linux alatt a legelterjedtebb DNS-kiszolgáló a BIND, azaz a Berkeley Internet Name Domain, annak is a 8-as, illetve újabban a 9-es kiadása. SuSE alatt a YaST-tal természetesen mindkét kiszolgáló telepíthető, előbbi a `bind8`, utóbbi a `bind9` csomag telepítésével. Én az utóbbi telepítését javaslom, tekintettel arra, hogy frissebb változatról van szó.
Ha végeztünk a telepítéssel, akkor a következő könyvtáraknak, illetve állományoknak kell megjelenniük.

- A */etc/init.d* könyvtárban kell megjelennie a ***named*** nevű futtatható állománynak. E program meghívásával tudjuk a DNS-kiszolgálót indítani, leállítani vagy éppen újratölteni a beállításfájlokat.
- A */etc* könyvtárba kerül a ***named.conf*** állomány, amely a DNS-kiszolgáló alapvető beállításait tartalmazza. Erre az állományra nagy szükségünk lesz a továbbiak folyamán is.
- A további beállításokat, valamint a különböző zónaleírókat a */var/lib/named* könyvtár alatt találjuk, illetve az általunk létrehozott állományokat is ide célszerű elhelyezni.

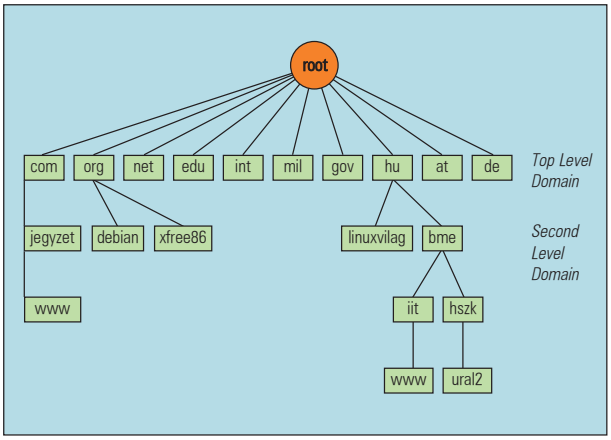
A named.conf

A ***named.conf*** állományban tudunk beállítani minden fontosabb jellemzőt, amely DNS-kiszolgálónk futása során felmerül. Ebben lehet beállítani az úgynevezett ***Master*** és ***Slave*** zónákat és a kiszolgáló jellemzőit, például a használt könyvtárakat, naplózási beállításokat és az elérhetőségi beállításokat.
Kezdjük az `options` résszel. Az első bejegyzés a `directory`, itt adhatjuk meg, hogy a rendszer melyik könyvtárat használja a futása során. Alapértelmezettként a */var/lib/named* van beállítva.
A következő bejegyzés `forwarders` kapcsoló. Ezzel adhatjuk meg azoknak a DNS-kiszolgálóknak az IP-címét vagy csoportját (lásd később), amelyeket a hozzánk beérkező kérések kiszolgálására használni szeretnénk. Amennyiben egy kérés érkezik hozzánk, és nem tudjuk a címet visszafejteni, alapesetben a fő névkiszolgálók valamelyikéhez fogunk fordulni. Ha ebben a bejegyzésben egy vagy több névkiszolgálót adunk meg, először őket kérdezzük le, mielőtt magasabb szinthez fordulnánk.
A `forward first` kapcsoló beállításával azt érjük el, hogy a beérkező kéréseket először külső kiszolgálóval próbáljuk meg feloldani, mielőtt a saját adatbázisunkhoz fordulnánk.
A `listen-on` kapcsolóval be lehet állítani, hogy melyik helyi hálózati csatoló melyik kapuján nyújtsunk DNS-szolgáltatást.
A `listen-on-ipv6` bekapcsolásával értelemszerűen a megadott hálózati csatolón fogadjuk el az IPv6-os kéréseket.

Az `allow-query` kapcsolónál be tudjuk állítani, hogy mely ügyfelek futtathassanak lekérdezéseket a rendszerben. Alapértelmezésként bárhonnan elfogadunk kéréseket.
A `notify` kapcsolóval be tudjuk állítani, hogy a naplóban megjelenjenek-e a kiszolgáló állapotáról szóló üzenetek. Sok tartomány esetén javasolt a naplózás tartományonkénti beállítása, ugyanis a túl sűrű naplózás nagyon leterhelheti a kiszolgálót.
Az `allow-transfer` kapcsolóval korlátozhatjuk azt, hogy mely kiszolgálók kérhessék le DNS-kiszolgálónk zónaállományait. Ezt érdemes beállítani, hogy csak a szükséges kiszolgálók kapjanak ilyen jogosultságot, mivel ezen a felületen keresztül akár szolgáltatásmegtagadásos (Denial of Service, DoS) támadást is lehet indítani a gép ellen, arról nem is beszélve, hogy nem feltétlenül szeretnénk a hálózatunk felépítését mindenki számára láthatóvá tenni. (Tartománynév bejegyzésekor a bejegyző szervezet ellenőrzi, hogy a bejegyzendő zóna megfelel-e a szabályoknak, ehhez viszont `transfer` jogosultságot kell biztosítani.)
A `logging` résznél beállíthatóak a rendszer naplózással kapcsolatos jellemzői. Ezeket hagyhatjuk alapértéken. Amennyiben mégis hozzányúlunk, figyeljünk oda, hogy egy rossz naplózási beállítás felhizlalhatja a napló mértét, vagyis felesleges adatokkal töltheti fel.
Elérkeztünk a zónarészhez. Ebben tudjuk beállítani a különböző kiszolgált zónák leíróinak helyét, típusát, illetve egyéb jellemzőket is megadhatunk, például naplózással kapcsolatos beálltásokat. Gyakorlatilag minden zónához külön-külön megadhatjuk az előbb látott `options` részben beállított értékeket.
Egy zóna bejegyzéséhez legalább az alábbi értékek megadása szükséges:

```
zone "zóna_neve" {
        type zóna_típusa;
        file zóna_állomány;
};
```

Típusnak meg kell adnunk, hogy az adott zóna a kiszolgáló szempontjából milyen típusú. A `forward` zóna minden kérést egy adott kiszolgálóhoz továbbít. A `hint` egy különleges zóna, amely egy fő névkiszolgálóra mutat, arra az esetre, ha olyan névvel találkoznánk, amit nem tudunk feloldani. A `master` – amúgy a leggyakrabban használt típus – egy olyan zóna, amelynek a leírófájlja a kiszolgálón található.
A `slave` egy olyan típus, amit ugyan ismerünk, de a zóna leírójának szerkesztése nem a mi feladatunk, készen kapjuk őket a zóna elsődleges (master) kiszolgálójától.
Az állomány változó értékének azt a fájlt kell megadnunk a ***named*** munkakönyvtárán belül, amelyik az adott zóna leírását tartalmazza.
Ennek a fenti két kapcsolón kívül további korlátozó kapcsolók állíthatók be, mint például az `allow-query` és az `allow-transfer`, ezekről már esett szó, valamint az `allow-update`. Ez utóbbinál megadott gépek vagy gépek csoportja jogosult dinamikus bejegyzéseket elhelyezni a zónaleíró állományban. Erre a dinamikus frissítésre szükségünk lehet, amennyiben például egy windowsos Active Directory DNS kiszolgálását szeretnénk ellátni.



Még egy fontos csoport létezik, amit a ***named.conf*** állományon belül létre tudunk hozni. Amennyiben több IP-címet szeretnénk egy csoportban kezelni, például az `allow-transfer` kapcsolónál, úgynevezett gyűjtőket (access control statement, acl) hozhatunk létre. Egy ilyen acl alá több IP-t is megnevezhetünk, így a gyűjtőnévre való hivatkozás során az összes gyűjtőben lévő gépre hivatkozunk. Az alábbi példával a Budapesti Műszaki Egyetem DNS-kiszolgálóit tettük egy csoportba.

```
acl BME-DNS {
    152.66.115.1;
    152.66.116.1;
};
```

Ezután egy zóna beállításain belül nyugodtan használhatjuk a következő beállítást:

```
allow-transfer { BME-DNS; };
```

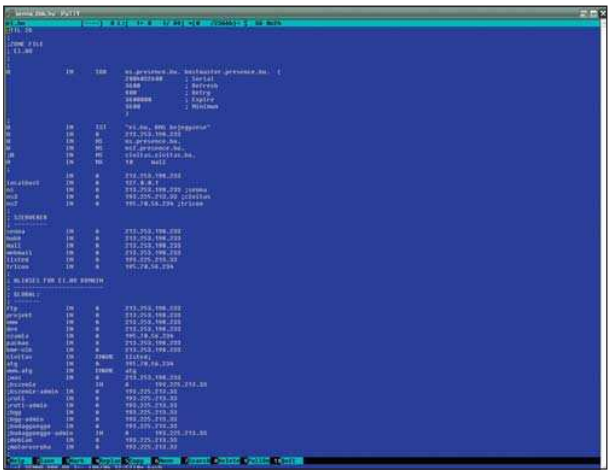
Ezzel a BME DNS-kiszolgálóit jogosítjuk fel zónaátvitelre.

Zónaleírók szerkesztése

Ha végeztünk a ***named.conf*** beállításával, akkor itt az ideje létrehozni a megfelelő zónaleírókat. Ezeket az állományokat célszerű a ***/var/lib/named*** könyvtárban alkönyvtárakba elhelyezni.
A zónaállományok legfontosabb bejegyzéseit, úgynevezett Resource Recordjait (RR) a következőkben foglalhatjuk össze.
Az első Resource Record a SOA (Start of Authority). A SOA olyan fontos adatokat tartalmaz a zónáról, mint a zóna sorszáma, frissítési ideje, lejárati ideje, DNS-címe, illetve a karbantartó elektronikus levélcíme.
Lássunk egy példát!

```
example.com.  800  IN  SOA  ns.example.com.
info.example.com.  1234  3600  900  604800  1800
```

Példánkban az ***example.com*** a SOA rekordja, amit a DNS-kiszolgálók 800 másodpercig tartanak a gyorstárban. Az ***ns.example.com*** a DNS-kiszolgálója, az ***info@example.com***-ra lehet a tartománnyal kapcsolatban levelet írni. Az 1234 a sorszáma, 3600 másodpercenként frissít, 900 másodpercen-



ként történik újrapróbálkozás, ha nem sikerül a művelet, 604 800 másodperc után jár le a bejegyzés érvényessége, valamit legkevesebb 1800 másodpercig érvényes a bejegyzés. Miután létrehoztuk a SOA-t, lehetőségünk nyílik a többi rekordot meghatározni. Általában a következő rekordok szoktak szerepelni egy zónaleírásban.

NS (Name Server)

Ezzel lehet meghatározni a névkiszolgálót. Egy ilyen rekordra mindenképpen szükség van. Például:

```
example.com    IN    NS    ns.example.com
```

MX (Mail Exchanger)

Ezzel a bejegyzéssel tudunk a tartományhoz levélkiszolgálót rendelni. Amennyiben rendelkezünk az ***example.com*** tartománnyal, és benne a ***mail.example.com*** nevű kiszolgálóval, akkor egyfelől küldhetünk levelet az ***info@mail.example.com*** címre, hiszen pontosan megadtuk, melyik gép melyik felhasználójáról van szó. Ha viszont az ***info@example.com*** címre szeretnénk levelet küldeni, szükségünk lesz az MX bejegyzésre, mert ez mondja meg, melyik az alapértelmezett levélkiszolgáló a tartományban. Létre kell hozunk a következő bejegyzést:

```
example.com    IN    MX    10    mail.example.com.
```

Ebben a 10 a kiszolgáló súlyát jelöli, így lehetőségünk nyílik elsőbbségi sorrendet felállítani a különböző kiszolgálók között, ezzel biztosítjuk a megfelelő levelezési szolgáltatást üzemzavar esetére is.

A (Address)

Ez a rekord használatos egy adott névhez IP-cím hozzárendelésére.

```
www    IN    A    123.456.789.001
```

CNAME (Canonical Name)

Ezzel a bejegyzéssel egy adott A rekorddal társított névhez további neveket lehet rendelni. Megtehetjük, hogy az előbb létrehoz ***www.example.com*** kiszolgálóhoz hozzárendeljük az ***ftp.example.com*** nevet a következő módon:

```
ftp    IN    CNAME    www
```

Ugyanakkor azt is megtehetjük, hogy az ***ftp.pelda.hu*** kiszolgálóhoz létesítünk egy bejegyzést ***ftp.examle.com*** néven:

```
ftp    IN    CNAME    ftp.pelda.hu.
```

Itt hívnám fel a figyelmet arra, hogy amennyiben teljes tartományneveket használunk, ne felejtsük le a cím végéről a pontot. Az utolsó pont ugyanis a fő (root) szintet jelöli a bejegyzésben.
A `CNAME`-bejegyzéssel lehetőségünk van rá, hogy dinamikus IP-címmel rendelkező géphez saját magunk által bejegyzett tartománynevet regisztráljunk.
Mindössze valamelyik dinamikus DNS-szolgáltatónál – például a ➲ http://www.dyndns.org – szükséges készíteni egy bejegyzést, majd az így kapott tartománynévre kell egy CNAME-bejegyzést csinálni a saját DNS-kiszolgálónkban.

AAAA (Address for IPv6)

Ugyanazt a feladatot tölti be, mint az IPv4 esetén az A, azzal a különbséggel, hogy itt IPv6-os IP-címet kell megadni.
Egy zóna leírásához használhatók további rekordok is, ezekkel további adatokat közölhetünk a rendszerről, például a kiszolgálógép felépítését, a használt operációs rendszert vagy akár a földrajzi elhelyezkedést. Utóbbi jópofa dolog, ha grafikus `traceroute` programok számára fellelhetővé szeretnénk tenni a kiszolgálót. Paranoiás rendszergazdák ne töltsék ki ezeket a rekordokat.

A DNS frissítése

Amennyiben változás áll be zónában, akkor módosítani kell a megfelelő zónaállományt is. De nem elég, ha módosítjuk a megváltozott bejegyzést, hanem az adott zónához tartozó sorszámot is meg kell növelnünk legalább eggyel. Miután a zóna leíróját mentettük, a `/etc/init.d/named reload` paranccsal frissíthetjük a kiszolgálót. Innentől fogva pár percen, órán belül elterjed a világban a módosított bejegyzés.

Összegzés

A DNS-kiszolgálók felépítéséről, beállításáról rengeteg útmutatást találhatunk az interneten. Erre legjobb kiindulás a Google, vagy valamelyik másik nagy kereső. Maga a DNS-szolgáltatás többet is rejt magában puszta név- és címfeloldásnál, erre jó példa a grafikus útkereső programok által nyújtott szolgáltatás, de DNS-en keresztül akár tanúsítvány- vagy kulcskiosztást is készthetünk.
A lehetőség adott, mindenki nyugodtan ássa bele magát. Jelen cikk terjedelme sajnos nem tett lehetővé bővebb ismertetést, de arra mindenképpen elég volt, hogy áttekintő képet adjon a DNS elvi és gyakorlati működéséről.

Illés Viktor (viktor@ei.hu)
23 éves, a BME műszaki informatikus szakának hallgatója, mellette weblapokkal, linuxos és windowsos rendszerekkel foglalkozik. Szabadidejét legszívesebben a szabadban tölti, teniszezik és kerékpározik.