FreeBSD – a szomszéd vár (7. rész)
Tűzfal készítése

A fájlrendszer megtekintése után elérkeztünk a FreeBSD igazi területéhez, a hálózat kiszolgálásához. Ezen belül először egy olyan területet vizsgálunk meg, amely szinte minden hálózati rendszer esetén alapfeladat: tűzfallal védeni egy számítógépet. Ehhez alapvetően szükségünk lesz egy hálózati csomagszűrő programra, vagy tűzfal programra.

Mi az a tűzfal?

A tűzfalak két hálózatot elválasztó eszközök, bővebb értelemben ezek teszik lehetővé, hogy megszűrjük a rajtuk átfolyó a be-, illetve kimenő forgalmat. Ennek értelmében a tűzfalak feladata a belső (személyes) hálózat számára különféle szolgáltatások nyújtása, illetve ésszerű korlátozások alkalmazása. Gondoljunk csak általánosságban arra az esetre, amikor egy cég a széles sávnak becézett vékonyka ***ADSL*** kapcsolattal tartja a világhálóval a kapcsolatot. Az ***ADSL*** sávszélessége ugyanis nagyon sok mindent kibír, amíg meg nem jelenik a belső hálózaton néhány notórius (film)letöltögető kolléga, akik miatt a többi munkatárs esetleg a munkájában lesz akadályozva. Érdemes ebben az esetben egyfajta letöltési korlátozást bevezetni, így egyenlő esélyt adni minden dolgozónak. Szolgáltatás tekintetében megfontolandó a web adatforgalmát gyorstáron átvezetni, ezzel – kismértékben ugyan, de – gyorsíthatjuk a böngészést.
A tűzfalak fő feladata a csomagok szűrése, s ez sok feltételtől függhet: a feladó vagy a címzett IP címétől, a forrás vagy cél porttól, a csomagok állapotától vagy az integritásától. A szűrés mindenképpen magasabb biztonságot jelent, bár az túl erőteljes korlátozás a munkát is zavarhatja.

Milyen tűzfalak vannak?

Az elkészített tűzfalak két nagy csoportba sorolhatók: megengedő és tiltó. Az előbbi csoport alapvetően minden forgalmat enged, és ezt tudjuk szabályokkal szűkíteni. A tiltó tűzfal mindent tilt, és szabályokkal lehet engedélyezni a meghatározott forgalmat. Az előbbi kényelmesebb, a második biztonságosabb. A másik csoportosítási mód szerint léteznek úgynevezett „***stateful***” tűzfalak, amelyek a kapcsolat kiépítésétől egészen annak megszűnéséig követik az adatokat, ezzel egy sor biztonsági problémát megakadályoznak (csak olyan hálózati csomagokat engednek át, amelyek egy új kapcsolatot nyitnának, illetve egy már létezőhöz tartoznak). Megkülönböztetünk személyes és vállalati tűzfalakat is. Az előbbiek egy-egy gépet védenek a „külvilágtól”, a vállalati tűzfalak a vállalat minden számítógépét a külvilágtól. Szokás a két tűzfalat kombinálni, így a belső hálózat felől induló támadások ellen is védettek a számítógépek, és a vállalaton kívüli támadásoktól is. Ebben a cikkben inkább a személyes tűzfalak kialakításáról szólok, a a vállalati tűzfalakkal következő részben foglalkozunk.

Milyen eszközeink vannak a megvalósításhoz?

Linux esetén minden feladatot a rendszermagba ágyazott modulokkal, és a modulokat kezelő ***iptables*** programmal tudunk megoldani. ***FreeBSD*** esetén a feladatok szét vannak választva, illetve több eszközt is használhatunk, ezek az ***IPFilter (IPF)***, az ***IPFirewall (IPFW)*** és a ***PacketFilter (PF)***. Az ***IPFW*** tartalmaz ***sávszélesség korlátozó eszközt (DummyNet)***, amely kényelmesen használható. Az ***IPF*** ilyet nem tartalmaz, de használhatjuk hozzá az ***AltQ*** programot a ***ports*** adatbázisból. A ***PF*** az ***OpenBSD*** rendszerből származik, szintén kényelmes eszköz a tűzfal elkészítéséhez. Én az ***IPFW*** programot használom, így ezt fogom bemutatni, de a többi eszköz is pont ennyire használható.

Az IPFW beállítása

A rendszermag már tartalmazza modulként a jelenlegi ***FreeBSD*** terjesztések esetén a szükséges komponenseket, így nem kell újrafordítanunk a rendszermagot.
A */etc/rc.conf* állományhoz kell a

```
firewall_enable="YES"
```

sort hozzáfűznünk, és a következő induláskor már működik is az ***IPFW*** rendszermag modul. Természetesen nem kell újraindítani a kiszolgáló gépünket, mivel az ***ipfw*** nevű modul betöltésével azonos eredményt érünk el

```
$ kldload ipfw.ko
```

így a következő induláskor az ***rc.conf*** szerint fog a modul betöltődni. Ellenőrizzük le, hogy működik-e a tűzfal program, a rendszermag üzenetei között meg kell látnunk az alábbi sort:

```
ipfw2 initialized, divert disabled, rule-based
➥forwarding disabled, default to deny, logging
➥disabled
```

Különféle finomítások

A tűzfal beállítása során a legfontosabb, hogy értesüljünk a szabályok működéséről, illetve a program üzeneteiről. Ehhez a naplózást be kell állítani, és ideiglenesen bőbeszédű naplózást is beállíthatunk. Ehhez a `sysctl` programmal a

```
$ sysctl net.inet.ip.fw.verbose=1
```

parancsot kell kiadnunk (vagy ezt be is írhatjuk a */etc/sysctl.conf* állományba).
Az *rc.conf* állományban megadhatjuk a tűzfalszabályokat tartalmazó állomány nevét, amely alapértelmezés szerint a */etc/rc.firewall*.

```
firewall_script="/etc/ipfw.rules"
```

Az ipfw program

A rendszermag modulja valósítja meg a tűzfalat, illetve kezeli annak szabályait. A modul működését egy programmal tudjuk, amely `ipfw` névre hallgat. Ha betöltöttük a modult, akkor azonnal elvágtuk magunkat a külvilágtól, mivel a program minden forgalmat tilt, amely a hálózati csatolófelületeken át folyna. A modul betöltésétől óvakodjunk olyan gépen, amelytől fizikailag messze vagyunk, mivel izzasztó meglepetés érhet minket: nem férünk hozzá a géphez a hálózaton át. Ezt az alapvető viselkedést a rendszermag `IPFIREWALL_DEFAULT_TO_ACCEPT` opciójával tudjuk elkerülni, de ez biztonsági megfontosálok okán nem ajánlatos.
Az alapértelmezett viselkedésről a parancs `list` opciójával győződhetünk meg:

```
$ ipfw list
65535 deny ip from any to any
```

A 65535 szám már sejteni enged egy korlátot: a szabályok maximális száma mindössze ennyi lehet. A számok szerint kerül ellenőrzésre az adott szabály, a tiltás tehát a legutolsó a sorban; ha lenne előtte bármilyen másik szabály, akkor először azt értékeli ki. A tiltás ténye igen látványos a programok felől, ugyanis jogosultsági problémára panaszkodnak majd a hálózatot használni kívánó programok:

```
$ ping -c 1 10.1.1.211
PING 10.1.1.211 (10.1.1.211): 56 data bytes
ping: sendto: Permission denied

--- 10.1.1.211 ping statistics ---
1 packets transmitted, 0 packets received, 100%
➥packet loss
```

Új szabály hozzáadása nagyon beszédes formában történik, az `ipfw` parancssora olyan, mintha angolul elmondanánk a kívánságainkat:

```
$ ipfw add 1 allow ip from me to any
00001 allow ip from me to any
$ ipfw list
00001 allow ip from me to any
65535 deny ip from any to any
```

Ezzel a szabállyal engedélyeztük a saját címről (`me`) az összes másik cím (`any`) felé induló csomagok továbbítását.

Gondolhatnánk, hogy ezzel megoldottunk minden szükséges beállítást, azonban ez kevésnek bizonyul:

```
$ ping -c 1 10.1.1.211
PING 10.1.1.211 (10.1.1.211): 56 data bytes

--- 10.1.1.211 ping statistics ---
1 packets transmitted, 0 packets received, 100%
➥packet loss
```

A hiba kiírása ugyan megszűnt, a csomagok vígan átjutnak a tűzfalon, elérik a célként megnevezett gépet is, onnan a válaszcsomag visszaindul, a tűzfalunkon azonban fennakad. Hozzá kell vennünk a szabályokhoz egy újabb szabályt, amely engedi a forgalmat a mi gépünk felé:

```
$ ipfw add 2 allow ip from any to me
00002 allow ip from any to me
$ ipfw list
00001 allow ip from me to any
00002 allow ip from any to me
65535 deny ip from any to any
```

S már működik is a kommunikáció, a csomagok oda-vissza átjutnak a személyes tűzfalon.

```
PING 10.1.1.211 (10.1.1.211): 56 data bytes
64 bytes from 10.1.1.211: icmp_seq=0 ttl=64
➥time=0.308 ms
```

Naplózás beállítása

A napló vezetése nagyon fontos a hibakeresésnél és a visszakereshető dokumentálás okán. Általában a visszautasított kapcsolatokat szoktuk naplóba vezetni, a legális fogalom naplózása feleslegesen rabolja a gépidőt, illetve időrabló az elemzése is. A feladatunk mindössze annyi, hogy a naplózni kívánt szabálynál az `allow` (`accept`, stb) kulcsszó után felvesszük a `log` kulcsszót is:

```
$ ipfw add 3 allow log ip from 10.1.1.211 to me
```

A fenti szabály működéséhez azonban az előtte lévő másik szabályt át kell mozgatni e szabály utáni pozícióra, ugyanis az elkapja az összes (`any`) beérkező csomagot. Ezt egy törléssel majd egy új hozzáadással tudjuk megtenni:

```
$ ipfw show
00001  981  77572 allow ip from me to any
00002   88  39287 allow ip from any to me
00003  577 160451 allow log ip from 10.1.1.211 to me
65535 1687 294213 deny ip from any to any
$ ipfw delete 2
$ ipfw add 4 allow ip from any to me
00004 allow ip from any to me
$ ipfw show
00001 1313 101607 allow ip from me to any
00003   14   5486 allow log ip from 10.1.1.211 to me
00004    0      0 allow ip from any to me
65535 1879 326108 deny ip from any to any
```

Ha mindent jól csináltunk, akkor a */var/log/security* állományban megjelennek a naplózott forgalmak:

```
ipfw: 3 Accept TCP 10.1.1.211:110 10.1.1.210:63187
➥in via fxp0
last message repeated 8 times
```

Éles helyzetben jól jöhet, hogy megadhatjuk a maximális számát a naplózott soroknak (gondoljunk csak egy jól kivitelezett *DoS* támadásra). Ezen sorok számát lehetőségünk van globálisan, illetve szabályokra lebontva meghatározni. Globálisan a

```
$ sysctl net.inet.ip.fw.verbose_limit=100
```

parancs állítja be ezt, s így minden egyes szabály csak 100 sorral szaporítja a naplóállományunkat, ezzel elegendő információt szolgáltatva a hibákról (vagy támadásokról). Érdemes rendszeresen nullázni a naplózott sorok számát, amely a

```
$ ipfw resetlog
```

parancs hatására fog bekövetkezni, így például minden nap tiszta lappal indulhat a naplózás.
Egyes szabályok esetén külön megadhatjuk a naplózott sormennyiséget, egyszerűen a `logamount n` paramétert kell megadnunk:

```
$ ipfw add 10 allow log logamount 4 ip from
➥10.1.1.211 to me
```

Szűrés port szerint

Személyes tűzfal esetén érdemes szűrni néhány portra, így meghatározhatjuk a legális szolgáltatásokat, amelyeket a helyi hálózat számára használni engedünk. A megoldáshoz egyszerűen csak a megadott IP címek mögé kell írni a port számát:

```
$ ipfw add 2 allow log ip from any to me 22
00002 allow log ip from any to me dst-port 22
$ ipfw list
00001 allow ip from me to any
00002 allow log ip from any to me dst-port 22
00003 allow log ip from 10.1.1.211 to me
00004 allow ip from any to me
65535 deny ip from any to any
```

A naplózást tehát kicseréljük, és csak azon csomagokat kérjük a naplóba írni, amelyek a gépünk 22-es portjára (SSH) érkeznek, s a ***security*** állományban látnunk kell a kéréseket:

```
ipfw: 2 Accept TCP 10.1.1.211:55978 10.1.1.210:22
➥in via fxp0
```

Ha naplózás és engedélyezés helyett eldobnánk a csomagot (`drop`), vagy hibát küldenénk vissza (`unreach`), akkor egy kis változtatással ez is könnyedén mehet:

```
$ ipfw delete 2
$ ipfw add 2 unreach net log ip from any to me 22
00002 unreach net log ip from any to me dst-port 22
$ ipfw list
00001 allow ip from me to any
00002 unreach net log ip from any to me dst-port 22
00003 allow log ip from 10.1.1.211 to me
00004 allow ip from any to me
65535 deny ip from any to any
```

Ha kapcsolódni szeretne valaki, akkor erről értesülni fogunk:

```
ipfw: 2 Unreach 0 TCP 10.1.1.211:58164
➥10.1.1.210:22 in via fxp0
```

Miközben a kapcsolódni vágyó az

```
ssh: connect to host 10.1.1.210 port 22: No route
➥to host
```

üzenetet kapja. Az `unreach` paraméterét (`net`) változtatva szinte bármilyen „hibát” elő tudunk állítani.

Tűzfal programocska

A `firewall_script` paraméterében megadott állományt feltölthetjük a kialakított tűzfal szabályokkal, így minden rendszerinduláskor megfelelően fog működni a személyes tűzfalunk. Az adott állomány tartalma lehet például a következő pár sor:

```
ipfw -q -f flush
ipfw add 100 allow ip from me to any
ipfw add 200 unreach net log ip from any to me
➥dst-port 22
ipfw add 300 allow log ip from 10.1.1.211 to me
ipfw add 400 allow ip from any to me
```

Érdemes az egyes szabályok számozását megfelelően nagy különbségre hagyni, így könnyedén kettő meglévő szabály közé tudunk egy újabbat szűrni különösebb nehézség nélkül:

```
ipfw add 200 unreach net log ip from any
➥to me dst-port 22
ipfw add 250 allow log ip from 10.1.1.211
➥to me dst-port 80
ipfw add 300 allow log ip from 10.1.1.211
➥to me
```

További lehetőségek

Az `ipfw` parancs többi lehetősége a kézikönyve oldalain megtalálható, illetve a következő részben – egy a vállalati tűzfal ismertetésével – sok más hasznos tulajdonságát ismerhetjük meg.

Auth Gábor (auth.gabor@enaplo.hu)
Egy pécsi középiskolában informatikát és programozást oktat. Tíz éve botlott először a UNIX rendszerekbe, 7 év Linux használat után kapta el a FreeBSD lázat, amiből máig nem tudott kigyógyulni.

KAPCSOLÓDÓ CÍMEK

A FreeBSD projekt honlapja: ➲ http://www.freebsd.org

A magyar FreeBSD honlap: ➲ http://www.freebsd.hu

A magyar BSD honlap: ➲ http://www.bsd.hu

A kézikönyv magyar fordítása
➲ http://www.enaplo.hu/FreeBSD/handbook/