Az iptables rejtett értékei

Néhány érdekes iptables bővítmény segítségével nemcsak karakterláncokat vagy kaputartományokat tudunk egyeztetni iptables szabályokban, de csapdát is állíthatunk a hálózatot rossz szándékkal használóknak.

Az iptables használatával akár néhány perc alatt is hatékony tűzfalat építhetünk, ráadásul a költségek a kereskedelmi termékek árának töredékét teszik ki. Az alapszintű iptables tűzfalak csomagszűrést végeznek, vagyis a hálózati forgalmat csomagonként vizsgálják, illetve a csomagok kezelésével kapcsolatos döntéseiket különkülön hozzák meg minden csomagra. Egyszerű beállításokkal megadott csomagok eldobását és továbbengedését érhetjük el. Annak meghatározása, hogy az adott csomagra éppen melyik házirendet kell alkalmazni, általában a cél IP-címének és kapuszámának, illetve az utazás irányának alapján történik. Az iptables állapotadatokat is képes használni, tehát képes megalapozottabb döntéseket hozni annak a kapcsolatnak az állapotát figyelembe véve, amelyhez a csomag tartozik. Egyszerű, mégis hatékony az a tűzfal, amely például tiltja a bejövő TCP/IP csatlakozási csomagokat és az Internet felől kezdeményezett, UDP alapú adatcseréket, miközben engedélyezi a kimenőket. Ez a megoldás szabad hozzáférést biztosít a felhasználóknak a külvilághoz, ám a nemkívánatos bejövő próbálkozásoktól megvédi őt. Természetesen egy ilyen rendszer önmagában még túl egyszerű lenne, igazi használhatóságához további szűrésekre is szükség van, de nagyvonalakban erről van szó. Az iptables természetesen ennél az egyszerű csomagszűrésnél sokkal többre képes. Ezek azok a rejtett kincsek, amelyekkel meg szeretném ismertetni a kedves olvasókat. Mivel a program szolgáltatásainak és beállításainak ismertetéséhez egy vastagabb könyv terjedelme is kevés lenne, most csak arra szorítkozom, hogy rövid tájékoztatást adjak ezekről.

Ismerkedés a POM-mal

A Netfilter két összetevőcsoportból áll, az egyik a rendszermag-térben a másik a felhasználói térben fut. A felhasználói térben futó programcsoportba az iptables és a hozzá kapcsolódó segédprogramok, könyvtárak, kézikönyvek és parancsfájlok tartoznak. A rendszermag térben futnak a rendszermagfoltok és számos további kiegészítő modul. Egy avatatlan felhasználó számára rémisztő lehet a foltok telepítése egy olyan nagy és összetett rendszerre, mint amilyen a Linux rendszermag, az út ráadásul buktatókkal és zsákutcákkal teli.

Egy hibás, vagy az összetevők együttműködését akadályozó folt hatására könnyen előfordulhat, hogy a rendszermagot nem lehet lefordítani, vagy – ami még rosszabb – nem lehet vele elindítani a rendszert A Netfilter csapata éppen ezen nehézségektől próbál megkímélni bennünket, ezért készítették el önműködő útitársunkat ***Patch-o-matic***, röviden POM névvel. A POM foltok gyűjteménye, melyhez tartozik egy, azok telepítésére való, kezdők számára is könnyen használható parancsfájl. Ezen foltok némelyike alapvető, ezekre minden iptables/Netfilter telepítésnél szükség van. Mások kiegészítő jellegűek, ezek pedig érdekes szolgáltatásokat nyújthatnak. Ezek tehát azok a rejtett kincsek, melyeket a POM leírása csak a „hol haszontalan, hol remekül használható” bővítményekként emleget.
A POM futtatása roppant egyszerű. Le kell tölteni a legújabb Patch-o-matic tar archívumot az ➲ ftp.netfilter.org /pub/patch-o-matic könyvtárból, majd – root jogosultságokkal – futtatni kell az alábbi parancsot. Ügyeljünk arra, hogy a `KERNEL_DIR`-ben megadott érték a rendszermag forrását tartalmazó könyvtár nevét pontosan tartalmazza:

```
KERNEL_DIR=/usr/src/linux-2.4 ./runme extra
```

Innen kezdve a telepítés interaktív és többé-kevésbé közérthető módon történik.

Karakterlánc-darabkák

A POM talán legelterjedtebb darabja a ***string*** modul. Segítségével a csomagok adatrészének tartalmát lehet összevetni karakterláncokkal. A modul számtalan célra használható, de beállításakor némi figyelemre is szükség van. Tegyük fel például, hogy meg kívánjuk akadályozni az ELF futtatható állományok letöltését. E célból beállíthatunk egy szűrőt, amely az Internet felé néző hálózati csatolón beérkező forgalom TCP/IP alapú, 80-as forráskapuról érkező részét vizsgálja. Ha tudjuk, hogy az ELF fájlok a 7f hexadecimális karakterpárral kezdődnek, amit az ELF karakterlánc követ, akkor karakterláncok egyeztetésével megtalálhatjuk az ilyen részleteket. Nem ASCII karaktereket a csővezeték szimbóluma közé zárva illeszthetünk be a karakterláncba, mint például |7F|ELF. Feltéve, hogy az internet felé az `eth0` csatoló néz, a szükséges parancs a következő:

```
iptables -A FORWARD -i eth0 -p tcp -sport 80 \
   -m string –string ´|7F|ELF´ -j DROP
```

A hexadecimális karakterek beágyazásának lehetősége az iptables 1.2.8-as változatával jelent meg. Ha korábbi változatot használunk, akkor valamilyen trükkhöz kell folyamodnunk. Például a

```
--string "`dd if=/bin/ls bs=4 count=1
➥2>/dev/null`"
```

paranccsal a */bin/ls* program első négy karakterét vehetjük, és mivel ez is egy ELF állomány, pontosan a szükséges karakterláncot fogjuk kapni. Példánkat tovább finomíthatjuk, ha azt mondjuk, hogy a 192.168.0.5 című kiszolgálóról származó tartalomban megbízunk, így az ő esetében nem akarjuk alkalmazni a szűrést. Ezt könnyedén megtehetjük, ha fordított egyeztetés segítségével ellenőrizzük az IP-címet, valahogy így:

```
iptables -A FORWARD -i eth0 -p tcp ! \
   -s 192.168.0.5 --sport 80 -m string \
   --string ´|7F|ELF´ -j DROP
```

Sajnos a példánk több sebből is vérzik, és ezek ráirányítják a figyelmet a karakterlánc-egyeztető modul hiányosságaira. Először is, ezzel a szűréssel a megadott karakterláncot az adatsorban bárhol megtaláljuk, nemcsak a fájlok elején. A szabály hatására tehát nem kívánt egyezések is fellépnek, és olyan csomagokat is eldobunk, amelyeket nem kellene. Másodszor, ha a keresett karakterlánc két egymást követő csomagba oszlik el, akkor nem fogjuk felismerni. A modul csak akkor dolgozik eredményesen, ha a teljes karakterláncot megtalálja az adott csomagban.
Szép és jó tehát a `string` modul, de csak alapszintű szolgáltatásokat nyújt. Nem teszi lehetővé, hogy megkülönböztessük a kis- és nagybetűket, vagy meghatározzuk a keresett karakterlánc elhelyezkedését, ráadásul az adatfolyamon belül több csomagba szétosztott karakterláncok felismerésére sem képes. Nem mondhatjuk tehát, hogy ne lenne hová továbbfejleszteni a modult.

Kevesebb szabály: szebb az élet

Az `mport` bővítmény lehetővé teszi, hogy egyetlen szabállyal több kaput vagy kaputartományt is megadjunk. Alapesetben az iptables parancsokban csak egy-egy kapu vagy a szomszédos kapukat felölelő kaputartomány adható meg. Az mport telepítése után bonyolultabb szabályokat is megadhatunk: egyetlen paranccsal is megoldhatjuk például az X-terminálok használatának, a webezésnek és a levelezésnek az engedélyezését:

```
iptables -A INPUT -p tcp -m mport \
   --dports 80,110,21,6000:6003 -j ACCEPT
```

Az mport nélkül mindehhez négy külön parancsra lett volna szükség:

```
iptables -A INPUT -p tcp --dports 80 -j ACCEPT
iptables -A INPUT -p tcp --dports 110 -j ACCEPT
iptables -A INPUT -p tcp --dports 21 -j ACCEPT
iptables -A INPUT -p tcp --dports 6000:6003 \
       -j ACCEPT
```

Négy helyett egyetlen szabály – akár a rendszer teljesítményének javulását is eredményezheti, hiszen a csomagok feldolgozása kevesebb munkával jár. A szabályfájlok karbantartása is egyszerűbb, hiszen az azonos jellegű csomagkezelést igénylő szolgáltatásokat csoportosítani tudjuk. Mint néhányan talán már kitalálták, az `mport` név a "multiple ports", azaz a "többszörös kapu" szókapcsolat rövidítése.

Időalapú szabályok

A `time modul` segítségével a szabályok érvényességét időszakok és napok alapján tudjuk korlátozni. Megtehetjük például, hogy a személyes weboldalak elérhetőségét az ebédidőre korlátozzuk, vagy a szokásos karbantartások időtartamára a webes forgalmat egy tartalék kiszolgálóra irányítjuk.
Az alábbi szabállyal minden péntek reggel 4:00 és 6:30 között elérhetetlenné tehetjük a webszolgáltatást, például a rendszer karbantartása miatt:

```
iptables -A INPUT -p tcp -d 80 -m time \
   --timestart 04:00 --timestop 06:30 --days Fri \
   --syn -j REJECT
```

Megjegyezném, hogy a `-timestart`, `-timestop` és a `-days` kapcsolók egyikét sem szabad elhagyni. Ha tehát olyan szabályt akarunk írni, amely napoktól független, akkor mind a hét napot fel kell sorolnunk, mivel a kapcsolót nem hagyhatjuk el.

Csapdában

Az iptables TARPIT összetevőjét amolyan hálózati légypapírnak is nevezhetnénk. Az alapelv: aki túl közel jön, az nem egykönnyen távozik. Aki elég merész ahhoz, hogy TCP/IP kapcsolatot nyisson egy légypapírkapura, az csak nagyon nehezen tudja azt lezárni, így rendszerének erőforrásait sem tudja felszabadítani, és újra felhasználni.
A csapdaállítást az iptables úgy végzi, hogy először is fogadja a bejövő TCP/IP kapcsolatot, ezt követően pedig nulla méretű ablakra vált. Ilyenkor a támadó nem tud több adatot küldeni. Nagyjából olyan a hatás, mintha CTRL-S billentyűkombinációt nyomnánk egy terminálon.
A támadónak a kapcsolat lezárására irányuló próbálkozásait figyelmen kívül hagyjuk, így a kapcsolat aktív marad, és jellemzően csak 12-24 perc elteltével bomlik fel. A módszer a támadó erőforrásait felemészti, a légypapírt futtató Linux kiszolgáló vagy tűzfal erőforrásait viszont nem. A légypapírra a következő paranccsal irányíthatunk át csomagokat:

```
iptables -A INPUT -p tcp -m tcp -dport 80 -j TARPIT
```

A `conntrack` és a `TARPIT` azonos rendszeren való használata erősen ellenjavallt, ugyanis minden beragadt kapcsolat fogyasztja a conntrack erőforrásait.
Nagyon jó módszer a támadók megzavarására, ha a netbios kapukat hagyjuk válaszolni a kapupásztázásokra, és ezzel linuxos rendszerünket Windowst futtató gépnek láttatjuk.

Az ezekre befutó próbálkozásokat csak át kell irányítanunk a légypapírra. A hatás lenyűgöző, a támadó hosszasan fog vacakolni azzal, hogy a biztonsági résen keresztül hozzáférést próbáljon nyerni rendszerünkhöz.
A hibás működésűnek tűnő távoli gép, és a hosszúra nyúló időtúllépések jó eséllyel az őrületbe fogják kergetni. Ha ezt a hatást kívánjuk elérni, a következő szabályt kell használnunk:

```
iptables -A INPUT -p tcp -m tcp -m mport \
   --dports 135,139,1025 -j TARPIT
```

Egy másik lehetőség, ha minden kaput TARPIT-játékszerré avatunk, kivéve az eredetileg is használni kívántakat. Ilyenkor a kívülállók úgy látják, mintha minden kapu nyitva lenne, és rengeteg időt pazarolnak a bejutásra. Ráadásul ilyesfajta beállításokkal megakadályozhatjuk a *tcpdump*ot a kiszolgálón futó operációs rendszer helyes felismerésében. Az alábbi szabályokkal a webes és a levelezési forgalmat engedélyezzük, a többit a légypapírra pöcköljük:

```
iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
iptables -A INPUT -p tcp -m tcp --dport 25 -j ACCEPT
iptables -A INPUT -p tcp -m tcp -j TARPIT
```

Megemlíteném, hogy a ➲ www.spinics.net/lists/netfilter/msg17583.html címen egy érdekes, valóban megtörtént esetről olvashatunk, ahol a TARPIT és a string húzott ki a bajból egy rendszergazdát.

Véletlenül

A random modul nem tesz mást, mint véletlenszerűen mutat egyezést a csomagal. Működését úgy szabályozhatjuk, hogy megadjuk az egyezés valószínűségét, ennek értéke 0 és 100 százalék közé eshet. A modul alkalmas például meghibásodott kapcsolat vagy kiszolgáló szimulálására, illetve a terhelés elosztására több, tükrözött kiszolgáló között. Az alábbi példa szerint a webes forgalmat három kiszolgáló között osztjuk el. Az első szabály a kapcsolatok egyharmadát a 192.168.0.100 címen üzemelő kiszolgáló felé irányítja. A második egyharmad a 192.168.0.101, a maradék pedig 192.168.0.102 címre jut:

```
iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -m random --average 33 \
   -j DNAT --to-destination 192.168.0.100:80

iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -m random --average 50 \
   -j DNAT --to-destination 192.168.0.101:80

iptables -t nat -A PREROUTING -i eth0 -p tcp \
   --dport 80 --syn -j DNAT \
   --to-destination 192.168.0.102:80
```

Többet és többet

Némi kutakodással számolatlan további kincset találhatunk – és ezek rengeteg örömet fognak szerezni. Itt csak néhányat említhettem meg, s ez csak elenyésző töredéke a kincses ládikó tartalmának. Ha tudni akarjuk, mi minden áll rendelkezésünkre, futtassuk a ***runme*** parancsfájlt, és böngésszük át a foltok leírásait. Csak néhány kiragadott példa a gyűjteményből:

- Kapcsolatkövetés RSH, MMS (médiafolyamok), PPTP, Quake, RPC és Talk forgalomhoz
- A beállítások és állapotadatok /proc fájlrendszeren keresztüli elérésének kiterjesztett támogatása
- Az IPv6 szolgáltatásainak kibővített támogatása
- Az IP-csomagokban található beállítások, élettartamérték stb. módosítása
- A NAT-olt kapcsolatok kifinomultabb kezelése
- Sávszélesség-használatra vonatkozó korlátok és kvóták kezelése
- A futó operációs rendszer lekérdezését meghiúsító, és a kapupásztázást felismerő megoldások
- Kapcsolatok megjelölése és a jelölések ellenőrzése

Tudástárak

A POM-ban szereplő foltok leírásai nem kerülnek be az ***iptables man*** oldalai közé, ezért azokat máshol kell keresnünk. A bővítmények használatához szükséges alapvető írásmódot az iptables beépített súgójával jeleníthetjük meg. Ha például kiadjuk az `iptables -m random -help` parancsot, akkor egyrészt a szokásos súgószöveget látjuk, másrészt ennek végén a random modul kapcsolói is felbukkannak. Hasonló eljárással a többi modulról is nyerhetünk alapszintű tájékoztatást.
Az egyes modulok leírásait a Patch-o-matic könyvtárszerkezetének megfelelő pontjain is megtaláljuk. Például a random modul leírása a ***base/random.patch.help*** fájlban szerepel. Hasonló fájlok tartoznak a többi folthoz is.
Végül érdemes ellátogatni a Netfilter webhelyére, a ➲ www.netfilter.org/patch-o-matic címre, itt minden POM folt részletes ismertetését megtaláljuk.

Új iptables modulok telepítése

Hogy a POM modult hogyan adjuk hozzá a rendszermaghoz, illetve hogyan vegyük használatba az iptables eszközöket, a ➲ www.lowth.com/howto/add-iptables-modules.php oldalon olvashatunk.

Összefoglalás

Láttuk, hogy a linuxos Netfilter számos nagyszerű szolgáltatással segít bennünket rendkívül hatékony tűzfalat építeni, ám a legtöbb Linux-terjesztés a szükséges kiegészítések jelentős részét alapesetben nem tartalmazza.
A Patch-o-matic gyűjtemény segítségével a rendszergazdák megfoltozhatják a rendszermagot, és kibővíthetik tűzfaluk alapszolgáltatásainak körét.

Linux Journal 2004. április, 120. szám

Chris Lowth az Intercai Mondiale (www.intercai.co.uk), egy angol távközlési, informatikai és üzleti tanácsadó cég munkatársa. Biztonsági programokat és hálózatfelügyeleti (OSS) megoldásokat tervez, néha megpróbálkozik a gitározással is. Chris a chris@lowth.com címen érhető el.