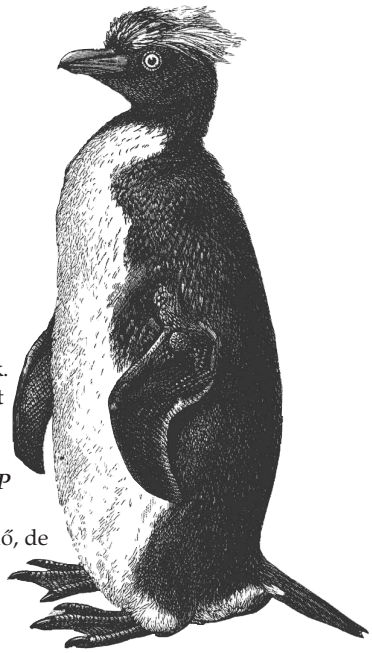
A syslog beállítása

Ha biztos akarsz lenni benne, hogy a rendszerfolyamatok és a fontos alkalmazások naplózzák az eseményeket és az állapotukat, mindenképpen meg kell ismerned a syslogot.



Sokat tehetsz Linux-rendszered biztonságáért, amelynek alapja a mindent magába foglaló, pontos és gondosan figyelt napló. A naplók többféle célt szolgálnak: először is segítenek a hibakeresésben – gyakorlatilag az összes elképzelhető gond esetén, amely a rendszerben vagy az alkalmazásokban keletkezik. Másodszor idejében felhívják a figyelmünket, ha rendszerünk használatával visszaéltek. Harmadszor, ha minden kötél szakad (ez azt jelenti, hogy vagy a rendszer omlik össze, vagy betörtek hozzánk), a naplók létfontosságú bizonyítékokat tartalmazhatnak.

Ez a cikk arról szól, miként naplózhatjuk a minket érdeklő rendszerfolyamatokat, és arról, hogy a fontos alkalmazások naplózzák az eseményeket és állapotukat. Kitűzött célunkat elérhetjük a jól bevált syslog programmal. A syslog adatokat vesz át a rendszermagtól (a `klogd`-n keresztül) vagy egy tetszőleges helyi folyamattól, de akár távoli rendszereken futó folyamatoktól is. Rugalmas, mivel megadhatjuk, hogy mit és hova naplózzon. Az előre beállított syslogtelepítés gyakorlatilag az összes Unix- és Linux-változatban része az alap operációs rendszernek.

Ebben a hónapban teljes mélységében tárgyaljuk a syslog beállítását és használatát, talán annál is részletesebben, mint amennyire elvárod tőlünk. Tapasztalatom szerint ugyanis a Linux-felhasználók túlnyomó többsége, még a rendszergazdák is a syslogot hajlamosak az alapértelmezett beállításokkal használni, esetleg apró módosításokat hajtanak végre. Ez azonban ritkán nevezhető jó ötletnek. Elöljáróban még megemlítenék néhány fontos dolgot: ha valakit tényleg érdekel a részletes és rugalmas naplózás, érdemes kipróbálnia ***Scheidler Balázs syslog-ng*** (syslog new generation) nevű kitűnő programját, amely közel annyira sem elterjedt, mint a syslog. A ***syslog-ng*** programról további ismereteket a ***Kapcsolódó címek*** részben található hivatkozás nyújt.

A syslog beállítása

Minden alkalommal, amikor a `syslogd`, azaz a syslogdémon naplóüzenetet kap, az üzenet típusának és fontosságának megfelelően cselekszik. A */etc/syslog.conf* fájl adja meg az üzenetek típusát és fontosságát, valamint a tevékenységek közötti összefüggést. A fájl minden sora egy vagy több típushoz, illetve fontosságkijelölőhöz tartozó tevékenységet ad meg. A kijelölő egy vagy több típusból és egy fontosságértékből áll.

A következő ***syslog.conf*** fájlban a `mail.notice` a kijelölő és a `/var/log/mail` a tevékenység (azaz „írd ki az üzenetet a */var/log/mail* fájlba”):

```
mail.notice   /var/log/mail
```

A kijelölőn belül a `mail` a típus (üzenetkategória), és a `notice` a fontosság szintje.

Típusok

A típusok egyszerű kategóriák. A Linux a következő típusokat támogatja: ***auth***, ***authprov***, ***cron***, ***démon***, ***kern***, ***lpr***, ***mail***, ***mark***, ***news***, ***syslog***, ***user***, ***UUCP*** és ***local0***-tól ***local7***-ig. Ezek közül néhány magától értetődő, de a következőkről érdemes pár szót ejteni:

- ***auth***: számos rendszerbiztonsággal kapcsolatos esemény használja,
- ***authpriv:*** hozzáférés szabályozásával kapcsolatos üzenetek esetében használatos,
- ***démon:*** rendszerfolyamatok és más démonok használják,
- ***kern:*** a rendszermag üzenetei esetében használatos,
- ***mark***: a syslog által létrehozott üzenetek, amelyek csak az időbélyeget és a `--MARK--` karakterláncot tartalmazzák; a két ilyen jelölés között eltelt percek száma a `syslogd -m [minutes]` kapcsolójával befolyásolható,
- ***user:*** az alapértelmezett típus, ha az alkalmazás vagy a kijelölő nem ad meg mást,
- ***local7:*** a rendszerindítás üzenetei,
- ****:*** helyettesítő, jelentése – „bármely típus”,
- ***none:*** helyettesítő, jelentése – „semelyik típus”.

Fontossági sorrend

A típusok semmilyen viszonyban ncsenek egymással, ezzel ellentétben a fontosságnak sorrendje van. A Linuxban a következő fontossági szinteket különböztetjük meg (sürgősségi sorrendben): ***debug***, ***info***, ***notice***, ***warning***, ***err***, ***crit***, ***alert*** és ***emerg***. Jegyezd meg, hogy egy adott üzenet sürgősségét a programozó határozza meg; a típust és a fontosságot az üzeneteket létrehozó programok állítják be, nem a syslog.

Akárcsak a típusoknál, a `*` és a `none` helyettesítők itt is használhatók. Egy kijelölőn belül csak egy helyettesítő vagy fontosság adható meg. A fontosság előtt szerepelhet az `=` és a `!` módosítók egyike vagy mindkettő.

Amennyiben a kijelölőben egyetlen fontosságot adsz meg (módosítók nélkül), akkor valójában nemcsak azt a fontossági szintet adod meg, hanem a felette lévőket is. Például a `mail.notice` azt jelenti, hogy „az összes ***mail***-típusú üzenet, amelynek a fontossága ***notice*** vagy magasabb”, azaz a fontosság ***notice***, ***warning***, ***err***, ***crit***, ***alert*** vagy ***emerg***.

Ezt a viselkedést a fontosság elé írott `=` (egyenlőslgjel) megszünteti. A `mail.=notice` kijelölő azt jelenti, hogy „az összes ***mail***-típusú üzenet, amelynek fontossága ***notice***”. A fontosságokat negálni is lehet: a `mail.!notice` jelentése: „az összes ***mail***-üzenet, kivéve a ***notice*** fontosságúak”.

Tevékenységek

A gyakorlatban a legtöbb naplóüzenet fájlba íródik. Ha egy fájl teljes elérési útját megadod a sor tevékenység részében a ***syslog.conf***-ban, akkor a sorra illeszkedő üzenetek hozzáíródnak ahhoz a fájlhoz (ha a fájl nem létezik, a syslog létrehozza). A fenti ***syslog.conf*** sorban arra utasítottuk a syslogot, hogy a megfelelő üzeneteket küldje a ***/var/log/mail*** fájlba.
Az üzeneteket más helyekre is küldheted. A művelet lehet fájl, névvel ellátott csővezeték, eszközfájl, távoli gép vagy egy felhasználó képernyője. A csővezetékeket általában hibakeresési célokra használják. A leggyakrabban használt eszközfájlok a TTY-k, de sokan szívesen küldik a biztonságot érintő adatokat a ***/dev/lp0***-ra, azaz a helyi sornyomtatóra. A kinyomtatott naplót a betörő nem tudja letörölni vagy megváltoztatni, és ez a szerep végre értelmet ad a régi mátrixnyomtatók életének.
A távoli naplózás képessége a syslog egyik leghasznosabb tulajdonsága. Ha a sor tevékenység részében egy `@` (atjellel) bevezetett IP-címet vagy gépnevet adsz meg, a syslog a megfelelő üzeneteket elküldi erre a távoli gépre. Például a

```
*.emerg                 @mothership.mydomain.org
```

sor a `syslogd`-t arra utasítja, hogy minden ***emerg*** fontosságú üzenetet a ***mothership.mydomain.org*** nevű gépre küldjön.
Jegyezd meg, hogy a távoli gépen (esetünkben a mothershipen) a `syslogd` folyamatot a `-r` kapcsolóval kell elindítani, hogy fogadja naplóüzeneteidet, ugyanis a `syslogd` alapértelmezés szerint semmilyen üzenetet nem fogad el távoli rendszerről.
Amennyiben központi naplókiszolgálót szeretnél működtetni, amit nagyon ajánlok, valamilyen módon befolyásolnod kell a bejövő üzenetek hozzáférési jogait. Használnod kell legalább a TCP-burkoló gépelérés-szabályozását (a forrás IP-címe szerint), vagy akár helyi tűzfalszabályokat (IP Chains vagy IP Tables).

Bonyolultabb kijelölők

Egyetlen ***syslog.conf*** kijelölőben több típust is felsorolhatsz vesszővel elválasztva. Bővítsük ki az eredeti ***syslog.conf*** sorunkat, hogy ne csak a ***mail-***, hanem az UUCP-üzeneteket is magába foglalja (továbbra is a ***notice*** és az afeletti fontosságúakat):

```
mail,uucp.notice            /var/log/mail
```

Ez a fontosságra nézvést nem működik. Ne feledd, egy adott kijelölőben csak egyetlen fontosság vagy fontosséghelyettesítő szerepelhet!
Egy sorban viszont több kijelölőt is meg lehet adni, pontosvesszővel elválasztva. Ha egy sor több kijelölőt is tartalmaz, a kiértékelés balról jobbra halad; az általános kijelölőket kell előre venned, utánuk következzenek a különlegesebbek. Gondolj úgy a kijelölőkre, mint a szűrőkre. Az üzenet balról jobbra haladva jut át a soron, először a durvább, majd a finomabb szűrőkön megy keresztül.
Folytatva egysoros példánkat, tegyük fel, hogy továbbra is a ***mail***- és UUCP-üzenetekre vagyunk kíváncsiak, amelyeket a ***/var/log/mail*** fájlba akarunk kiíratni, de ki szeretnénk zárni az ***alert*** fontosságú UUCP-üzeneteket. Sorunk ezután így néz ki:

```
mail,uucp.notice;uucp.!alert      /var/log/mail
```

Valójában a syslog viselkedése nem annyira kiszámítható, mint ahogy a példa sugallja. Ellentmondó kijelölők használatakor, vagy ha a különleges kijelölő megelőzi az általánost, váratlan eredményeket kaphatunk. Ezért pontosabb, ha úgy fogalmazunk, hogy a legjobb eredmény érdekében az általános kijelölőkkel kezdd a sort, majd a kivételekkel (és/vagy más különleges kijelölőkkel) folytasd.
Amennyire lehetséges, ne bonyolítsd túl a dolgokat. A `logger` parancs használatával ***syslog.conf*** szabályaidat kipróbálhatod (olvasd el „A rendszernaplózás kipróbálása a `logger` parancscsal” című részt a cikk vége felé).
Figyeld meg, hogy a második kijelölőben (`uucp.!=alert`) a fontosság előtt a `!=` előtagot használtuk, ami azt jelentette, hogy „nem egyenlő”. Ha ki szerettük volna zárni az összes ***alert*** és nála nagyobb fontosságú UUCP-üzenetet (azaz ***alert***-et és ***emerg***-et), az = jelet elhagytuk volna:

```
mail,uucp.notice;uucp.!alert      /var/log/mail
```

Mi történik egy ***info*** fontosságú UUCP-üzenettel? Illeszkedik a második kijelölőre, tehát a ***/var/log/mail***-be kellene naplózódnia, igaz? A fenti példák alapján azonban nem ez fog történni. Mivel a sor első kijelölője csak a ***notice*** és az annál fontosabb ***mail***- és UUCP-üzeneteket engedi tovább, a kérdéses üzenet nem fog eljutni a második kijelölőig.
Természetesen semmi sem akadályozza meg, hogy létrehozz egy másik sort, amelyben az ***info***-szintű UUCP-üzenetekkel foglalkozol. Akár több ilyen sorod is lehet, ha úgy akarod.
A tűzfalszabályokkal ellentétben itt a ***syslog.conf*** összes során minden naplóüzenet végigfut, és annyi műveletet hajt végre, ahányszor illeszkedik.
Tegyük fel, azt szeretnénk, hogy az ***emerg***-szintű üzeneteket minden bejelentkezett felhasználó megkapja, és ezenkívül a megfelelő alkalmazások naplóiba is bekerüljenek. Az *1. listán* olvashatókhoz hasonló megoldással célt érhetünk. Figyeld meg a `-` (mínuszjel) a fájlba író műveletek előtt – ez arra utasítja a `syslogd`-t, hogy az adott naplófájlt a sorra illeszkedő üzenet kiírása után ne hangolja össze.
Az összehangolás elhagyása növeli a hibák lehetőségét: a naplóüzenetek esetleg töredékesen vagy sehogysem kerülnek be

1. lista syslog.conf példafájl

```
# PØlda syslog.conf fÆjl, amely az  zeneteket
# a mail, kernel Øs  egyØb  csoportokba
# osztja, Øs az emerg-fontossÆgœakat
# minden bejelentkezett felhasznÆl
kØpernyıjØre ki rja

# a legt bb rendszeresemØny a tty10-re
# Øs az xconsole csıbe  r dik, az emerg
# mindenkinek

kern.warn;*.err;authpriv.none
|/dev/console
*.emerg                                 *

# a mail, news (leginkÆbb) Øs rendszermag-,
# illetve tßzfal zenetek a megfelelı
# napl fÆjlba ker ljenek
mail.*      -/var/log/mail
kern.*      -/var/log/kernel_n_firewall

# a t bbit ments k egyetlen fÆjlba
*.*;mail.none      -/var/log/messages
```

***1. táblázat* – A syslog.conf használatának és értékeinek összefoglalása**

Típusok	Típusok kódja**	Fontosság (növekvő sorrendben)	Fontosság kódjai	Műveletek	
auth	4	none	n/a	/valami/fajl	*(naplózás a megadott fájlba)*
auth-priv	10	debug	7	~/valami/fajl	*(naplózás a megadott fájlba, de nem hangol össze írás után)*
cron	9	ifo	6		
daemon	3	notice	5	/valami/csovezeték (pipe)	*(naplózás a megadott csővezetékbe)*
kern	0	warning	4	/dev/valami/tty_vagy_konzol	*(naplózás megadott konzolra)*
lpr	6	err	3	@tavoli.gepnev vagy IP-cím	*(naplózás a megadott távoli gépre)*
mail	2	crit	2	felhasznalo1, felhasznalo2 stb.	*(naplózás a felhasználó képernyőjére)*
mark	n/a	alert	1		
news	5	emerg	0		
syslog	7	* (minden fontosság)	n/a		
user	1				
uucp	8				
local	(0–7)16–23				
* *(minden típus)*	n/a				

A „!” és az „=” előtagok használata a fontosság előtt

*.notice = minden esemény, amely .notice vagy magasabb fontosságú
*.!notice = egyik esemény sem, amelyik .notice vagy magasabb fontosságú
*.=notice = csak a .notice fontosságú esetek
*.!=notice = nem nézi a .notice fontosságú eseteket

** A számkódokat *syslog.conf*-ban nem szabad Linux-rendszereken használni. Csak a tájékoztatás kedvéért adtuk meg őket, például ha syslog-üzeneteket szeretnénk küldeni a naplókiszolgálóra, és esetleg nem linuxos syslogdémont kell beállítanod, amely csak a számadatokat ismeri, mint például a Cisco IOS.

Mi az a klogd?

Létezik egy démon, amelynek beállításait valószínűleg nem kell megváltoztatnod, de nem árt tudnod róla. Ez a `klogs`, a Linux-rendszermagjának naplózódémona. Ezt a démont rendszerindításkor ugyanaz a parancsfájl indítja el, mint az általános rendszernaplózót (ez lehet a */etc/init.d/syslogd* vagy a */etc/init.d/sysklogd* az általad használt Linux-terjesztéstől függően).
Alapértelmezés szerint a `klogd` minden rendszermagtól érkező naplóüzenetet a rendszernaplózónak továbbít, ezért a legtöbbünknek nem kell a `klogd` miatt aggódnia. A rendszermag üzeneteinek kezelését a `syslogd` beállításait tartalmazó fájl módosításával befolyásolhatod.
A `klogd` egyedülálló naplózóként is elindítható, azaz a rendszermag üzeneteit közvetlenül a konzolra vagy fájlba küldheti. Ráadásul, ha még nem fut démonként, a `klogd` arra is használható, hogy a rendszermag átmeneti naplótárolóit (azaz a legfrissebb rendszermagüzeneteket) kiírja fájlba vagy a képernyőre. A `klogd` ilyen alkalmazása főleg a rendszermag fejlesztői számára hasznos.
A legtöbbünknek elég annyit tudni, hogy a szokásos esetekben a `klogd`-ot nyugodtan békén lehet hagyni (azaz meghagyva az alapbeállításokat és az alapértelmezett indítási módot sem szabad letiltani). Csak annyira emlékezz, hogy ha a syslogot használod Linux alatt, akkor minden rendszermagüzenet először a `klogd`-n megy keresztül.

a fájlba, viszont csökkenti a lemez használatának gyakoriságát, ezért teljesítménynövelő hatású. Ahol gyakori fájlírási műveleteket vársz, azoknál a soroknál használd a - (mínuszjelet).
Az *1. listán* egy bizonyos hasznos ismétlődés látható. A rendszermag figyelmeztetései, valamint az összes ***error***- és annál magasabb szintű üzenet, kivéve az ***authpriv***-üzeneteket, az X-konzol ablakába íródnak. Minden ***emerg***-szintű üzenet nemcsak ide, hanem minden bejelentkezett felhasználó képernyőjére is kiíródik.
Továbbmenve minden ***mail***- és ***kernel***-üzenet a megfelelő naplófájlba íródik. Az összes egyéb üzenet – kivéve a ***mail***-üzeneteket – a */var/log/messages* fájlba íródik.
Az előző példákat abból az alapértelmezett ***syslog.conf*** fájljából vettem át, amelyet a SuSE 7.1 telepített az egyik gépemre. Miért nem felel meg ez az alapértelmezett beállítás? Minek ezt egyáltalán megváltoztatni?
Talán nem kell, azonban valószínűleg érdemes. Az alapértelmezett ***syslog.conf*** fájl a legtöbb esetben egy fontos üzenethez vagy olyan tevékenységet rendel, amely nem hatékonyan hívja fel a figyelmedet (például kiírja az üzenetet a TTY-konzolra, míg te a rendszert csak SSH-n keresztül éred el), vagy bizonyos üzenettípusokat az igényeidhez képest túlságosan alaposan vagy túl felületesen kezel.
Az *1. táblázat* összefoglalja a ***syslog.conf*** nyelvtanát, a típusok értékeit, a fontosság fokozatait és a műveletek fajtáit. Jegyezd meg, hogy a három fő oszlop független egymástól: semmilyen kapcsolat nincs a típusok, a fontosság és a tevékenységek között, azaz bármilyen típusú üzenet bármilyen fontossággal bírhat, és tetszőleges művelet hajtható végre rajta. Szintén lényeges, hogy a típusok és a fontosság számkódjai szigorúan a teljesség kedvéért vannak felsorolva, a ***syslog.conf***-ban ne

2. táblázat – **A syslogd indítási kapcsolói**

Kapcsoló	*Leírás*
`-m` (percek a jelek között)	Ennyi perc telik el két jelölőüzenet között (csak időbélyeget tartalmazó üzenetek, amelyek a naplófájlt nézőpontodtól függően áttekinthetővé vagy kuszává teszik. A 0 érték azt jelenti, hogy nincs jelölés).
`-a` (/további/foglalat)	Segítségével további foglalatok adhatók meg a */dev/log*-on kívül, ahonnan a `syslogd` üzeneteket fogad el.
`-f` (/utvonal/syslog.conf)	Ha nem a */etc/syslog.conf*-ot használjuk, a beállításokat tartalmazó fájl elérési útját itt kell megadni.
`-r`	Távoli gépek syslog-üzeneteinek figyelése.

használd őket. Esetleg találkozhatsz velük a forráskódokban vagy a hálózati forgalom fájlba mentett csomagjaiban.

A syslogd futtatása

Ahogy az alapértelmezett ***syslog.conf*** esetleg nem felel meg az igényeidnek, úgy a syslogd alapértelmezett elindítási módja is változtatásra szorulhat. A 2. *táblázatban* és a következő bekezdésekben néhány olyan `syslogd`-kapcsolót mutatunk be, amelyek különösen érdekesek a biztonság szempontjából, de a teljes listát a sysklogd(8) súgóoldalon láthatod.
Ráadásul arra is célszerű figyelmet fordítanod, hogy amikor a syslogd beállításait és indítási módját megváltoztatod, a `syslogd`-t és a `klogd`-t általában egyszerre kell elindítani és leállítani (ha nem tudod, mi a `klogd`, olvasd el a „Mi az a klogd?” széljegyzetet). Az a legjobb, ha ezeket úgy indítod és állítod le, ahogy a rendszered is teszi; azt javaslom, használd rendszered `syslogd`-, illetve `klogd`-indító parancsfájljait.
A legtöbb Linux-rendszeren ez az indító parancsfájl vagy a */etc/init.d/syslog* vagy a */etc/init.d/sysklog* (a sysklog a „syslog és klogd” rövidítése).
Az első bemutatandó `syslogd` kapcsoló az egyetlen, amelyet a Red Hat 7.x az alapértelmezett */etc/init.d/syslog* parancsfájlban használ, ez az `-m 0`, amely a jelölőüzeneteket tiltja. Ezek az üzenetek csak az időbélyeget és a `--MARK--` karakterláncot tartalmazzák. Ezt számos felhasználó hasznosnak tartja, ha hosszú naplófájlokban kell eligazodni, sokan viszont ellenszenvesnek és feleslegesnek gondolják, hiszen minden üzenetnek van időbélyege.
A jelölőüzenetek bekapcsolásához a `-m` után egy pozitív egész számot kell megadnunk, amely megmondja a `syslogd`-nek, hogy hány percenként küldjön magának jelölőüzenetet. Ne feledd, a jelölőüzenet más típusba tartozik, a ***mark***-ba. Legalább egy kijelölőnek ***mark***-típusú üzenetekre kell vonatkoznia (például ilyen a `mark.`, amely minden ***mark***-típusú üzenetre illeszkedik, vagy a `*.*`, amely minden típusú üzenetre illeszkedik).
Például adjuk meg, hogy a `syslogd` félóránként hozzon létre jelölőüzeneteket, és jegyezze fel őket a */var/log/messages*-be. Először a ***syslog.conf***-hoz a következőhöz hasonló sort add hozzá:

```
mark.*          -/var/log/messages
```

Ezután indítsd el a `syslogd`-t:

```
mylinuxbox:/etc/init.d# ./syslogd -m 30
```

Egy másik hasznos `syslogd`-kapcsoló a `-a [foglalat]`. Ennek segítségével adhatod meg a */dev/log* eszközön kívüli egyéb foglalatokat, amelyről a `syslogd` üzeneteket fogadhat.
Ha már próbáltál valaha biztonságossá tenni egy BIND-ot futtató névkiszolgálót, akkor esetleg használtad a `-a` kapcsolót, hogy a `chroot`-olt `named`-folyamat átadja az üzeneteit a nem `chroot`-olt `syslog`-folyamatnak egy *dev/log* eszközfájlon keresztül. Ebben az esetben a `named` nem éri el a */dev/log*-eszközt, de látja a sajátját, amely például a */var/named/dev/log*. Ezért a következő sort kell a */etc/init.d/syslog* fájlba írni:

```
daemon    syslogd  -m 0  -a  /var/named/dev/log
```

A démonutasítás a sor elején kizárólag a Red Hat indító parancsfájljaira jellemző, a fontos rész ez:

```
syslogd  -m 0  -a  /var/named/dev/log
```

Egynél több `-a` kapcsoló is megadható, például így:

```
syslogd -a /var/named/dev/log
➥-a /var/masikchroot/dev/log
➥-a /megintmasik/dev/log
```

Folytatva a 2. táblázat kapcsolóinak ismertetését, tegyük fel, hogy szeretnéd kipróbálni a ***syslog.conf.test*** fájlban megadott új syslog-beállításokat, de nem kívánod felülírni az eredeti ***syslog.conf***-ot, ahonnan az alapértelmezés szerint a `syslogd` a beállításait veszi. Használd a `-f` kapcsolót, amely után megadhatod a `syslogd` új beállításait tartalmazó fájl nevét:

```
mylinuxbox:/etc/init.d# ./syslogd
➥-f ./syslog.conf.test
```

Már említettük a `-r` kapcsolót, amelynek hatására a `syslogd` távoli gépekről is elfogad üzeneteket, viszont nem beszéltünk még a biztonsági megfontolásokról. Egyrészt a biztonság egyértelműen nő, amennyiben központi naplókiszolgálót használsz, vagy ha bármi olyat csinálsz, ami könnyebbé teszi a naplók kezelését és figyelését.
Másrészt figyelembe kell venned különféle fenyegetéseket. Érzékenyek a naplóadatok? Ha az üzenetek nem megbízható hálózaton utaznak keresztül, és az üzeneteket küldő kiszolgálók belső működését jobb titokban tartani, akkor a kockázat gyakorlatilag nagyobb, mint a haszon (legalábbis mint a `syslogd` hitelesítés nélküli, egyszerű szöveges távoli naplózási folyamatának a haszna).
Amennyiben ez az eset áll fenn, mindenképpen fontold meg a ***syslog-ng*** használatát. A syslog-ng képes a TCP-protokollon keresztül küldeni az üzeneteket, így együtt tud működni az Stunnel, az SSH-val és más programokkal, amelyek a biztonságát nagymértékben növelhetik. Mivel a syslog távoli naplózásra csak a kapcsolat nélküli UDP-protokollt használja, és ennek következtében az üzeneteket nem tudja átküldeni az SSH- vagy Stunnel-alagúton, kevésbé biztonságos, mint a syslog-ng.

Ha az üzenetek nem érzékenyek (legalábbis azok, amelyeket távoli kiszolgálón naplózol), még mindig ott van a szolgáltatás megtagadásának és az üzenethamisításnak a gondja. Amennyiben a `syslogd`-t a `-r` kapcsolóval indítod, minden távoli üzenetet elfogad, és egyáltalán nem vizsgálja sem az üzenet forrását, sem az üzenet tartalmát. Ez a kockázat is úgy csökkenthető a leginkább, ha áttérsz a syslog-ng használatára.

Létezik olyan eszköz, amelyet ha a sysloggal együtt használsz, részben mérsékelhető az érvénytelen távoli üzenetek kockázata: ez a TCP-burkoló. Pontosabban a TCP-burkolók gépelérés-engedélyező tulajdonságának segítségével egyszerűen megadható, hogy melyik gép kapcsolódhat és milyen protokollon keresztül a naplókiszolgálóhoz. A gépelérés-engedélyezőt könnyű becsapni a forrás IP-címének hamisításával (főleg azért, mert a syslog-tranzakciók szigorúan egyirányúak), azonban ez is jobb, mint a semmi, és valószínűleg elegendő ahhoz, hogy megakadályozza a kártékony, de lusta támadókat a syslog megzavarásában.

Ha a te véleményed is ez, szerezd meg és telepítsd a TCP-burkolót (bináris csomagját minden korszerű Linux-terjesztés tartalmazza, közülük számos alapértelmezés szerint telepíti is), a részletes útmutatóért olvasd el a host_access(5) súgóoldalt. Megjegyzendő, hogy bár a program neve mást sugall, a TCP-burkoló gépelérés-engedélyezését az UDP-t használó alkalmazások is használhatják.

Titkos naplókiszolgálók

Lance Spitzner, a Honeynet Project alkotója (➲ http://www.honeynet.org) javasolt egy trükköt, amely csalétekhálózatokon, de talán éles DMZ-kben (lásd Linuxvilág 2001. májusi szám 40. oldal) is használható: ez a titkos naplózás. A trükk segítségével lehetővé válik, hogy egy elosztóra (hub) vagy más osztott eszközre csatlakoztatott gép a naplófájlokat nem IP-címmel azonosított rendszerre küldje, amely látja és elkapja a naplóüzeneteket, de nem érhető el közvetlenül a hálózaton keresztül – ezért a hálózatba behatolónak a naplófájlokat sokkal nehezebb módosítania.

Az ötlet egyszerű: tegyük fel, hogy egy *syslog.conf* műveletben megadsz egy hamis IP-címet, azaz egy olyan IP-címet, amely érvényes a géped helyi hálózatában, csak éppen nem használja semmi, ami `syslogd`-t futtat. Mivel a syslog üzenetei a kapcsolat nélküli (egyirányú) UDP-protokollon keresztül utaznak, a küldő fél semmilyen visszajelzést nem vár, miután üzenetet küldött el.

Tegyük fel továbbá, hogy a DMZ gépei egy osztott eszközre, például egy elosztóra csatlakoznak, és minden hálózaton keresztül küldött syslog-üzenet szétszóródik a helyi hálózaton. Nem szükséges, hogy ezen a helyi hálózaton elhelyezkedő központi naplókiszolgálónak IP-címe legyen, mert a csomagokat passzívan leszippanthatja a `snort`-on vagy más csomagszippantón keresztül (a `tcpdump` erre nem alkalmas, mert csak a csomagok fejlécét figyeli, azonban az adatokat nem).

Magától értetődik, hogy az IP-cím nélküli naplókiszolgáló a szokásos IP-alapú rendszerfelügyeleti eszközökkel nem lesz elérhető, a naplófájlok megnézéséhez hozzá kell férned a konzoljához (hacsak nem nyomtatod ki őket sornyomtatón). Ráadásul nem elég a hamis IP-címet minden DMZ-be tartozó gép *syslog.conf*-jába beírni, minden küldő gépen egy hamis ARP-bejegyzést is be kell jegyezni. Ha nem teszed, a rendszerek hiába próbálnák meghatározni a gép ethernet-címét, amely az adott IP-hez tartozik, nem küldenének semmit.

Ha például azt akarod, hogy egy adott gép tegyen úgy, mintha a 192.168.192.168 hamis címre küldene csomagokat, add meg a @192.168.192.168 műveletet a *syslog.conf* megfelelő sorában vagy soraiban, és add ki a következő parancsot a héjból:

```
arp -s 192.168.192.168 03:03:03:31:33:77
```

Ez nem szükséges, ha a csomagokat a szokásos módon küldöd a naplózó gépre, azaz 192.168.192.168 a `syslogd`-t a `-r` kapcsolóval a futtató gép IP-címére.

2. lista Üzenet létrehozása minden üzenettípushoz minden fontossági szinten

```
#!/bin/bash
#
# A parancsfÆjl minden egyes  zenett pushoz
# minden egyes fontossÆgi szinten lØtrehoz
# egy  zenetet
#
for i in
{auth,authpriv,cron,daemon,kern,lpr,mail,
mark,news,syslog,user,uucp,local0,local1,loc
al2,
local3,local4,local5,local6,local7}

do
       for k in
{debug,info,notice,warning,err,crit,
              alert,emerg}
       do
              logger -p $i.$k "Pr ba zenet,
t pusa $i,
                              fontossÆga $k"
       done
done
```

A rendszernaplózás kipróbálása a „logger” paranccsal

Mielőtt befejeznénk a rendszernaplózó beállításának és használatának témakörét, még beszélnünk kell arról az eszközről, amellyel az új beállításokat kipróbálhatjuk – a használt naplózódémon típusától függetlenül. A `logger` parancssori alkalmazás, amely üzeneteket küld a rendszernaplózónak. Nemcsak állapotvizsgáló eszközként lehet használni, segítségével a héj parancsfájljai is felruházhatók a naplózás képességével.

Itt és most számunkra a diagnosztikai képességek érdekesek (bár gondold csak meg, ezt az eszközt minden fontos parancsfájlban alkalmazni kellene, amelyet rendszeresen futtatsz, különösen azokban, amelyek felügyelet nélkül a `cron`-ból vagy `at`-n keresztül futnak). A `logger` működése legegyszerűbben egy példán keresztül szemléltethető.

Tegyük fel, hogy megváltoztattad a syslog beállításait, és minden ***warn*** fontosságú ***daemon***-üzenetet a */var/log/warnings* fájlba küldesz. Az új ***syslog.conf*** fájlt úgy próbálhatod ki, hogy

További érdekességek

klogd(8): a `klogd` rendszermagnaplózó programot leíró súgóoldal, amely a nem alapértelmezett használati módokra is kitér (diagnosztika stb.).
logger(1): a logger segédprogramot leíró súgóoldal.
sysklogd(8): a syslog linuxos megvalósítását leíró súgóoldal, amely a `klogd`-vel való kapcsolatra is kitér; továbbá megadja a syslogd indítási kapcsolóinak listáját és leírását
syslog.conf(5): a syslog beállítófájljának lehetőségeit, nyelvtanát és használatát részletesen leíró súgóoldal.
syslog-ng: következő nemzedékbeli rendszernaplózó, amely sokkal hatékonyabb, mint a syslog.
➲ http://www.balabit.hu

először újraindítod a `syslogd`-t és a `klogd`-t, majd kiadod a következő parancsot:

```
mylinuxbox:~# logger -p daemon.warn
➥ Ez csak egy pr ba.
```

Láthatod, hogy a `logger` nyelvtana egyszerű. A `-p` kapcsoló után kell megadni a típust és a fontosságot. A `logger` mindent, ami ez után következik, legyen az kapcsoló vagy kijelölő, az üzenet részének tekint.
Mivel gyorsan gépelek, sokszor használok `while-do` ciklusokat a `bash` parancssorban, így rögtönzött parancsfájlok jönnek létre. A következő `bash` parancssorozat akár közvetlenül beírva, akár parancsfájlként is működik:

```
mylinuxbox:~# for i in {debug,info,notice,
➥warning,err,crit,alert,emerg}
> do
> logger .p daemon.$i  $i szintß daemon
➥pr ba zenet
> done
```

Ez ***daemon***-típusú próbaüzeneteket küld az összes lehetséges fontossági szinten. A 2. listán olvasható parancsfájl minden lehetséges üzenettípusból az összes szinten egy üzenetet hoz létre.

Összegzés

Remélhetőleg ennyi elég ahhoz, hogy elkezdd a saját syslog-beállításaidat felépíteni, kipróbálni és használni. Kívánom, hogy naplóid legyenek részletesek, bőségesek, jól ellenőrzöttek és unalmasak!

Mick Bauer (mick@visi.com) hálózati biztonsággal foglalkozó szaktanácsadó. 1995 óta a Linux elkötelezett híve, 1997 óta pedig OpenBSD prófétaként tevékenykedik. Mick minden kérdést és megjegyzést szívesen fogad.