Személyi tűzfal használata Linux munkaállomáson (1. rész)

Tűzfalak használata Internetes környezetben ma már nélkülözhetetlen feltétele a biztonságos kommunikációnak. Ha intézményi hálózatról érjük el a „Net”-et, akkor a hálózat üzemeltetője minden bizonnyal megoldja a problémát helyettünk, kérés nélkül is gondoskodik a biztonságunkról, még ha ez sokszor kényelmetlenséget is okoz.

Otthoni munkaállomás esetén azonban nincs mentség, a biztonságos környezetet magunknak kell felépíteni. Első és legkézenfekvőbb intézkedés a magára a munkaállomásra telepített személyi tűzfal használata. Ha abban a kiváltságos helyzetben vagyunk, hogy otthoni gépünkön Linux operációs rendszert használhatunk, akkor a tűzfalért nem is kell messzire menni, be van építve a rendszermagba.
A cikk célja tehát ennek megfelelően:

- A tűzfalak fajtáinak rövid ismertetése
- A ***netfilter*** működésének bemutatása
- Az ***iptables*** használatának alapszintű, de részletes ismertetése sok példán keresztül
- Egyszerű otthoni tűzfal konfiguráció bemutatása

A tűzfalak típusai

Amint az zsenge ifjúkorunk óta valamennyiünk számára jól ismert, tehenekből három félét különböztet meg a tudomány, nevezetesen feketét, fehéret és tarkát (lásd még ***Egyszer volt egy Mehemed***). Így a tűzfaltechnikában kevéssé jártas olvasót sem érheti meglepetésként a tény, hogy tűzfal típusból is többfélét találhat. Mielőtt azonban az rövid bemutatásukra rátérnénk, egyáltalán mik azok a tűzfalak? A tűzfal olyan eszköz, mely az egymással kommunikáló végpontok (pl. felhasználói munkaállomás és távoli webszerver) között helyezkedik el, s bizonyos előre beállított szabályoknak megfelelően engedélyezi (átengedi) vagy tiltja (megszakítja) a forgalmat.
A hálózati kommunikáció legkisebb egységei az adatcsomagok (***IP*** csomagok), melyek fejrésze tartalmazza a csomag kézbesítéséhez, kezeléséhez szükséges kísérő információkat (például forrás cím, cél cím, portok, protokollok azonosítói, stb.), az adatmező rész pedig a ténylegesen átvinni kívánt felhasználói adatokat szállítja.
Az összetartozó adatcsomagok összefüggő adatfolyamot, kommunikációs csatornát alakítanak ki.
A legegyszerűbb tűzfal, az úgynevezett ***csomagszűrő***, mely adatcsomag szinten foglalkozik a rajta keresztül áramló információval, nem vizsgálja, hogy a csomag milyen kommunikáció (például ***TCP*** kapcsolat) része. A csomagszűrő tehát minden adatcsomag sorsáról külön hoz döntést, a megelőző forgalomtól függetlenül. Ennek megfelelően a szűrési szabályok is csak az aktuális csomag fejrészéből kiolvasható információkra hivatkozhatnak. Megadhatjuk például hogy mely ***IP*** címekre/címekről engedélyezzük a csomagok továbbítását, milyen protokollt engedünk át, illetve milyen portok elérését tesszük lehetővé a tűzfal két oldalán, stb. A csomagszűrő tűzfalak csak nagyon egyszerű elválasztást képesek megvalósítani, hatékony védelemre nem alkalmasak.

Fejlettebb megoldás jelentenek az úgynevezett ***állapotfigyelő tűzfalak*** (vagy állapotfigyelő csomagszűrők). Ezek jórészt szintén csak a csomagok fejrészében található információkat vizsgálják, viszont nyilvántartják, hogy az aktuális csomag melyik élő kommunikációs kapcsolat (például ***TCP*** csatorna, ***ICMP*** üzenetváltás, ***DNS*** lekérdezés, stb.) része, s amikor a csomag sorsáról döntenek (továbbítják/eldobják), a csomag kapcsolaton belüli szerepét is képesek figyelembe venni. Állapotfigyelő tűzfallal pl. megvalósítható az alábbi szűrési feltétel is, mely egyszerű csomagszűrővel nem megoldható:

- Kimenő ***ping (icmp-echo-request)*** engedélyezése (kiengedése) tetszőleges külső IP cím (hálózati eszköz) felé
- A kérésre érkező „válasz ***ping***” ***(icmp-echo-reply)*** beengedése

A „válasz ping”-et tehát csak akkor engedi át a tűzfal ha előtte kiment egy kérés csomag, s a választ csak arról a külső címről fogadja el, melyre az eredeti ***ping***-et küldték. A beérkező csomag sorsát tehát a csomagnak a tűzfal által nyilvántartott nyitott kommunikációs kapcsolatokhoz való viszonya is befolyásolja. (***Ping*** küldéskor a tűzfal „megjegyzi”, hogy erre választ várunk, s azt is, hogy a válasznak honnan kell érkeznie. A válasz beérkezésekor a „megjegyzést” törli, így

ha ugyanarról a külső címről egy újabb, kérés nélküli „válasz *ping*”érkezik, az már nem jut át.)
Állapotfigyelő tűzfalak szűrési szabályainak megadásakor tehát a normál fejrész információkon kívül a csomag kommunikációs kapcsolaton belüli szerepére is hivatkozhatunk.
A példában említett ***ping*** esete ugyan talán a legegyszerűbb kommunikációs kapcsolat, de bonyolultabb (például ***TCP***) csatornák esetén is érvényesül ugyanez az elv.
A hálózati forgalom feletti legmagasabb szintű kontrollt az ún. ***alkalmazási réteg szintű*** tűzfalak valósítják meg. Ezek már belelátnak a csomagok adatmezőibe is, s az ott szállított információkat összefüggő adatfolyamnak tekintve alkalmazásszintű szűrést is végezhetnek. Blokkolhatják a rosszindulatú tartalmakat szállító web kapcsolatokat, a veszélyesnek ítélt email adatátvitelt, azonnali üzenetküldő csatornákból kiiktathatják a fájl továbbítást, stb. A legmagasabb szintű kontroll azonban nyilvánvaló módon a legnagyobb bonyolultsággal is együtt jár, mind a működés, mind a konfiguráció szintjén.
Az intézményi tűzfalak rendszerint különálló gépen helyezkednek el, s az intézmény belső hálózatát választják el a külső, nem biztonságos hálózattól (például az internettől). A személyi tűzfalak ezzel szemben általában egy-egy munkaállomást védenek, s magára a védendő munkaállomásra telepítjük őket. Az alábbiakban megvizsgáljuk, hogyan alakítható ki egyszerű személyi tűzfal megoldás a Linux kernel beépített tűzfal funkcionalitására támaszkodva.
A megvalósítani kívánt védelem lényege tehát a következő: Saját linuxos munkaállomásunkon felélesztjük a beépített tűzfal szolgáltatást és azt a munkaállomás védelmére alkalmas szűrési szabályokkal látjuk el.

Kernelszintű tűzfalfunkciók a Linuxban

Amint arról korábban már szó esett, a ***Linux*** kernel beépített tűzfal funkcióval rendelkezik, melynek fontosabb jellemzői az alábbiak:

- A beépített tűzfal működhet egyszerű csomagszűrőként vagy állapotfigyelő tűzfalként. A két üzemmód nem lehet egyszerre aktív, a kernel fordításánál kell eldöntenünk, melyik megoldást választjuk. (Aki nem maga fordítja a kernelt, az ma már minden bizonnyal az állapotfigyelő funkciót találja a készen kapott bináris rendszermagban.)
- A kernel szintű csomagszűrő funkciót ***netfilter***nek nevezik, mely közvetlenül vagy modulként is belefordítható a kernelbe.
- A ***netfilter*** támogatja a ***NAT***-ot és a ***MAC*** alapú címzést is. (Ha valaki nem tudja, mit jelentenek ezek a rövidítések, beleértve az előző pontban említett „modulként való fordítás” lehetőségét is, semmi gond, a cikket azért még érdemes tovább olvasni.)
- Az állapotfigyelő csomagszűréshez legalább 2.4-es kernelre van szükség.
- A kernel szintű csomagszűrés konfigurálása (vagyis a tűzfal szűrési szabályainak kijelölése) két felületen, az ***ipchains***-en vagy az ***iptables***-en keresztül történhet. Az ***ipchains***-en az állapot független, az ***iptables***-en az állapotfüggő funkciók konfigurációja végezhető. Ha a kernel fordításakor az állapotfüggő tűzfal megoldás beépítése mellett döntöttünk, akkor értelemszerűen a szabályok kijelöléséhez az ***iptables*** alkalmazást kell használnunk.

A továbbiakban feltételezzük, hogy a kernel tartalmazza az állapotfüggő tűzfal funkcionalitást (ami a készen kapott rendszermagok esetében általános) és a konfigurálásához az ***iptables*** programot használjuk.

1. táblázat

INPUT	Azok a csomagok kerülnek ebbe a láncba, melyek kívülről, a hálózatról érkeznek és címzettjük a védett gép
OUTPUT	Azok a csomagok kerülnek ebbe a láncba, melyek forrása a védett gép, címzettje pedig egy másik, a hálózaton keresztül elérhető eszköz.
FORWARD	Azok a csomagok kerülnek ebbe a láncba melyek forrása és címzettje egyaránt valamely hálózaton keresztül elérhető külső eszköz.. A védett gép feladata csak ezeknek a csomagoknak a továbbítása. (Akkor érdekes ha az adott gépen a csomagtovábbítás engedélyezve van.)

Az állapotfigyelő csomagszűrés működése

A kernel a hálózatról érkezett vagy oda kilépni szándékozó csomagokat a ***netfilter*** feldolgozási láncaiba irányítja. A legfontosabb feldolgozási láncok a következők: *1. táblázat*.
Vannak még más előre definiált láncok is, és mi magunk is hozhatunk létre láncokat. A láncok kezelésével kapcsolatos fontosabb tudnivalók a következőkben foglalhatók össze:

- Az egyes láncokra szabályokat adhatunk meg, melyeket a ***netfilter*** érvényesít. A szabályok a láncba került csomagok tovább engedésére, eldobására vagy naplózására vonatkozhatnak.
- A szabályok sorrendje fontos. Amikor a rendszer a soron következő csomag sorsáról dönt, megvizsgálja, hogy a megadott szabályok közül illeszkedik-e valamelyik a csomagra. A keresést mindig a szabálylista elején kezdi, és az első illeszkedő szabálynál fejezi be. (Kivéve például a naplózást. Lásd később.)
- A szabály rendszerint egy feltételből és egy akcióból áll. Ha a feltétel illeszkedik a csomagra, akkor a rendszer végrehajtja az akciót, például eldobja a csomagot.
- Ha a csomagra egyik szabály sem illeszkedik, akkor a lánc alapértelmezett szabálya lép életbe. (Minden láncra vonatkozik egy alapértelmezett szabály.)

Mivel az otthoni munkaállomások rendszerint nem végeznek csomagtovábbítást, így az alábbiakban csak az ***INPUT*** és ***OUTPUT*** láncok kezelésével foglalkozunk.

A szűrési szabályok megadása

Annak érdekében, hogy a beépített tűzfal működését saját igényeinkhez illesszük, a ki és belépni szándékozó csomagokra (`INPUT` és `OUTPUT` lánc) szabályokat kell meghatároznunk.
A szabályok megadása hagyományos módon az `iptables` paranccsal (parancssor programmal) történik. Léteznek ugyan különböző grafikus felületek is az ***iptables*** fölé, melyek megkönnyítik a tűzfal szabályok kijelölését, ezek azonban rendszerint a lehetőségeket is szűkítik. Így a továbbiakban csak az `iptables` parancs használatára koncentrálunk.
Szabály megadásának formátuma az `iptables` paranccsal:

```
iptables parancs láncnév sor-
➥szám feltétel művelet
```

Lássuk az egyes paraméterek jelentését.
A `parancs` az új szabály felvételének módja a meglévő szabályok közé. Lehetséges értékei: *2. táblázat*.
A `láncnév` annak a láncnak a megadására való, melyre a szabály vonatkozik. Szokásos értékei: *3. táblázat*.
A `sorszám` annak a szabálynak a sorszáma, melyre a művelet vonatkozik. `-A` (hozzáfűzés) esetén nem használjuk.
A `feltétel` a kezelni kívánt csomag meghatározása. (Vagyis itt kell megadnunk, hogy a szabály milyen csomagokra vonatozzék.) A fontosabb lehetőségek: *4. táblázat*.
A feltételekből természetesen több is megadható ugyanabban a parancsban. A !-el a az illeszkedési feltételek általában invertálhatók.
Végezetül a `művelet` a feltételre illeszkedő csomagok kezelési módja. (Itt határozzuk meg, hogy a megadott feltételre illeszkedő csomaggal pontosan mit is kell csinálni). Megadása:

```
-j kezelési_mód
```

A `kezelési_mód` lehetséges értékei: *5. táblázat*.
Az `iptables` parancs a fenti formájában alkalmas arra, hogy jól meghatározott csomagokat válasszunk ki az átlépni szándékozó üzenetfolyamból és rendelkezzünk a kezelésük módjáról. Mi történik viszont azokkal a csomagokkal, melyekre a rendszer nem talál illeszkedő szabályt? Nos, ezek sorsáról a láncok alapértelmezett kezelési szabályai döntenek.

2. táblázat

`-I`	Szabály beszúrása az adott sorszámú helyre
`-D`	Az adott sorszámú szabály törlése
`-R`	Az adott sorszámú, már létező szabály cseréje a parancsban megadott szabályra
`-A`	A szabály hozzáfűzése a meglévő szabálylista végéhez

Láncok alapértelmezett kezelési szabályai

Ha egyik megadott kezelési szabály sem illeszkedik az éppen feldolgozásra váró csomagra, akkor a csomag sorsát az alapértelmezett kezelési mód dönti el. Az alapértelmezett kezelési mód beállítása a következőképpen történik:

```
iptables -P lánc_név
➥kezelési_mód
```

A `kezelési_mód` lehetséges értékei: `DROP, ACCEPT, REJECT, LOG`

Ennek értelmében (például) a beérkező csomagok feldolgozása az alábbi módon megy végbe:

1. A csomag megérkezik a hálózatról
2. A csomagot a kernel az `INPUT` láncba irányítja
3. A csomag feldolgozásakor a ***netfilter*** sorszám szerint növekvő sorrendben végignézi az `INPUT` láncra megadott szabályokat. Az első illeszkedő szabálynál megáll, végrehajtja a szabály által kijelölt akciót, majd veszi a következő csomagot. (Kivétel a `LOG` kezelési mód. Ebben az esetben a tűzfal naplózza a csomag megjelenését, s folytatja az illeszkedő szabály keresését.)
4. Ha a rendszer az `INPUT` lánc szabálylistájában nem talál illeszkedő kezelési utasítást, akkor a csomag sorsát az `INPUT` lánc alapértelmezett kezelési módja dönti el.

Lássunk két példát

Először nézzük meg, mit kell tennünk, ha azt szeretnénk elérni, hogy valamennyi hálózati kommunikáció alapértelmezés szerinti tiltva legyen.

3. táblázat

`INPUT`	Beérkező csomagok lánca
`OUTPUT`	Kilépni szándékozó csomagok lánca

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

A művelet eredményeként a rendszer eldob minden beérkező és távozni szándékozó csomagot, hacsak egy illeszkedő szabály másként nem rendelkezik. Működő tűzfalaknál rendszerint ez a beállítás használatos, alapértelmezés szerint minden kommunikáció tiltva van, amit mégis engedélyezni akarunk, arról (alkalmas illeszkedő szabály megadásával) külön rendelkezünk.
A második példában a beérkező csomagokat letiltjuk, a kimenőket azonban engedélyezzük, vagyis a hálózati kommunikáció számára amolyan egyirányú utcát alakítunk ki.

```
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
```

A művelet eredményeként a rendszer eldob minden beérkező csomagot, viszont a csomagok távozását engedélyezi. (Minden, a védett gépről induló csomagot kiengedünk, de hogy mely csomagok bejutását engedélyezzük, azt külön határozzuk meg, az ***INPUT*** láncra megadott szabályokkal.) Személyi tűzfalak esetén gyakori beállítás. Ilyenkor az ***INPUT*** láncra olyan csomagok átengedését engedélyezzük, melyek már egy létező (belülről kezdeményezett) kommunikációs kapcsolat részeként érkeznek.

4. táblázat

`-s forrás_IP_cím`	A forrás IP cím, ahonnan a csomagot küldték. Megadható pontozott decimális alakban (pl. 10.21.32.43) vagy domén neves formában (pl. mozdony.sihu.hu) Lehet hálózati cím is cím/maszk alakban. Ilyenkor a szabály valamennyi olyan csomagra vonatkozik, melynek forrása az adott hálózat. Pl. `-s 10.21.32.43` – azokat a csomagokat választja ki, melyeket a 10.21.32.43 IP című hálózati helyről küldtek.
`-d célcím`	Cél IP cím, ahová a csomagot küldték. A részletek megegyeznek az előző pontban írtakkal.
`-i bemenő_interfész`	Az az interfész (hálózati kártya), melyen a csomag belépett. Pl. eth0. (Csak az INPUT lánc esetén használható.)Pl. `-i eth0` – azokat a csomagokat jelenti, melyek az eth0 jelű hálózati interfészen (hálózati kártyán) keresztül lépnek be a védett gépre.
`-o kimenő_interfész`	Az az interfész (hálózati kártya), melyen a csomag távozni fog. (Csak az OUTPUT lánc esetén értelmes.)
`-p protokoll`	A csomag adatmezője által használt kommunikációs protokoll. Lehetséges értékek: tcp, udp, icmp, all (minden).Pl. `-p tcp` – a feltétel a TCP csomagokra illeszkedik
`--dport célport`	A cél port, ahol a csomagot várják. Lehet szám, vagy az /etc/services-ben megadott név. Port tartomány megadása: kezdő_sorszám:záró_sorszámPl. `-dport 80` – azokat a csomagokat választja ki, melyeket valamely hálózati eszköz 80-as portjára küldtek
`--sport forrásport`	A forrás port, ahonnan a csomagot küldték. (ld. fent)
`--tcp-flags maszk lista`	A maszk-ban felsoroljuk (vesszővel elválasztva), hogy a csomag fejrészében mely TCP jelzőbiteket (flag-eket) kívánjuk vizsgálni, a listában pedig megadjuk, hogy ezek közül melyiknek kell beállítva lennie. A lehetséges jelzőbitek: SYN, ACK, PSH, URG, RST, FIN, ALL Pl. `--tcp-flags SYN,ACK SYN` – Kiválasztja mindazokat a TCP csomagokat, melyek SYN flag-je be van állítva (1), ACK flag-je pedig törölt (TCP kapcsolat felvétel első fázisa).
`--icmp-type típus`	Ha a szabállyal ICMP csomagokat kívánunk kezelni, akkor itt adhatjuk meg az illeszkedő ICMP csomag típusát. A lehetséges típusok az iptables -p icmp -h paranccsal listázhatók ki. Csak ICMP protokoll (-p icmp) esetén használható.Pl. `--icmp-type echo-request` – a ping csomagok kiválasztása
`-m state --state állapotlista`	A szabállyal kezelni kívánt csomagok kommunikációs kapcsolaton belüli állapota határozható meg ezzel a paraméterrel. Lehetséges értékek (állapotok):· • NEW – A csomag egy új kapcsolatot indít· • ESTABLISHED – A csomag egy már létező kapcsolat része· • RELATED – A csomag egy új kapcsolatot indít, de társítható egy létező kapcsolathoz is. (Pl. FTP adattranszfer, vagy ICMP hibaüzenet, mely egy másik kommunikációs kapcsolat eredményeként keletkezett)· • INVALID – A fentiek szerint nem azonosítható csomag Pl. `-m state --state ESTABLISHED` – azokra a csomagokra illeszkedik, melyek egy, már létező kommunikációs kapcsolat részei.

5. táblázat

ACCEPT	A csomag tovább engedése. (Áthaladhat a tűzfalon.)
DROP	A csomag eldobása hibaüzenet nélkül. (A csomagot a tűzfal eldobja, anélkül, hogy értesítené az eldobott csomag küldőjét.)
LOG	A csomag feljegyzése a rendszernaplóba (syslog). (Ezt követően tovább folytatódik a listában az illeszkedő szabályok keresése, hiszen a csomag sorsa még nem dőlt el.)
DROP	A csomag eldobása ICMP hibaüzenet visszaküldése mellett. (A hibaüzenet típusa megadható.) A csomagot a tűzfal eldobja és értesíti a csomag küldőjét a műveletről.

Ezzel a ***netfilter*** és az ***iptables*** működésével, használatával kapcsolatos alapismeretek bemutatásának a végére értünk. A sorozat következő részében az itt leírtakat a gyakorlatban is alkalmazni fogjuk, konkrét célokat szolgáló szűrési szabályokat fogunk beállítani, és összeállítunk egy egyszerű, otthoni munkaállomás védelmét szolgáló szabálylistát is.

Nagy Sándor
(nasi63@gmail.com)