Biztonság a helyi hálózaton: IP Tables

Mick azt fejtegeti, hogyan kell az IP Tableshez kapcsolódó tulajdonosegyeztetési kiterjesztést (Owner Match Support) használni ahhoz, hogy megakadályozzuk a felhasználók hálózati protokollokkal kapcsolatos visszaéléseit.

A legtöbbünk talán határozottan tűzfaleszközként gondol az IP Tablesre, amellyel a külső támadókat sarokba lehet szorítani. De vajon tudta-e mindenki, hogy ezzel a helyi felhasználók tevékenységének is határt lehet szabni? A kísérleti tulajdonosegyeztetés új lehetőségeket teremt az IP Tables számára, amelyekkel meg lehet akadályozni, hogy a helyi felhasználók mások hálózati folyamatain keresztül csomagokat küldjenek.

A példa kedvéért tételezzük fel, hogy a rendszergazdai azonosító (`root`) egyik `cron` által ütemezett munkája az Stunnel programot használja arra, hogy egy távoli ***rsync***-folyamat számára állományokat küldjön. Amíg az alagút nyitva van, ezt bármelyik helyi hálózaton levő felhasználó használhatja a távoli rsync-kiszolgálóhoz való hozzáférésre. Az IP Tables a segítségedre siethet az említetthez hasonló hálózati élősködés megelőzésében. Írásunk pontosan azt fogja bemutatni, hogyan is teszi ezt.

A kérdés

Az alagút-segédprogramok a rendelkezésre álló biztonsági eszközök egyik legfontosabb új csoportját alkotják. Lehetővé teszik a számunkra az olyan megbízhatatlan szolgáltatások burkolását, mint amilyen a Telnet, az IMAP és a POP3 – titkosított látszólagos alagutakban átláthatóan és hatékonyan. Már hosszasan írtam ezeken az oldalakon a Secure Shellről (SSH) és annak foglalattovábbító képességeiről; az Stunnel és az SSL Wrap is ehhez hasonló, szintén erre a célra szolgáló ingyenes eszköz a Linux-rendszerben.

Vajon mi történik, ha egy ilyen alagutat telepítesz egy többfelhasználós rendszerbe? Mi lesz képes megállítani a meghatalmazással nem rendelkező helyi hálózaton levő felhasználókat abban, hogy a saját forgalmukat átküldjék ezen az alagúton? Mostanáig gyakorlatilag semmi sem tudta ezt megakadályozni. Minthogy az alagút-segédprogramok többsége úgy működik, hogy új figyelőkaput hoz létre – például: `localhost:992` – az alagút innenső oldala számára, rendszerint csak a kiszolgálóalkalmazástól függ az alagút túloldalán, hogy hitelesíti-e felhasználókat.

Tegyük fel, hogy az Stunnelt használom SSL-alagút felállítására a saját ***crueller*** nevű rendszeremtől a távoli ***strudel*** rendszerig, amelyen Telnetet szeretnék futtatni. Hagyjuk figyelmen kívül, hogy ez a fajta tevékenység SSH-val egyszerűbb lenne; most valamilyen oknál fogva nem szeretném helyileg telepíteni az SSH-t. A távoli gépen, ami az `inetd`-n keresztül a TCP 23-as kapuján át már futtatja a Telnet démont, az Stunnelt démon módban futtatom a következő paranccsal:

```
stunnel -d 992 -r localhost:23
➥-p /etc/stunnel/strudel.pem
```

A helyi gépen az Stunnelt ügyfélüzemmódban fogom futtatni, és a helyi gép TCP-protokolljának 992-es kapuján fog figyelni, viszont a kapcsolatot az ***strudelen*** lévő TCP 992-es kapujára az alábbi paranccsal továbbítja:

```
stunnel -c -d 992 -r strudel:992
```

Amennyiben még sohasem használtad az Stunnelt, és ezek a parancsok semmit sem mondanak neked, még ne kezdj el aggódni. Ebből annyi lényeges, hogy ebben a példában a ***cruellerről*** az ***strudelre*** a Telnettel való bejelentkezéshez a ***crulleren*** az alábbi parancsot kell használnom:

```
telnet localhost 992
```

Ezen a ponton az ***strudel*** kérni fogja a felhasználói azonosítómat és a jelszavamat, de bejelentkezési hitelesítéseimet az Stunnel a Telnettől eltérően titkosítani fogja, és nem a megszokott szövegként lesznek továbbítva. A távoli gépen működő Stunnel-folyamat a csomagokat vissza fogja fejteni, és át fogja őket adni a Telnet-folyamatoknak. Ez a folyamat a hitelesítési szakaszban nemcsak egyszer fog lezajlani, hanem a teljes Telnet-művelet során végig ismétlődik. Az Stunnel a teljes tranzakció során mindkét fél számára közvetítőként dolgozik, mégpedig olyan közvetítőként, amelynek nincs tudomása az alagútba juttatott adatok mibenlétéről és nem is foglalkozik velük, feltéve, hogy azok TCP alapúak.

Ez eddig szép és jó: rendelkezem titkosítással, ami az Stunnel nélkül nem létezne, és szerény mértékű hitelesítéssel – magának a Telnetnek a természetéből adódóan. Most már csak az okoz fejtörést, hogy bármelyik ***cruelleren*** levő felhasználó képes a helyi gép TCP 992-es figyelő kapujára telnetelni, és bejelentkezni az ***strudelre***. Az is aggaszthat, hogy valaki máris megpróbálja a ***strudelre*** való belépéshez használt jelszavamat kitalálni. Kezdjük ott, hogy vajon hogyan lehet ezeket az embereket megállítani abban, hogy mindenféle csomagot küldözgessenek keresztül az alagúton?

Íme a válasz: az IP Tables programmal és annak tulajdonosegyeztetési kiterjesztésével.

Az eszköz

Az IP Tables tulajdonosegyeztetési kiterjesztése négy ellenőrzési szemponttal egészíti ki az `iptables` parancsot:

`--uid-owner UID`: egyezteti a csomagokat, amelyek az adott felhasználói azonosítóval (UID) jöttek létre

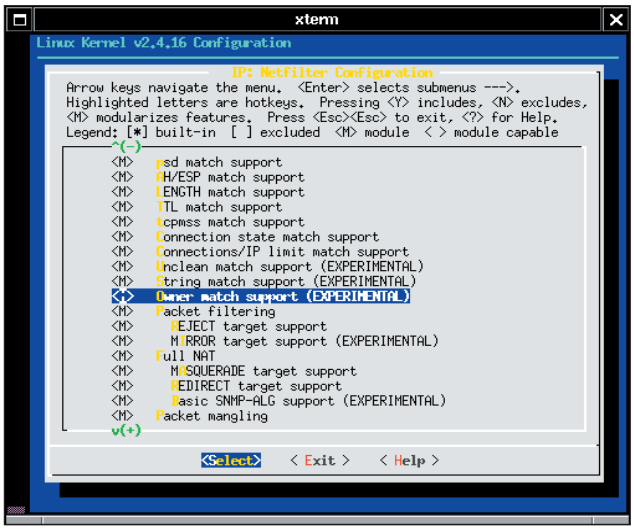
`--gid-owner GID`: egyezteti a csomagokat, amelyek az adott csoportazonosítóval (GID) jöttek létre

`--pid-owner PID`: egyezteti a csomagokat, amelyek az adott folyamatazonosítóval (PID) jöttek létre

`--sid-owner SID`: egyezteti a csomagokat, amelyek az adott szezonazonosítóval (SID) jöttek létre

A négy közül mostani céljaink szempontjából az első két adat fontos. Nem biztos, hogy a tulajdonosegyeztetési kiterjesztés szerepel a Linux-változatodban működő rendszermagban. Ha az IP Tables fejlesztőcsoportja nem is feltétlenül, de a Linux-rendszermag fejlesztőcsapata kísérleti szolgáltatásnak tekinti, ezért ennek rendszermagba történő fordítását magadnak kell elvégezned. A program forráskódja ezzel szemben a 2.4-es rendszermag kódalapjának része, és emiatt bármilyen friss Linux-változat rendszermagjának forrásprogramcsomag-

jával (2.4.x) könnyen kivitelezhető.
A rendszermag újrafordításakor több jellemzőt közvetlenül be kell állítani. Először is a ***Kódérettségi szint*** lehetőségnél válaszd a ***Prompt for development and/or incomplete code/drivers*** (Fejlesztés alatt álló rendszer és/vagy befejezetlen programkód/meghajtók). Ezután a hálózati protokolloknál az egyénileg kiválasztott hálózati protokollokon és szolgáltatásokon kívül ellenőrizd, hogy a ***Network Packet Filtering*** lehetőség ki legyen választva. Ez be fogja kapcsolni az IP-részcsoportot: a Netfilter beállítását, amelyet az *1. képen* tekinthetünk meg. Ezeket a lehetőségeket – csillaggal való kijelöléssel – befordíthatjuk a



1. kép Rendszermag fordítása tulajdonosegyeztetési kiterjesztéssel

rendszermagba, vagy modulokká is fordíthatjuk őket (az M betűvel). A legtöbb ember modulként fordítja őket, egy időben ritkán van rájuk szükség. Az a modul, amelyik a jelen pillanatban a legfontosabb számunkra, az Owner Match Support (tulajdonosegyeztetési kiterjesztés).
A Linux-rendszermag fordításáról, telepítéséről és moduljainak hátralevő részéről másutt nagyon jó leírást olvashatunk, nevezetesen a rendszermag forráskódjának README állományában. Amint végeztél a rendszermag fordításával, telepítésével, és a gépet is újraindítottad, máris használhatod vadonatúj tulajdonosegyeztetési modulodat, ami az `ipt_owner` nevet fogja kapni. A modul betöltéséhez használd a `modprobe` parancsot:

```
modprobe ipt_owner
```

A gyakorlatban az IP Tables-szabályokat talán a */etc/init.d/inetd* héjprogramból szeretnéd indítani. Ha az IP Tablest valóban így kívánod használni, a fenti `modprobe` parancsot szúrd be a héjprogram elejére, pontosabban bármilyen tulajdonosegyeztetést felhasználó IP Tables-parancs elé. Fontos, hogy a tulajdonosegyeztetési kiterjesztést sem a Bastille-Linux önműködő tűzfalbeállítása, sem a SuSE Linux SuSE tűzfal-héjprogramja jelentősebb módosítás nélkül nem támogatja. Ez nem is meglepő, hiszen ezeket és más egyszerű Internet csomagszűrő eszközöket elsődlegesen alacsony szintű internetes védelmi célokra szánták, semmint a felhasználói jogok haladó szintű szabályozására. Az utóbbihoz az IP Tables-szabályokat magadnak kell megalkotnod.
Ezek után térjünk vissza a ***crueller*** és az Stunnel ügyfélprogramhoz. Tételezzük fel, hogy a ***crueller*** rendszermagját az `ipt_owner` modullal fordították, a modult a `modprobe` utasítással töltötték be; minthogy az IP Tables egyelőre még nincs beállítva, semmilyen csomagszűrés nem történik.
Tegyük fel azt is, hogy az Stunnel csatlakozón keresztüli Telnet-használatot kizárólag a rendszergazdára szeretnénk korlátozni, ahogyan azt a cikk elején elterveztük. Talán még nem felejtetted el, hogy a ***cruelleren*** az Stunnel alagútprogram számára a 992-es TCP-kapun figyelést állítottunk be, ami a csomagokat az ***strudel*** ugyanilyen TCP-kapujára titkosítja és továbbítja. Amennyiben a ***crueller*** nem tűzfalnak lett beállítva, a kimeneti lánc (OUTPUT) számára az „alapértelmezés szerinti elfogadás” alapelvvel dolgozhatunk. Tűzfalakon viszont a láncoknak általában a „mindent” vagy a „mindent visszautasít” alapelv szerint kell működniük. Az egygépes kiszolgálóknál – vagyis amelyekbe csupán egy kártya van szerelve – a bástyakiszolgálókon a kimenő forgalmat illetően néha az engedékenyebb hozzáállás is megengedett. Amennyiben a ***cruelleren*** éppen ez a helyzet, a megkívánt korlátozáshoz egyetlen szűrési szabály használata is elegendő:

```
iptables -A OUTPUT -p tcp --dport 992
➥-d localhost -m owner ! --uid-owner root
➥-j REJECT
```

Boncolgassuk ezt a parancsot mezőnként:
- `A OUTPUT`: ez jelzi az IP Tables számára, hogy az OUTPUT lánc végére új szabályt szeretnénk fűzni.
`-p tcp`: arra utasítja az IP Tablest, hogy csak TCP protokollú csomagokat fogadjon, és töltse be az IP Tables TCP-lehetőségeit.
`--dport 992`: ez a TCP-re jellemző lehetőség megparancsolja az IP Tablesnek, hogy csak a 992-es kapura irányuló TCP-csomagot fogadja.
`-d localhost`: azt parancsolja az IP Tablesnek, hogy csak a localhostnak küldött csomagokat fogadja el.
`-m owner`: előírja az IP Tables számára a tulajdonosegyeztetési kiterjesztés betöltését.
`! --uid-owner` root: megtiltja az IP Tables számára, hogy a rendszergazdához tartozó folyamatok által létrehozott csomagokon kívül más csomagot is elfogadjon.
`-j REJECT`: az IP Tablestől azt követeli meg, hogy az ebben a sorban megadott összes szempontnak megfelelő csomagot utasítsa vissza.
Röviden: ez a szabály az IP Tablesen keresztül megmondja a rendszermagnak, hogy a sajátgép 992-es TCP-kapujához érkező csomagokat ejtse el, hacsak nem a rendszergazda valamelyik folyamata küldte azokat.
Képzeljük most el, hogy a ***crueller*** házirendje inkább az elővigyázatos „elvetés”-hez áll közelebb, semmint az „elfogadás”-hoz. Az alapértelmezés szerinti elvetés a legtöbb IP Tables-telepítésnél előnyben részesítendő. A legkevesebb előjog alapelve az egyik legfontosabb fogalom az adatbiztonság-technikában: ami nincs félreérthetetlenül engedélyezve, az tilos!
Nekünk azonban most egy hosszabb kimeneti láncra van szükségünk. Egy üres láncból kiindulva: meg kell mondanunk az IP Tablesnek, hogy a már elfogadott szezonokhoz tartozó csomagokat továbbítsa:

```
iptables -I OUTPUT 1 -m state --state
➥ESTABLISHED,NEW -j ACCEPT
```

A `-state` egyeztetési kiterjesztés az IP Tablest állapotellenőrző képességekkel ruházza fel, amelyek révén az IP Tables a csomagokat képes az érvényes szezonok és adatfolyamok viszonylatában kiválogatni. A szolgáltatás – önmagáért való értelmén és szépségén túl – nagymértékben csökkenti az egy-egy művelethez meghatározandó szabályok számát. Az állapotkövetés nélkül egy helyett két szabályra lenne szükség, például egy kifelé irányuló Telnet-műveletnél egyre a „bemenő” és egyre a „kimenő” láncban. Ezért is jó lenne a fenti szabályt szinte

mindig, bármilyen olyan lánc felett alkalmazni, amelynek működési alapleve a „csomagok elejtése (DROP)”.
Ezután az Stunnel számára lehetővé kell tennünk az ***strudelhez*** történő kapcsolódást:

```
stunnel -A OUTPUT -p tcp -dport 992
-d strudel -j ACCEPT
```

Ez a parancs a „kimenő” lánc végére új szabályt fűz, ami a kimenő kapcsolatoknak megengedi, hogy a TCP 992-es kapuján keresztül a strudelhez kapcsolódjanak.
Végül kiadunk egy parancsot, amely az „alapértelmezés szerinti elfogadás”-hoz hasonlít, azzal a különbséggel, hogy egyrészt ez a célgépnél a visszautasítás helyett az elfogadást célozza (`ACCEPT`), másrészt a `--uid-owner` lehetőségnél a tagadó szerepet betöltő felkiáltójel is hiányzik:

```
iptables -A OUTPUT -p tcp --dport 992 -d
➥localhost -m owner --uid-owner root -j ACCEPT
```

Vegyünk még egy példát! Az ***.rsync*** nagy teljesítményű állományátviteli segédprogram, ami képes az állományok közti eltérések alapján dolgozni. Képes egy távoli gépen lévő állományt a helyi gépen szereplővel összehasonlítani, így csak az eltérő részleteket kell letölteni. Az ***.rsync***-et az SSH-val együttműködve használhatjuk, és igen, kitaláltad, az Stunnel programmal együtt!
Induljunk ki abból, hogy a ***cruelleren*** működik egy `cron` által ütemezett alkalmazás, amely az rsynckel a ***strudelen*** levő ***stuff.txt*** állománynak és a helyi gépen tárolt másolatának az összehasonlítását végzi. Tételezzük fel azt is, hogy az említett állomány érzékeny adatokat tartalmaz, ezért az állománytovábbítások titkosítására az Stunnelt használjuk. Kizárólag a helyi rendszergazdáknak – akik mindnyájan a „wheel” csoport tagjai – kell tudniuk szabályozni a héjprogram működését vagy használni az alagutat.
Az ***strudelen*** az `rsync` démon módban működik, és úgy lett beállítva, hogy egy modult, egy virtuális kötetet megosztva használjon, ennek neve: ***attic***. Feltéve, hogy a ***/etc/rsyncd.conf*** helyesen lett beállítva, az rsyncet démon üzemmódban futtató parancs igen egyszerű:

```
rsync --daemon
```

Az Stunnel a TCP-n ugyancsak rendelkezik a 273-as figyelő kapuval, ami az ***rsync***-folyamathoz titkosítja és továbbítja a forgalmat, ez viszont a TCP 873-as kapuján figyel. Az ***strudelen*** az Stunnelt ilyen módon futtató parancs a következő:

```
stunnel -d 273 -r localhost:873
➥-p /etc/stunnel/strudel.pem
```

A ***cruelleren*** a megfelelő ügyfél-üzemmódú Stunnel-figyelőkaput az alábbi módon kellene megszólítani:

```
stunnel -c -d 273 -r strudel:273
```

Minden rendben: beüzemeltünk egy alagutat, amivel a ***cruelleren*** levő TCP 273-as kapuján keresztül küldött csomagok titkosítva lesznek, és az ***strudel*** TCP-jének 273-as kapujához lesznek továbbítva, ahol visszafejtésre kerülnek, és az ***strudel*** helyi ***rsync***-folyamatához a TCP 873-as kapuján át továbbítódnak.
Ha a ***crueller***-en lévő közönséges felhasználó, az akadályt jelentő IP Tables-szabályok távollétében, most megpróbálná használni az alagutat, erőfeszítése sikerrel járna:

```
rsync --port=273 -v localhost::attic/stuff.txt .
stuff.txt
wrote 508 bytes  read 575 bytes  2166.00 bytes/sec
total size is 48188  speedup is 44.49
```

Ha azonban a cruelleren érvényesítjük azt az IP Tables-szabályt, amely az ***rsync***-alagút használatát a „wheel” csoport tagjaira korlátozza:

```
iptables -A OUTPUT -p tcp -d localhost
➥--dport 272 -m owner ! --gid-owner wheel
➥-j REJECT
```

A közönséges felhasználónak az a kísérlete, hogy elcsenje a ***stuff.txt*** állományt, most már kudarcba fog fulladni:

```
rsync --port=273 -v localhost::attic/stuff.txt .
rsync: failed to connect to localhost:
     Connection refused
rsync error: error in socket IO (code 10)
     at clientserver.c(97)
```

Amennyiben viszont a wheel csoportba tartozó admin7 próbál meg a kiszolgálógéphez kapcsolódni, sikerül csatlakoznia:

```
rsync --port=272 -v localhost::chumly/stuff.txt .
stuff.txt
wrote 508 bytes  read 575 bytes  2166.00 bytes/sec
total size is 48188  speedup is 44.49
```

Remélhetőleg feltűnt, hogy ez „alapértelmezés szerinti elfogadás”-t feltételez. Ha viszont a „kimenet” a „mindent elvet” alapelv alapján működik, egy olyan szabályra lenne szükségünk a kimeneti láncon, ami lehetővé tenné a kimenő kapcsolatot a strudel TCP-jének 273-as kapujával. A „kimenet” láncnak ugyancsak egy megengedő létező, illetve kapcsolódó szezonszabállyal kell kezdődnie. Mivel mindkét szabály erősen emlékeztet az előző példánkban szereplőkre, mellőzni fogom a bemutatásukat.

A tulajdonosegyeztetési kiterjesztéshez és az Stunnelhez kapcsolódó vegyes megjegyzések

Mint látható, a `--uid-owner` és a `--gid-owner` használata meglehetősen egyszerű. Még nem említettem, de a példában már bemutattam, hogy mindkét módszer megengedi nevek és számszerű azonosítók használatát.
A másik fontos téma, amivel most foglalkoztam, a TCP-burkolókhoz hasonló hozzáférés-szabályozás. Bármilyen TCP-burkolót használó rendszer – vagy olyan rendszer, amelynek Stunnel binárisát a `libwrapper`-támogatással fordították – ***/etc/host.allow*** állományát ki kell egészíteni megfelelő bejegyzésekkel, amelyek azon az adott gépen az Stunnelt képessé teszik feladata ellátására: akár ügyfélmódban, akár démonként. Ez tulajdonképpen jó dolog; ahelyett, hogy újabb akadályt látnál benne az Stunnel számára, inkább úgy gondolj rá, mint a biztonságot jelképező hagyma egyik újabb rétegére.
Végül rád hagyom, kedves olvasóm, a `--pid-owner`, `--sid-owner` lehetőségekkel való szösszmötölést. Adnék azonban egy tanácsot. Sok démon szülő-PID-jeit az indulás során meghatározott helyre teszi, nevezetesen a ***/var/run/ssh.pid***-be. Az IP Tables indulása során beolvasott ehhez hasonló PID-állomány segít beazonosítani az egyes folyamatoktól származó csomagokat. Szerencse fel!

Linux Journal 2002. augusztus, 100. szám

Mick Bauer
(mick@visi.com) hálózati biztonsági tanácsadó az Upstream Solutions Inc.-nél Minneapolisban (Minnesota). Mick a szerzője a hamarosan megjelenő új O'Reilly-könyvnek, amelynek címe „Building Secure Servers With Linux”.

Kapcsolódó címek

A Netfilter weblapja ➲ http://netfilter.samba.org
Az rsync weblapja ➲ http://rsync.samba.org