Best Of sys.help

Matko Kosmat | 31 ožujka, 2026

Kako i sam naslov govori, ovdje smo pripremili kolekciju najkorisnijih savjeta za sistemce. Najčešće su to odgovori na pitanja sa Helpdeska za sistemce (skraćeno zvan sys.help), ali se mogu naći i savjeti općenite naravi.

Ukoliko ste svoje probleme s backupom riješili preko (hvaljene) CARNetove usluge sys.backup, vjerojatno ste u nekom trenutku dobili ovakvu poruku o greški:

Promijenio se "mtime"? Aha, to je onaj "modification time", vjerojatno zato jer je korisniku stigao mail nakon što je backup već počeo. No, što je tu je, tu se ne može ništa. Ipak je to *live backup*, nakon početka kopiranja datotečni sustav se i dalje mijenja.

Radi se, dakle, o informativnoj poruci o kojoj ne treba previše razmišljati. Samo, što je sa vremenima **ctime** i **atime,** koja se također dodjeljuju svakoj datoteci? Atime ("access time") je vrijeme koje se zapisuje svaki put kada se pristupi datoteci, nekim editorom (vi, joe) ili preglednikom (more, less, cat...). Očigledno se vrijeme **atime** ne uzima u obzir za backupiranje, jer sadržaj datoteke može ostati u potpunosti isti. Čemu backupirati datoteku koja se uopće nije promijenila, iako joj je možda pristupano nakon izmjena?
Sasvim je drugačija situacija s vremenom "ctime". **Ctime nije "creation time",** odnosno vrijeme kreiranja! Ovo se vrijeme uopće ne zapisuje u strukturama datotečnog sustava na Linuxu, nego je **ctime** zapravo vrijeme promjene pripadajućeg **inodea**. Inode je podatkovna struktura na disku, odnosno metapodatak u kojem se nalaze svi podaci o datoteci (tko je vlasnik, koje su dozvole, veličina itd), osim imena i sadržaja datoteke.

Ljudskom načinu razmišljanja je zapravo vrlo zgodno zapamtiti: **atime** je *access time*, **ctime** je *creation time*, **mtime** je *modification time*. No, to **nije točno** za vrijeme **ctime**.

Ponovimo, **ctime** jest vrijeme promjene, **ali pripadajućeg inodea**, što se događa prilikom promjene bilo kojeg podatka unutar strukture. U moguće uzročnike promjene vremena ctime možemo uvrstiti promjenu vlasništva (naredbom **chown**) ili modusa datoteke (naredbom **chmod**).

Ovo ima za posljedicu da promjena vremena **ctime** ne povlači promjenu vremena **mtime**, ali obrnuto itekako vrijedi. Pomoću naredbe ls možemo vidjeti kako se ta vremena mijenjaju. (Možete rabiti i naredbu **stat**, koja je opisana u članku /sys-trek-objave/naredbe-za-koje-mozda-niste-znali/#naredbe-za-koje-mozda-niste-znali-13-stat)

Za probu, kreirane su tri prazne datoteke, a, b i c. Naredba **ls** *po defaultu* prikazuje vrijeme mtime, a s opcijom "-u" prikazuje vrijeme atime:

Vidimo da jednostavna uporaba naredbe **cat** mijenja vrijeme pristupa datoteci, dok ostale vrijednosti ostaju nepromijenjene.

Sada možemo pogledati kako će se vremena promijeniti kada promijenimo **ctime** (ctime naredba ls prikazuje ukoliko uključimo prekidač -c):

Vrijeme pristupa i modificiranja se na datoteci c nisu promijenili. Promijenjeno je samo vrijeme promjene inodea (**ctime**), kako smo i namjeravali.

Pogledajmo još što će se dogoditi kada izmijenimo samu datoteku c.

Kao što smo i očekivali, promjena sadržaja datoteke c (a time i vremena **mtime**), je prouzrokovalo i promjenu vremena **ctime** (jer su se promijenili podaci u strukturi inodea). Ono što je možda iznenađujuće je da se vrijeme pristupa (**atime**) nije promijenilo. Ovo je stoga što se vrijeme atime ažurira samo kod operacija čitanja, a vremena **ctime** i **mtime** samo kod operacija pisanja.

Ponašanje kod osvježavanja vremena pristupa se može promijeniti s opcijama "**noatime**",

"**relatime**" i "**strictatime**". One se mogu upisati za svaku particiju posebno unutar datoteke /etc/fstab, ali se s njima u ovom članku nećemo baviti.

Kako te opcije mogu utjecati na performanse ulazno/izlaznih operacija, bit će zanimljivo obraditi i tu temu (pogotovo stoga što bi to značilo da bi neki programi mogli prestati raditi - npr. mutt, tmpwatch, neki programi za backup itd).

Javilo se nekoliko kolega sa sličnim problemima u radu web poslužitelja Apache. Naime, apache se nakon pokretanja ruši, ne radi stabilno i općenito ne radi pouzdano, pa ni monit ne može trajno pomoći (a ni nije mu namjena da stalno podiže neispravne servise). U logovima nalazimo prilično čudne zapise:

Ovo je prvi put da smo se susreli s ovim tipom greske, ali Google je pronasao dva rješenja. Prvi je recept promjena mutexa, koji se može pronaći na adresi **http://tinyurl.com/byc4tga**. Dovoljno je u datoteku /etc/apache2/apache.conf upisati sljedeće i restartati apache:

Čini se da se greška zna dogoditi kada apache naiđe na neki limit u resursima (broj dopuštenih otvorenih datoteka ili memorijski limit). Ukoliko gornji recept ne pomogne, možete probati promijeniti limite pomoću naredbe **ulimit**, koju ćete upisati negdje na početak datoteke /etc/init.d/apache2:

Sličan savjet smo našli na službenim stranicama projekta Apache: http://httpd.apache.org/docs/2.4/vhosts/fd-limits.html.

Alternativno, postoji bolje mjesto za postavljanje vrijednosti maksimalnog broja otvorenih datoteka, a to je u **/etc/apache2/envvars**:

Nadamo se da će vam neki od gornjih savjeta pomoći.

CARNet Debian 4.0 (baziran na Debian Etch distribuciji) sa novom verzijom kernel-2.6-cn paketa (specifično 3:2.6.24-1) prelazi na Debian-like kernele. To znači da se dosadašnji model praćenja recentnih "čistih" inačica verzija Linux jezgri napušta i prelazi na vlastiti derivat Debian kernela u kojem se nalaze sigurnosne zakrpe i eventualni dodaci za CARNet-Debian poslužitelje. Također napušta se LILO boot loader i prelazi na GRUB kao primarni i jedini mehanizam koji će koristiti buduće jezgre.

Ovo tipično znači sljedeće:

Nova jezgra se bazira oko etchnhalf inačice jezgre (2.6.24 jezgra) sa dodanim zaštitnim ExecShield patchem iz Red Hat Enterprise Linux distribucije te Layer 7 Netfilter dodacima.

Instalacija

Instalacija nove jezgre i prelazak se obavljaju na praktički uobičajeni način, s razlikom da je potrebno napraviti Grub direktorij prije instalacije:

Instalacija će automatski povući i odgovarajuću jezgru (linux-image-2.6.24-etchnhalf.1+cn1-686-bigmem_2.6.24-6~etchnhalf.7+cn1 paket), dok se u kernel-2.6-cn paketu sada nalaze rutine potrebne za fino podešavanje i prelazak na Grub.

Izrazito je bitno da ne restartate (*rebootate*) poslužitelj ako instalacijske rutine jave ikakvu grešku, već da se javite na sustav pomoći za sistem-inženjere, ili direktnije, na e-mail paketi@carnet.hr.

Rješavanje problema

U slučaju da ipak dođe do problema (što se ne bi smjelo desiti!) i da se poslužitelj ne želi podići nakon restarta jer se Grub nije ispravno instalirao, najjednostavnija metoda oporavka je iskoristiti Linux RIP (http://rip.7bf.de/current/) distribuciju. Sa navedene Web stranice skinite odgovarajuću ISO datoteku i snimite na CD medij, te podignite poslužitelj s tog medija. Kad dobijete root prompt, dovoljno je iskoristiti ove naredbe:

Te u Grub promptu nastavite i zabilježite si na papir izlaz sljedeće naredbe:

U slučaju da ne dobijete ništa, probajte sa:

Tipičan rezultat je (hd0,0), odnosno Grub pronalazi da se njegova stage1 datoteka nalazi na prvoj particiji (0) na prvom disku (hd0). Shodno tome, instaliramo Grub koristeći te parametre (koriste se uvijek parametri koje je izbacila gornja naredba i niti jedni drugi!) i dalje pišući unutar Grub prompta:

Posljednja naredba mora ispisati niz redova i završiti sa "Done." u posljednjem retku. Iz programa izađite sa kombinacijom tipki Ctrl+C te restartajte poslužitelj, nakon čega bi se morao pojaviti uobičajeni Grub izbornik.

Kriptografski certifikati, kao što već znate, omogućuju sigurnu autentikaciju, a time i komunikaciju između dvije točke na Internetu, ali i sigurnu identifikaciju suprotne strane. Ovo omogućuje asimetrična kriptografija, no teorijom se u ovom članku nećemo baviti. Iako postoje tvrtke koje prodaju certifikate i time nedvojbeno potvrđuje da je web site vaš, certifikati koje sami potpisujete (*self-signed*) omogućuju isto što i komercijalni. Jedina razlika su dosadni *pop-up* prozori u *browseru* koji će stalno pitati korisnika vjeruje li dobivenom certifikatu ili ne.

Već neko vrijeme, CARNet nudi "prave" certifikate, u vidu DigiCert certifikata za akademsku zajednicu. Nažalost, neke institucije članice CARNeta nemaju pravo na ove certifikate, zato ćemo obraditi kreiranje i *self-signed* certifikata za one koji nemaju pravo na DigiCert certifikate.

1. Pravljenje certifikata preko CARNetovog TCS servisa

2. Pravljenje samopotpisanih certifikata

3. Ugrađivanje certifikata u servise

4. Provjera rada

5. Dodatne napomene

1. DigiCert certifikati

Na web stranicama http://certifikati.carnet.hr možete pronaći detaljne upute kako doći do ovih certifikata. Kontakt e-mail je tcs-ra@carnet.hr, a možete se obratiti i telefonom na broj 01/6661-770.

Sve članice CARNET-a imaju pravo na DigiCert certifikat, a ne samo punopravne članice. No, prije svega morate registrirati svoju ustanovu.

Kako su na gore navedenim stranicama detaljno napisane sve upute, ovdje nećemo ponavljati iste informacije. Možete konzultirati točku 2.b) kako bi vidjeli kako se kreira zahtjev za certifikatom s više FQDN-ova, ali inače možete odmah skočiti na točku 3.

Još ćemo samo napomenuti:
- nemojte kriptirati ključ! (stavite "encrypt_key = no" u *.cnf* datoteci)
- rabite datoteku MultiSCSreq.cnf, jer je vrlo izvjesno da ćete trebati više FQDN-ova u istom certifikatu

2. Samopotpisani (*self-signed*) certifikati

Samopotpisani certifikati omogućuju sigurnu komunikaciju, ali će korisnik morati potvrditi "da vjeruje vašem certifikatu" svaki put kad posjeti vaš site preko SSL-a. Kod drugih servisa ovaj problem nije tako izražen, odnosno krajni korisnici ne vide problem (HTTP, IMAP, POP3...). Popup prozor će se pojavljivati sve dok korisnik trajno ne prihvati certifikat izdan od strane vaše institucije.

Možemo napraviti:

a) certifikat za jedan DNS unos (FQDN - Fully Qualified Domain Name, puno ime poslužitelja)
b) certifikat za višestruke FQDN-ove

Preporučujemo da odmah izradite certifikate za sve poslužitelje koje imate ili koje ćete imati.

Kako možete znati koje ćete poslužitelje imati? Možete jednostavno staviti kakva generička imena (npr. geografske pojmove), ili ime poslužitelja podesiti prema njegovoj funkciji (mail, www, student i

slično, a prijedloge smo dali u točki 2.b).

Zar ne postoje *wildcard* certifikati (*.domena.hr)? Postoje, ali njihova uporaba nije preporučljiva iz sigurnosnih razloga. Treba uzeti u obzir i da nijedna akademska institucija u HR neće imati više od nekoliko javno dostupnih servisa, stoga nema ni prevelike potrebe za *wildcardovima*.

2. a) certifikati za jedan FQDN

Certifikat možete kreirati za samo jedan FQDN na jednostavan način (navest ćemo tri primjera):

I) Najbrži način (postupak vrijedi samo za paket apache2-cn!)

Preko apachejeve install skripte generiramo nove certifikate, koji se obično nalaze u /etc/apache2/ssl.* direktorijima (inače, svi se certifikati mogu prebaciti u /etc/ssl/certs direktorij). Ovakav postupak će kreirati potpuno novi certifikat, ali će uzeti u obzir samo osnovni FQDN. Dakle, ako vaš poslužitelj ima ime "server.domena.hr", i certifikat će glasiti na to ime, te primjerice https://www.domena.hr neće raditi (samo https://server.domena.hr).

Gornji primjer za apache2 je specifičan (jer interno rabi posebnu skriptu carnet-generate-ssl), pa ćemo prikazati druge, generičke, slučajeve. Iako ovaj postupak rabe i drugi paketi, navest ćemo primjer za **dovecot** (dok se za primjerice postfix rabi certifikat snakeoil.pem iz paketa ssl-cert):

II) Uporaba skripte iz paketa apache2-cn (postupak vrijedi samo ukoliko imate CARNetov paket apache2-cn!)

Kako automatsko generiranje certifikata uzima u obzir samo osnovno ime računala, maloprije spomenuta skripta **carnet-generate-ssl** isporučena uz CARNetov paket apache2-cn vam pomaže da zadate **bilo koje ime** koje će se naći u certifikatu (uz uvjet da je unešeno u DNS). Najčešće će to biti "www.domena.hr".

III) Potpuno manualni način (ovaj postupak vrijedi za sve servise!)

Sve postupke iz gornjih točaka možete napraviti i ručno, uz možda malo više tipkanja:

Na gore opisani način možete generirati (odnosno obnoviti) certifikat za bilo koji servis.

2.b) certifikati za višestruke FQDN-ove

Za self-signed certifikate možete uporabiti istu datoteku kao i za kreiranje DigiCert certifikata, MultiSCSreq.cnf. U nju, u sekciji "[req]" dodajte redak "encrypt_key = no". Ona će tada izgledati poput ovog:

Brojeve (DNS.1, DNS.2..., 3...) možete povećavati do broja poslužitelja koji vam treba, no nemojte pretjerivati. Predlažemo sljedeća imena (uz osnovno ime poslužitelja), no to naravno ovisi samo o vašim potrebama:

1. www
2. mail (pop, imap, smtp)
3. student
4. webmail
5. forum
6. sluzba_x (studentska služba ili slično) itd.

Više od 10-ak imena vam sasvim sigurno neće trebati, pogotovo ako rabite osnovno ime poslužitelja za više servisa (npr. "server.domena.hr" za mail, web i ostalo).

3. *Ugrađivanje* certifikata u mrežne servise

Preostaje samo certifikate uključiti u svaki servis koji mislite rabiti. Ovo ovisi o konkretnom načinu konfiguracije svakog servisa, a mi ćemo prikazati sve standardne servise koji se mogu susresti na CARNetovim poslužiteljima.

Certifikate je potrebno zaštititi prije instaliranja. To ćemo napraviti s naredbom (naravno, treba

promijeniti ime datoteke po vašim potrebama):

Drugim riječima, certifikat ne smije biti dostupan nikome osim rootu i korisniku pod čijim se UID-om servis vrti.

Apache 2.x

Konfiguracijska datoteka nije nužno u **/etc/apache/conf.d/ssl.conf**, kako je to uobičajeno, nego može biti u konfiguraciji bilo kojeg virtualnog poslužitelja u direktoriju sites-available ili samostalno, primjerice u datoteci **/etc/apache2/sites-available/ssl**. Ukoliko je konfiguracija u samostalnom VHOST-u, provjerite je li taj virtualni poslužitelj sa SSL-om uključen:

Konfiguracija:

Ključ bi trebao biti u svom direktoriju, nedostupan običnim okrisnicima i zaštićen ovim pravima pristupa (chmod 600):

Treba još samo restartati Apache sa naredbama:

Napomena: apache obično zahtijeva potpuno zaustavljanje i ponovno pokretanje kako bi počeo rabiti certifikat.

Postfix

Konfiguracijske datoteke su **/etc/postfix/main.cf** i **/etc/postfix/master.cf**. Varijable **smtp_*** se rabe za konfiguraciju dijela za slanje pošte od vašeg serverra, dok se varijable **smtpd_*** rabe za konfiguraciju dijela za zaprimanje pošte (samo jedno slovo razlike, ali ta razlika je bitna). U /etc/postfx/main.cf:

Ovime ste omogućili STARTTLS na portu 25, što je dovoljno za sigurno slanje maila.

Ukoliko ipak želite omogućili **preporučeni** port 587 (submission) za **sigurno** i **autenticirano** slanje maila, u datoteci **/etc/postfix/master.cf** treba odkomentirati/dopisati sljedeće:

*Opcija **smtpd_tls_security_level** (Postfix 2.3 i noviji) zamjenjuje opcije **smtpd_enforce_tls** i **smtpd_use_tls** (< Postfix 2.3). Stara opcija je navedena jer se učestalo pojavljuje u raznim uputama na Internetu, pa smo htjeli naglasiti da se u Squeezeu i dalje ne treba rabiti.*

Port za SMTPS (465) koji se nekada rabio bi trebalo zakomentirati, jer je prenamijenjen za nešto drugo (multicast). No, u praksi se i dalje rabi za SMTPS, pa je odluka na vama:

Na kraju, treba restartati postfix servis s naredbom:

Dovecot

~~**Dovecot 1.0 / 1.2 (squeeze)**~~

~~Konfiguracijska datoteka je i dalje /etc/dovecot/dovecot.conf, ali je potrebno navesti i prijelazni certifikat.~~

```
~~ssl_disable = no~~
~~ssl_cert_file = /etc/ssl/certs/mojserver.cert~~
~~ssl_key_file = /etc/ssl/private/mojserver.key~~
~~ssl_ca_file = /etc/ssl/certs/mojserver.chain~~
```

~~Za **dovecot 1.2 (squeeze)**, parametar ssl_disable ne postoji, nego treba umjesto njega rabiti:~~

```
~~ssl = yes~~
```

~~I ovdje naravno treba restartati servis sa:~~

```
~~# /etc/init.d/dovecot reload~~
```

~~**Dovecot 2.1.7 (wheezy)**~~

~~U izdanju **wheezy** konfiguracijska datoteka je razlomljena na više manjih. Konfiguracija SSL-a se nalazi u datoteci **/etc/dovecot/conf.d/10-ssl.conf**, a ukoliko ste obavili nadogradnju s paketom **carnet-upgrade**, nalazi se i u datoteci **95-cn7-upgrade.conf**.~~

~~*Uvijek vrijedi definicija koja dolazi u "alfanumerički novijoj" datoteci, pa možete upisati retke i u vlastitu datoteku, primjerice 99-local.conf. Ta će konfiguracija onda vrijediti bez obzira što piše u prethodnim datotekama (ovo vrijedi za bilo koji konfiguracijski parametar).*~~

~~Relevantni retci se ponešto razlikuju:~~

```
ssl_ca = </etc/ssl/certs/mojserver.chain
ssl_cert = </etc/ssl/certs/mojserver.cert
ssl_key = </etc/ssl/private/mojserver.key
```

~~**Napomena**: inačica dovecota u wheezyu samopotpisane certifikate stvara unutar direktorija /etc/dovecot, odnosno ključeve u /etc/dovecot/private. Vidjeti http://www.dovecot.org/doc/README.Debian-wheezy.~~

Dovecot 2.2.13 (jessie)

U izdanju **jessie** konfiguracijska datoteka je razlomljena na više manjih, kao i u prethodnoj inačici za **wheezy**. Konfiguracija SSL-a se nalazi u datoteci **/etc/dovecot/conf.d/10-ssl.conf**, a ukoliko ste obavili nadogradnju s paketom **carnet-upgrade**, nalazi se i u datoteci **95-cn7-upgrade.conf**.

Uvijek vrijedi definicija koja dolazi u "alfanumerički novijoj" datoteci, pa možete upisati retke i u vlastitu datoteku, primjerice 99-local.conf. Ta će konfiguracija onda vrijediti bez obzira što piše u prethodnim datotekama (ovo vrijedi za bilo koji konfiguracijski parametar).

Relevantni retci se ponešto razlikuju od inačice u jessie, ali samo po tome da se očekuje da je certifikat spojen s prijelaznim certifikatom:

Napomena: Certifikat i prijelazni certifikat spajate na ovaj način:

Redoslijed certifikata je bitan, pa tako mora ići prvo vaš certifikat, pa onda prijelazni certifikat. Naziv nastale datoteke nije bitan, može biti .pem ili .crt ili .cert - svejedno je.

Vsftpd

Konfiguracijska datoteka je /etc/vsftpd.conf.

~~*Specifično je da starije inačice Vsftpda (do Lennyja) zahtijevaju da **ključ i certifikat budu u jednoj datoteci**, što možete napraviti sa*~~

```
# cat mojserver.pem mojserver.key >> mojserver-key-cert.pem
```

~~*Napravili smo posebnu datoteku kako ne bi potencijalno ometali druge servise koji traže da se ključ i certifikat nalaze u posebnim datotekama.*~~

~~*Sad možete restartati servis sa:*~~

```
# /etc/init.d/vsftpd reload
```

Novije inačice vsftpda (distribucija Lenny i dalje) imaju dodatnu direktivu **rsa_private_key_file** gdje **morate** navesti ključ:

Kako bi funkcionirao cijeli lanac povjerenja, spojite prijelazni certifikat s osnovnim:

U direktivi rsa_cert_file sada treba biti navedena ova spojena datoteka:

Sad možete restartati servis sa:

Proftpd

Konfiguracijska datoteka je /etc/proftpd.conf.

Servis nakon ovoga treba restartati sa:

4. Provjera ispravnosti instaliranih certifikata

Port 443 je port na kojem HTTPS servis radi, te ga morate promijeniti u odgovarajući za ostale servise (primjerice 993 za IMAPS, 443 za HTTPS, 995 za POP3S, itd).

Ispravan rad servisa označavaju DVA ulančana certifikata (pojavljuju se DVA zaglavlja -----BEGIN CERTIFICATE----- i -----END CERTIFICATE-----), poslužiteljski i prijelazni pomoću kojega SSL klijenti mogu uspostaviti lanac povjerenja.

5. Dodatne napomene

Datoteke certifikata i ključeva mogu imati različite nastavke (ekstenzije), no sve su one u PEM formatu (ostali mogu biti DER i PKCS12). Tako, u *.pem datoteci može biti ključ, certifikat, ili čak i certifikat i ključ zajedno. Status možemo provjeriti otvaranjem dateoteka:

Dakle, datoteke mogu imati različite nastavke, a sadržavati iste stvari. Ovo je bila samo brza provjera, detaljnije informacije možete naći uporabom naredbe openssl:

Dakle, radi se certifikatu za Dovecot za poslužitelj "server.domena.hr". Ako izostavite naredbu *grep*, dobit ćete punu informaciju o certifikatu.

Za certifikate s višestrukim FQDN-ima, pod retkom "X509v3 Subject Alternative Name" bi trebali pisati svi vaši poslužitelji koje ste odabrali.

Izgled generiranog zahtjeva za certifikatom (za provjeru prije nego ga pošaljete TCS timu) možete provjeriti s naredbom:

Ovime smo prošli sve što CARNet sistemcu može zatrebati oko certifikata. Naravno, poželjno je dalje samostalno istražiti informacije.

Ovaj članak u potpunosti zamjenjuje članak http://sistemac.carnet.hr/node/48
OSVJEŽENO: 2019-06-13

Dana 22. veljače u prijepodnevnim satima mnogi su sistemci dobili dojavu svojih korisnika da im mailovi ne stižu na odredište, ali i obrnuto, njima poslani mailovi nisu stizali u njihove sandučiće e-pošte. Nakon što su pogledali u logove, vidjeli su da poštu zaustavlja Amavis i stavlja u karantenu kao virus. Stavljanje u karantenu nije nešto neuobičajeno, ali broj zaustavljenih mailova svakako jest, kod nekih je iznosio 50%, pa i više. Je li riječ o nekoj greški u sustavu, ili se pojavio opasan virus koji je odjednom zarazio mnoga računala? Pogledajmo logove:

Korisnik je preko webmaila slao veliku datoteku, za koju je provjerom antivirusnim alatima utvrđeno da nije zaražena, što dodatno ukazuje na nekakvu grešku u mail sustavu. Kako se na CARNetovim poslužiteljima rabi isključivo besplatni antivirusni softver ClamAV (osim ako vaša institucija nije kupila neki drugi), vjerojatno je riječ o nekoj greški unutar ClamAV-a.

Pregledom web stranica ClamAV-a mogli smo naći informaciju da se ne radi o službenoj bazi (zbog ekstenzije UNOFFICIAL), nego o dodatnim antivirusnim definicijama koje se mogu naći u bazi MalwarePatrola. Naravno riječ je o definiciji MBL_207346, koja se nalazi u datoteci /var/lib/clamav/mbl.db:

Ova zadnja definicija, odnosno potpis (signature) je očigledno kraći, je li moguće da se datoteka nije skinula do kraja i da je puklo na ovom potpisu? Datoteke koje nisu unutar osnovnog ClamAV-a se osvježavaju svaki sat preko cron skripte /etc/cron.hourly/clamav-sanesecurity s adrese http://www.malware.com.br/cgi/submit\?action=list_clamav (iz Brazila!). Provjerom smo pronašli da je zaista riječ o datoteci koja nije kompletna, jer bi trebala završiti s ovim potpisom:

Dakle, mnogo ključeva nedostaje, a barem jedan je oštećen. No, zašto onda zaustavlja većinu

mailova i označava ih kao virus? Ovdje nam može pomoći vlastiti ClamAV-ov alat "sigtool" (hvala Valentinu!):

Za usporedbu, ovako izgleda taj potpis na neoštećenom sustavu:

Znači, ako se u mailu pojavljuje "www.thinkertec.com/trial" mail će biti blokiran. A u gornjem slučaju? **Bit će blokiran svaki mail koji sadržava niz znakova "www."!**

Ne moramo napominjati da ovo za neke znači gotovo potpuni prekid rada mail sustava, da ne spominjemo stotine i tisuće mailova u karanteni koje treba pronaći i "osloboditi".

No, nije sve tako strašno, jer su mailovi i dalje u karanteni, nisu obrisani i mogu se ponovo poslati. Neka vam pomogne ovaj *one-liner*:

(cijeli ovaj niz naredbi mora biti u jednom redu)

Ovaj će niz naredbi u logovima provjeriti jesu li zaustavljeni mailovi po potpisu MBL_207346, pripremiti popis oznaka u karanteni te ih proslijediti naredbi amavisd-release. Ukoliko samo želite provjeriti postoje li zaustavljene poruke, izostavite dio s naredbom xargs.

Napomene: ukoliko je vaš mail.log na drugoj lokaciji, poput /var/log/mail.log, navedite tako u naredbi. Također, ukoliko su se logovi već zarotirali (a svakako hoće prko noći), provjerite i datoteku mail.log.0.

Čini se da naposlužiteljima sa starom inačicom ClamAV-a ovaj problem nije izražen u većoj mjeri. Prepoznat ćete stari ClamAV po poruci "Your ClamAV installation is OUTDATED", iako to ne znači da ClamAV ne radi, samo da je izašla barem jedna novija inačica.

Situacija se navečer normalizirala, pa više ne postoje parcijalni zapisi. Ukoliko unatoč tome ne želite riskirati, onemogućite cijelu MalwarePatrol bazu, tako da preimenujete datoteku mbl.db i onemogućite daljnje osvježavanje:

Još samo trebate na samom početku datoteke /etc/cron.hourly/clamav-sanesecurity dodati "exit 0". Time onemogućujete pokretanje skripte. Ponovit ćemo, ove dvije operacije nisu nužan korak, jer trenutno sve radi kako treba i nema potrebe blokirati ovu korisnu dodatnu zaštitu od virusa, malvera i spama.

Nadamo se da smo pomogli i da će vaši korisnici biti tolerantni prema ovom problemu.

Cron je nezaobilazni sistemski servis, kojega koristimo u svakoj prilici i za svakovrsne potrebe. Iako cron nema veću rezoluciju od jedne minute, za većinu potreba je ovo sasvim dovoljno. Konfiguracija crona je svima poznata, ali ima detalja koje vrijedi proučiti.

Standardni redak u cronu (točnije datoteci crontab) se sastoji od 5 polja: minute, sati, dan u mjesecu, mjesec i dan u tjednu, plus jedno polje za naredbu koju imamo namjeru izvršiti. Prvih pet polja se mogu jednostavno zamijeniti s posebnim ključnim riječima:

Vrlo je zgodno, umjesto da pamtimo koje je koje polje, jednostavno staviti ove skraćene oblike, poput:

Posebno je zanimljiv "@reboot". On će, sukladno imenu, pokrenuti program kod reboota ili hladnog starta računala. Za ovu se operaciju obično koristi /etc/init.d/rc.local, ali to je skripta iz paketa "initscripts" pa svoje modifikacije možete slučajno pregaziti u nekoj nadogradnji. Također, rc.local izvršava naredbu kao korisnik root, pa morate koristiti su ili sudo kako bi izvršavali naredbe kao neki drugi korisnik.

Cron nema tih problema, i na startu će izvršiti kao standardni cronjob, dakle pod UID-om vlasnika tog cronjoba.

Standardni cron na Linuxu prihvaća i nekoliko drugih proširenja, kojih nema ili dugo nije bilo na standardnim Uniximа. Moguće je postaviti varijablu PATH, te tako proširiti popis direktorija u kojima će se tražiti naredba. Može i suprotno, iz sigurnosnih razloga ograničiti broj direktorija samo na

nekolicinu:

Varijabla se upisuje na vrhu crontab datoteke, kao i MAILTO:

Varijablu MAILTO ćete upotrijebiti kada želite obavijest o izvršenom cronjobu poslati na neku drugu adresu, a ne na adresu vlasnika cronjoba. Ukoliko ne želite nikakve obavijesti mailom, varijablu postavite na "praznu" vrijednost ovako:

Alternativno, možete preusmjeriti standardni izlaz i grešku u /dev/null, pa se mail neće poslati jer nema nikakvog izlaza (outputa):

Ostale varijable koje možete upotrijebiti, odnosno promijeniti su: HOME i SHELL.

Iz crona možete direktno poslati mail, bez da koristite posebnu skriptu za to. Iako iz skripte možete ispisati neki tekst, što će vam se poslati mailom, ponekad je praktičnije koristiti ovakve onelinere:

Svakog ponedjeljka u 8:30 ćete dobiti mail naslova "PODSJETNIK" i sadržaja kojeg upišete. Pri tome je znak "%" oznaka za novi red (\n), pa svako njegovo pojavljivanje znači prijelaz u novi red. Ukoliko baš želite upotrijebiti znak "%", morate ga upisati ovako: "\%".

Cron je dosta fleksibilan, ali je orijentiran na periodično izvršavanje. Ne možete odrediti točan datum kada se nešto treba izvršiti, jer nema mogućnosti upisivanja godine. Da biste to postigli, možete upotrijebiti ovakav oblik:

Prvog prosinca 2015. godine u podne će stići obavijest da će isteći certifikat za servise, iako cron ovakve stvari ne omogućava direktno. No, malo skriptiranja će pomoći.

Ukoliko imate naprednije korisnike, možete kontrolirati njihovu uporabu crona na način sličan tcp_wrappersu. Pomoću datoteka /etc/cron.allow i /etc/cron.deny možete kontrolirati tko može rabiti cron. Ukoliko nijedna datoteka ne postoji, svi mogu koristiti cron. Ukoliko postoji datoteka /etc/cron.deny, korisnici kojima je zabranjeno korištenje crona moraju biti navedeni. Ukoliko postoji datoteka /etc/cron.allow, korisnici koji mogu koristiti cron moraju biti navedeni unutar datoteke.

Koji ćete način koristiti ovaj mehanizam, ovisi o vašoj sigurnosnoj politici.

Cron je stari, ali pouzdani servis, a sve što treba je znati iskoristiti sve prednosti koje ekstenzije pružaju. Iako su neke ekstenzije specifične za linux, sumnjamo da je danas to problem, jer linuxi dominiraju nekad neprikosnovenim Unix tržištem.

Nakon nadogradnje na Debianovu distribuciju Squeeze poneki sistem-inženjeri pomalo panično pregledavaju logove u potrazi za neobičnim zapisima. Čine to kako bi na vrijeme uočili da neki servis ima problema u radu. No, mnogi se oslanjaju isključivo na logcheck i OSSEC da im dojavi sumnjive unose u logovima. Ovo, pak, nije poželjno, jer ti programi nikad neće poslati sve relevantne retke iz logova koji bi mogli pomoći da se problem riješi. Dakle, ipak je potrebno otići u logove i "baciti pogled". Jedan od upita koji su nastali nakon takvog "istraživanja" koji smo zaprimili na sys.help tiče se crona i čudne poruke "grandchild failed".

Iako čudna, poruka se odnosi na procese koje cron pokreće izvršavajući poslove koje ste mu zadali (*cronjobs*). U Unix i Linux svijetu ispravan završetak izvršavanja nekog programa ili skripte uvijek je popraćen izlaznim kodom 0 (exitcode). Kako to ovdje nije slučaj, jer je izlazni kod 1, cron će prijaviti grešku putem poruke u logovima. Da je prilikom izvršavanja skripte nastao i kakav izlazni tekst, korisnik root dobio bi mailom. Kako se to u ovom slučaju nije dogodilo, ostaje nam istržiti koji cronjob izaziva ove poruke. Ukoliko pokrenemo naredbu grep nad log datotekom, dobit ćemo ovakav rezultat:

Neki se proces pokreće svakih 5 minuta i uzrokuje navedene poruke u logovima. Sada imamo kakav-takav trag. U Linuxovom cronu vrijeme izvršavanja cronjoba može se zadati na dva načina, tradicionalnim:

ili skraćenim:

Pa krenimo u potragu za tim cronjobom. Cronjobovi se nalaze na nekoliko mjesta, pa je

najjednostavnije iskoristiti opciju "-r" naredbe grep koja uključuje internu rekurziju:

Osumnjičena datoteka je iskrsla, a u ovom konkretnom slučaju riječ je o zaostatku amavis-stats paketa, koji se više ne održava i može se obrisati:

Za dodatnu sigurnost, provjerite je li obrisana datoteka /etc/cron.d/amavis-stats, pa ukoliko nije, obrišite je. Paket
amavis-stats-cn je inačice izbačen iz CARNetove inačice distribucije Squeeze, kao uostalom i Debiana (zbog neodržavanja programa).

Cijelu ovu situaciju smo opisali uglavnom kako bismo vas naučili sami istražiti neobične ili rijetke manifestacije problema na vašim Linux poslužiteljima. U konkretnom slučaju, osim neobičnog unosa i zauzimanja mjesta u logu, nikakve stvarne opasnosti nije bilo, ali ovakvi "trikovi" vam mogu pomoći u nekom sljedećem, ozbiljnijem, slučaju.

Postavljanje kvote na disku spada u obavezne radnje sistemca u slučaju kada na poslužitelju postoji više korisnika. To se posebno odnosi na poslužitelje na kojima se nalaze neodgovorni korisnici, kakvi se uvijek nađu među studentima.

Postoje dvije osnovne vrste kvota. Prva određuje količinu diskovnog prostora koja se može koristiti, a druga se odnosi na broj datoteka i direktorija koje se mogu kreirati. Moguće je postaviti tzv. *soft* kvotu, limit nakon kojeg korisnik mailom dobiva obavijest da se približava efektivnom limitu, tzv. *hard* kvoti. Također, može se definirati i *grace period* koji dozvoljava korisnicima da kratkotrajno premaše postavljeni limit. On je obično postavljen na 7 dana.

Kvote se tipično implementiraju postavljanjem ograničenja na diskovni prostor korisnicima (*per user*) i/ili grupama korisnika (*per group*). Na taj se način korisnicima ne dozvoljava da iskoriste sav diskovni prostor na pojedinoj particiji i time dovedu u pitanje ispravan rad poslužitelja. U slučaju kvote nad grupama, "kvaka" leži u činjenici da se diskovni prostor ograničava cijeloj grupi, a ne pojedinim korisnicima.

Za primjer ćemo implementirati *per user* kvotu na */var* i */home* particije. Jedina stvar koja nam treba jest paket *quota*. Njegova je instalacija standardna. Sljedeći korak jest promjena datoteke */etc/fstab tako* da u konfiguraciju particije na koju želimo implementirati kvotu, dodamo parametar *usrquota* i/ili *grpquota*. Prepravljeni dio datoteke može izgledati na sljedeći način:

Nakon toga treba napraviti *remount* svih particija koje smo mijenjali:

Na svakoj particiji treba još kreirati datoteku *aquota.user* i prava čitanja i pisanja dodijeliti isključivo *root* korisniku. Ta datoteka sadrži tablicu trenutnog zauzeća diska na određenoj particiji. Za kvote nad grupama napravimo datoteku *aquota.group*.

Ako se rade izmjene na već uključenoj kvoti, nju treba isključiti kako bi promjene mogli kasnije aktivirati. U suprotnom, ovaj se korak preskače:

Kvotu zatim treba inicijalizirati tj. popuniti datoteke *aquota.user*:

Kvote za korisnike se postavljaju tako da odredimo kvotu za jednog proto-korisnika. To se obavlja naredbom *edquota*:

Otvorit će se zadani tekstualni editor u kojem postavimo limite. Za primjer ćemo proto-korisniku postaviti *soft* limit od 20MB, a *hard* limit nešto veći:

Stupac *blocks* prikazuje količinu diska (u jedinicama od 1k) koju korisnik trenutno koristi, a stupac *inodes* prikazuje broj datoteka i direktorija koje je taj korisnik već kreirao na pojedinoj particiji. Svaka od dvije vrsta kvota ima svoj *soft* i *hard* limit, a vrijednost 0 znači da kvota nije postavljena. Iz primjera je vidljivo da nije ograničen broj direktorija i datoteka koje korisnik može kreirati.

Nakon što smo proto-korisniku postavili limite, njih je potrebno preslikati na sve ostale korisnike koristeći naredbu

Ovo može biti zamorno jer treba navesti redom sve korisnike kojima želimo uključiti kvotu. Ako je riječ o poslužitelju na kojem se nalaze isključivo studentski korisnički računi, možemo iskoristiti činjenicu da njihovi *uid*-i idu redno. U datoteci */etc/passwd* treći podatak u svakom retku označava *uid* (primjerice, korisnik *www-data* ima *uid* 33). Provjerimo prvi i zadnji *uid* i preslikamo kvotu naredbom:

U ovoj naredbi koristeći awk iz datoteke */etc/passwd* izdvajamo *uid* svih korisnika koji se nalazi u traženom rasponu i na sve te korisnike dupliciramo kvotu našeg proto-korisnika. Važno je pripaziti da ne postavimo kvotu sistemskim korisnicima poput *www-data* jer bi time ugrozili rad nekih servisa. Srećom, oni imaju *uid* manji od 1000 pa ih je lako razlikovati od "ljudskih" korisnika. Zadnji korak je uključivanje same kvote na particijama:

Time smo dovršili postavljanje kvote. Za provjeru kvote pojedinog korisnika koristi se naredba

Zgodnom se može pokazati i naredba *repquota* kojom možemo zbirno prikazati kvote svih korisnika. Naredbom *edquota –t* moguće je postaviti već spomenuti *grace period*. Također, naredba *quotacheck* obavlja provjeru datotečnog sustava i kvote te ispravlja sve moguće greške. Nju je zgodno postaviti u dnevni *cron*. Naravno, preporuka je da se pogledaju *man* stranice svih navedenih naredbi. Na kraju napomenimo da je kvotu, naravno, moguće uključiti i na operativnom sustavu Windows Server.

P.S. Zaboravio sam napomenuti da se za obavijest o približavanju hard limitu koristi naredba *warnquota* čija se konfiguracija nalazi u datoteci */etc/warnquota.conf*. I ovu naredbu bi trebalo postaviti u dnevni *cron*.

Već smo se navikli na jednostavnost i upotrebljivost koju nudi Debianov paketni sustav. Ovo uključuje i upravitelje paketima (apt-get, aptitude, dpkg...) i repozitorije, koje možemo i sami kreirati za vlastite potrebe. Također, možemo rabiti i druge, neslužbene, repozitorije te tako rabiti pakete koje Debian ne podržava iz bilo kojeg razloga (najčešće nedovoljno otvorena licenca). No, trenutno aktualna stable distribucija "Etch" zahtijeva jedan mali dodatni korak kako bi uspješno mogli rabiti sve te repozitorije.

Promjena je zapravo jednostavna, a sastoji se od toga da svaki paket mora biti digitalno potpisan GnuPG ključem. Iako će se paket moći instalirati i bez odgovarajuće provjere, apt-get će se svaki put pobuniti:

Odgovorom "Y" i dalje ćete moći instalirati pakete, ali pravo rješenje je skinuti javni ključ repozitorija i instalirati ga na računalo. Ovo vrijedi kako za službene Debian repozitorije, tako i za sve ostale, uključujući i CARNetov. Iako u ovu svrhu već postoje gotovi paketi koji donose ključeve i sve čine umjesto vas, postoji i potpuno ručni način.

Primjerice, ključevi za službeni Debianov repozitorij dolazi s paketom "**debian-archive-keyring**", te ćete njegovom instalacijom riješiti problem neuatenticiranih paketa. Slično je i s paketom "**debian-backports-keyring**" i "**carnet-keyring**". No, ako gotovog paketa nema, možete to isto napraviti i ručno, na primjeru backports.org repozitorija:

Ovaj postupak podrazumijeva da je maintainer repozitorija stavio ključeve na javni poslužitelj ključeva. Kod malih repozitorija moguće je da ni ovo nije napravljeno. No, tada ostaje još jedna solucija:

Ovaj postupak vrijedi za bilo koji repozitorij, samo trebate naći gdje se ključ nalazi, skinuti ga te uvesti (importirati) u bazu koju rabi APT. Ne mora se rabiti niti wget za skidanje (možete skinuti ključ kako želite), pa je tako osnova naredba za uvoz ključeva:

FTP (File Transfer Protokol) protokol je star vjerojatno jednako kao i sam Internet (tada se još zvao ARPANET). Njegova specifičnost je da se naredbe prenose preko jednog komunikacijskog kanala (**ftp** na portu 21), a sami podaci preko drugog (**ftp-data na** portu 20). Sam protokol nećemo ovdje opisivati, jer je na njega potrošeno previše "digitalne tinte". Ono što nam je problem koji pokušavamo riješiti je slanje zaporke u čitljivom obliku, koji osnovni FTP protokol redovito radi.

Ako ste pomislili "pa čemu uoopće rabiti FTP, postoji scp, sftp, Secure FTP..."? Pa, kao i uvijek, zadržati kompatibilnost unatrag je jako korisno, kako se vaši korisnici ne bi previše bunili. S vremenom ćete ih već sve prebaciti na FTPS.

FTPS nemojte buniti s SFTP-om, niti bilo kojim drugim oblikom kvazi-ftp protokola ili tuneliranjem preko SSH-a. Ovo je "čisti" FTP protokol, s dodanom podrškom za SSL, odnosno TLS. Za ovo mu trebaju, dakako, dva nova TCP porta (**ftps-data** na **989** i **ftps** na **990**), ali zbog kompatibilnosti, može u nekim uvjetima nastaviti raditi na starim portovima 20 i 21. Možda mislite da ne postoji mnogo klijenata s podrškom za FTPS, ali mogli bi se prevariti. Podržava ga mnoštovo Linux/Unix i Windows klijenata (CuteFTP, cURL, FireFTP, WS FTP, WinSCP itd.)

FTPS ima, u principu, dva načina rada: FTPS i FTPeS. Prvi je bezuvjetni, implicitni oblik, i ukoliko vaš klijent ne podržava FTPS, neće se moći spojiti. Drugi je uvjetni, eksplicitni oblik, gdje klijent određuje koji će dio komunikacije biti enkriptiran, kojim načinom enkripcije i slično. Stari programi će se moći spojiti na ovakav poslužitelj. FTPeS je fleksibilan, i klijenti će moći primjerice zahtijevati da je autentikacija enkriptirana, ali sâm prijenos podataka ne (time se dobije nešto na brzini).

U članku "Certifikati za servise" na adresi /sys-trek-objave/best-of-sys-help/#certifikati-za-servise smo opisali kao dodati certifikat u vsftpd. Nije toliko bitno je li certifikat službeni ili samogenerirani, iako preporučujemo prvi (kako ga dobiti pogledajte u članku http://www.carnet.hr/sc_servis).

Postavke za vsftpd, koje dodajemo u /etc/vsftpd.conf i koje omogućuju i uobičajeni način rada su:

Slijede postavke koje će omogućiti da klijent izabere hoće li rabiti stari načina rada, ili rabiti neki od sigurnih protokola (izbacit ćemo ponovljene retke):

Ukoliko želite da i anonimni korisnici imaju pristup preko sigurne konekcije, postavite varijablu **allow_anon_ssl** na YES. Ukoliko ne želite da se i sami podaci enkriptiraju, stavite **NO** na varijable **force_anon_data_ssl** i **force_local_data_ssl** (prijava korisnika će i dalje biti enkriptirana).

U čemu je prednost FTPS-a nad drugim oblicima enkriptiranog prijenosa podataka? Ukoliko rabite neki od SSH protokola, morate imati korisnike na sustavu. Za FTPS to ne morate imati, dakle sigurnost je samo uvećana. Ovo je odlično ukoliko imate web hosting, gdje korisnici samo trebaju uploadati svoje web stranice, a mail i ostalo im ne treba.

Sigurno vam se već dogodilo da u sandučiću elektroničke pošte nađete duplikate poruka. Razlog može biti pogreška POP/IMAP poslužitelja, pogreška vašeg mail klijenta ili jednostavno ljudska pogreška kod konfiguracije (višestruki korisnički računi i slično). Problem će nam pomoći riješiti mala skripta, pisana u programskom jeziku Python.

Skripta će analizirati vaš sandučić, te po zaglavlju Message-ID kojeg ima svaka poruka, izbaciti duplikate. Ovi duplikati su filtrirani isključivo po Message-ID zaglavlju, poruke koji imaju samo istog pošiljatelja i isti Subject neće biti smatrani duplikatima.

Zaglavlje Message-ID izgleda kao dugačak niz znakova, i jedinstveno označava svaku poruku:

Po ovom zaglavlju se može prepoznati da je poruka identična nekoj drugoj, a neki mail sustavi čak automatski sprječavaju isporuku identičnih poruka. Drugi sustavi to mogu, ali tek uz neki dodatak (plugin). Postoje dodaci i za mail klijente, primjerice Evolution. No, sve je to kasno ukoliko imamo sandučić sa tisućama mailova u kojem biste htjeli maknuti višak.

Ovdje u pomoć priskače ova jednostavna skripta, koja se pokreće na ovaj način:

Moj_mailbox je sandučić kojeg želite pregledati, i ne smije biti u uporabi (kako se sadržaj ne bi promijenio tijekom analize). U suprotnom, riskirate da izlazni sandučić bude nečitljiv.

S opcijom -i određujete ulaznu, a s opcijom -o određujete izlaznu datoteku.

Opcija -h će generirati indeks (hash) datoteku, kako biste kasnije mogli brže napraviti redukciju sandučića, ali nije ju neophodno rabiti, pogotovo ako sandučić nije prevelik.

Primjer uporabe:

Program će za svaki novi Message-ID ispisati "New Message-ID", a kada nađe duplikat ispisat će, naravno, "Duplicate Message-ID". U datoteci Moj_mailbox.ok će biti skraćena inačica sandučića, bez duplikata.

Preporučujemo da skriptu primjenite ukoliko arhivirate poštu nekog neaktivnog korisnika, ili na velikim sandučićima gdje se uvijek mogu naći duplikati. Na taj način možete uštediti nešto prostora na disku i drugim resursima, što na velikom broju korisnika može biti primjetno.

Autor skripte je Marilen Corciovei (len@len.ro), a originalnu skriptu možete skinuti ovdje. Ukoliko se pojavi nova inačica, možete ju pronaći na njegovom blogu na adresi: http://www.len.ro/2009/01/remove-duplicate-mails/

Svi smo davno upoznati s činjenicom da proizvođači, odnosno njihovi marketinški odjeli u reklamama daju samo najbolje osobine svojih proizvoda. Ako se nešto može legalno i slagati, tim bolje. Sjećate se CRT monitora, kad su proizvođači mjerili dimenzije stakla ekrana, umjesto vidljive povrsine? Na taj način je proizvod imao bolju prodaju, iako se 17 inča znalo spustiti ispod 16. Slično je i čvrstim diskovima, uvijek se navodi lijepa, okrugla brojka, iako je gotovo uvijek pravi kapacitet bio niži. Primjerice, disk od 120 gigabajta (120 * 10^9 bajta) zapravo ima 111,79 gigabajta.

Proizvođač rabi SI jedinice koje su u dekadskom brojevnom sustavu, što ima zgodnu posljedicu prividnog povećanja kapaciteta. Da bi kapacitet bio ispravno prikazan, potrebno je rabiti binarni sustav. Svi smo u školi učili da 1 kilobajt nema 1000 bajtova, nego 1024, ali da se dogovorno koriste SI jedinice, iako postoji ta određena razlika. To je bilo prihvatljivo, dok se nisu pojavili veći diskovi i veće memorije. Razlike je oko 2% u razredu veličina kilobajta, i do 10% u razredu veličina terabajta. Naravno, razlika ide sve više kako se iznos povećava, pa je razlika između SI i binarnih jedinica za jotabajt nezanemarivih 20%. Više o tome možete pročitati u Wikipedijinom članku na http://en.wikipedia.org/wiki/Binary_prefix.

Kako bi se ovaj problem riješio, još od šezdestih se godina pokušava uvesti posebna jedinica. Tek 1999. je IEC (International Electrotechnical Commission) u dokumentu IEC 60027-2 uspio standardizirati nazivlje za binarne prefikse. Oni su navedeni su u tablici:

IEC		**Uporaba za memorije itd.**			**Uporaba za diskove**			
Ime	**Sim-bol**	**Baza 2**	**Baza 10 (binarno značenje)**		**Baza 10 (SI značenje)**		**Ime**	**Simbol**
kibi	Ki	2^{10}	1,024	$>10^{3}$	10^{3}	1,000	kilo	k/K
mebi	Mi	2^{20}	1,048,576	$>10^{6}$	10^{6}	1,000,000	mega	M
gibi	Gi	2^{30}	1,073,741,824	$>10^{9}$	10^{9}	1,000,000,000	giga	G
tebi	Ti	2^{40}	1,099,511,627,776	$>10^{12}$	10^{12}	1,000,000,000,000	tera	T
pebi	Pi	2^{50}	1,125,899,906,842,624	$>10^{15}$	10^{15}	1,000,000,000,000,000	peta	P
exbi	Ei	2^{60}	1,152,921,504,606,846,976	$>10^{18}$	10^{18}	...	exa	E
zebi	Zi	2^{70}	1,180,591,620,717,411,303,424	$>10^{21}$	10^{21}	...	zetta	Z
yobi	Yi	2^{80}	1,208,925,819,614,629,174,706,176	$>10^{24}$	10^{24}	...	yotta	Y

Dakle, po ovom prijedlogu, **gibibajt** bi imao oznaku **GiB**, a iznosio bi točno 2^{30} bajtova, odnosno

1073741824 bajtova. Gigabajt, s poznatom oznakom GB, bi sad nedvosmisleno iznosio 1000000000 bajtova. Popularni DVD-5 format (oko 4,700,000,000 bajtova) bi tako na kutiji zapravo trebao imati dvije vrijednosti: 4.7 GB (gigabajta) i 4.37 GiB (gibibajta).

Nova nomenklatura se već polako pojavljuje u određenim aplikacijama i situacijama, a koliko će trebati da nazivi **kibibajt**, **mebibajt**, **gibibajt**, **tebibajt** i ostali u potpunosti zažive ostaje za vidjeti. Do tada, lijepo je znati da će se situacija konačno raščistiti tijekom određenog vremena.

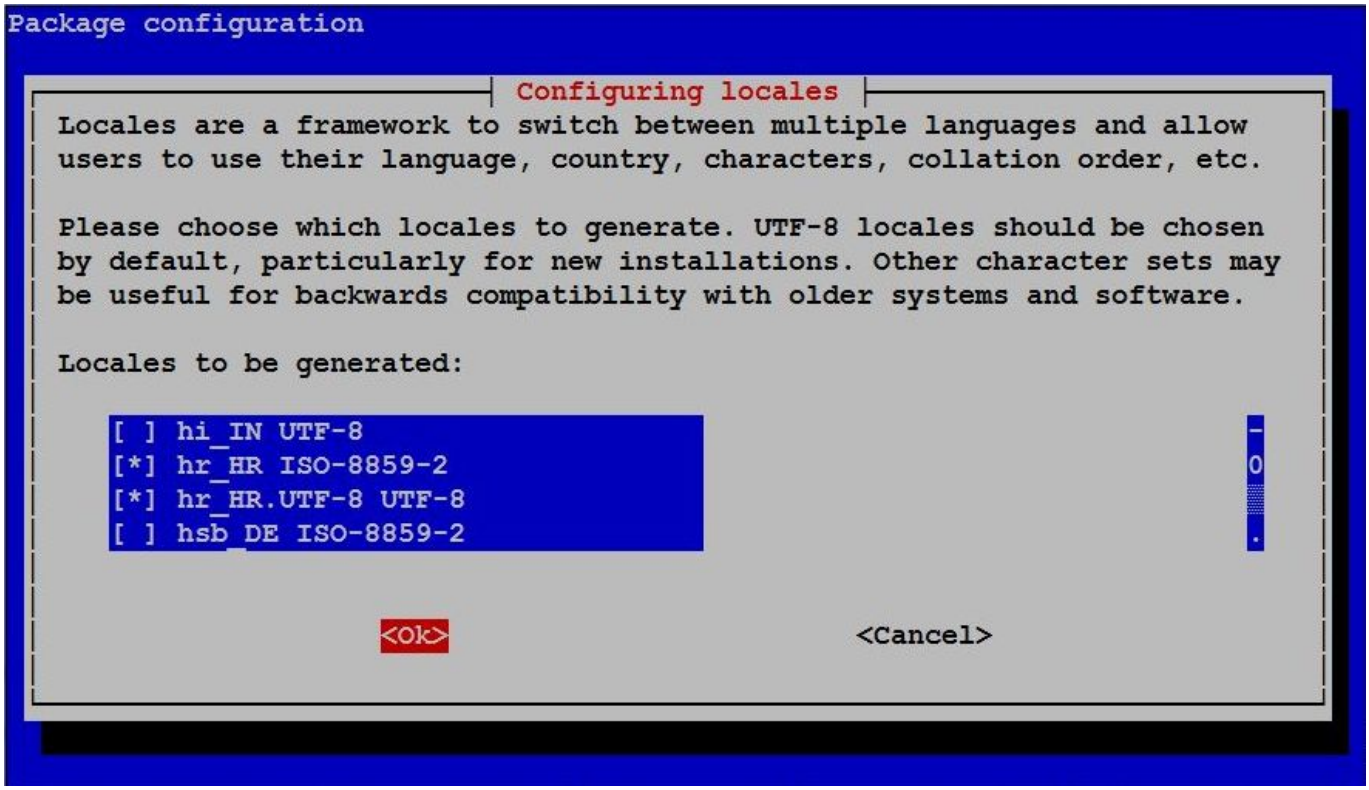
Greška "**Unable to open /etc/default/locale**" je bezazlena greška, ako se uopće može greškom nazvati. Neke kolege je uznemirila, jer se pojavljuje u kontekstu naredbe **su** (koja podiže privilegije korisnika na razinu drugog korisnika, najčešće administratora). No, radi se samo o nepostojanju datoteke koja određuje osnovni *locale* sustava (lokalizacijske postavke, poput datuma, sortiranja i poruka sustava itd.). Prijava problema koju smo zaprimili je glasila:

> To: logcheck@domena.hr
> Subject: server.domena.hr 2010-10-28 10:02 Security Events
>
>
> Security Events for su
> =-=-=-=-=-=-=-=-=-=-=-=-
> Oct 28 09:04:34 server su[822]: pam_env(su:session): Unable to open env file:
> /etc/default/locale: No such file or directory
> Oct 28 09:20:35 server su[2650]: pam_env(su:session): Unable to open env file:
> /etc/default/locale: No such file or directory
>
> o cemu se ovdje radi, jel nesto opasno ili mogu ignorirati?

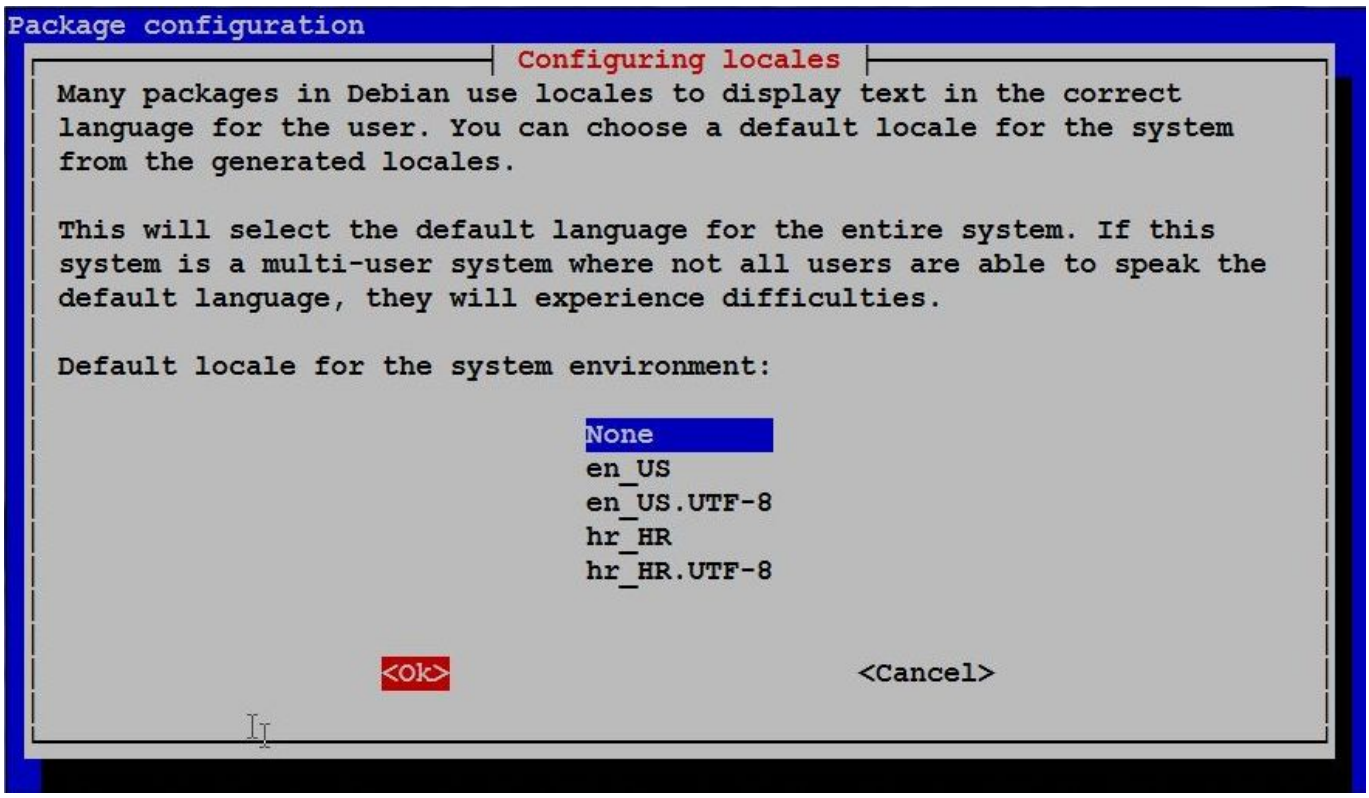
Ova datoteka se jos od distribucije Debian Etch rabi umjesto stare datoteke /etc/environment. Neki je programi eksplicitno traže, iako ona može biti potpuno prazna. Datoteka /etc/default/locale se nije kreirala ili je obrisana vjerojatno prilikom nadogradnje sustava ili paketa **locales**.

Ukoliko samo želite brzo rješenje, bit će dovoljno napraviti:

Time ste "ušutkali" sve programe koje traže da ova datoteka postoji. No, pravi put bi bio rekonfiguriranje paketa locales. S naredbom "dpkg-reconfigure locales" možete odabrati novi osnovni *locale*. Naravno, možete odabrati i "None", pa sustav neće imati osnovni *locale*, nego samo ugrađeni minimalni ("C"). Prvo, trebate odabrati sve *locale* koje želite izgenerirati:



Nakon toga odabirete osnovni *locale* sustava, u ovom slučaju "**None**":



Datoteka /etc/default/locale će se kreirati, ali bez definicije *locale*a, dakle efektivno će biti prazna (osim komentara u njoj):

Ukoliko u drugom prozoru odaberete bilo što osim "**None**", to će postati osnovni *locale* za sustav, pa će tako i pisati u /etc/default/locale.

Eto, nadamo se da smo vas naučili dvije stvari jednim člankom: kako popraviti "grešku" koju vam

dojavljuje sustav, te kako promijeniti *locale* na sustavu (ukoliko želite ispisivanje datuma na hrvatskom i slično).

Greška koju ćemo opisati u ovom članku nije nešto što ćete prečesto viđati, no svejedno, kad je vidite sjetite se Portala za sistemce, ukucajte "statoverride" u tražilicu i naći ćete rješenje vašeg problema.

Zašto se ova greška javlja, i što je uopće **statoverride** datoteka? Dpkg paketni sustav je moćan i kompleksan, a ovo je jedna od njegovih svojstava koja mu omogućava ovu snagu i fleksibilnost. Datoteka se nalazi u direktoriju /var/lib/dpkg/ i sadržava ovakve unose:

I površnom će čitatelju biti jasno da se ovdje radi o nečemu što utječe na vlasnika, grupu i prava nad datotekama ili direktorijima. I bit će u pravu, jer ako razdvojimo ime datoteke na "stat" i "override", funkcija postaje jasna: moguće je preko ovog mehanizma prijeći preko osnovnih prava datoteke u nekom paketu sa pravima po želji.

Ova datoteka se može uređivati ručno ali, ima i svoju naredbu: dpkg-statoverride. Obično se ova naredba ne rabi, nego je uobičajeno rabe postinst skripte paketa (i na taj način izbjegavaju "grub" način promjena sa chown/chmod). Naravno da naredbu možete po potrebi rabiti, ukoliko imate potrebu da određena grupa korisnika ima pristup određenoj datoteci ili direktoriju, a da reinstalacije paketa to ne poremete.

Vratimo se našem problemu, koji sprječava da napravite upgrade, ili instalaciju određenog paketa. Situacija obično nastaje propustom maintainera paketa da nakon operacije "purge" obriše unos u statoverride datoteci:

Greška može varirati, ali osnovni problem je isti. Pri tome ne možete dalje ništa instalirati. Problem možemo riješiti na dva načina, ručnim brisanjem unosa u /var/lib/dpkg/statoverride koji se ticu grupe 'freerad', ili nešto
zaobilaznijim načinom:

Nakon ovoga biste trebali moći reinstalirati željeni paket, odnosno instalirati i održavati paketni sustav i dalje.

Ukoliko od prije niste čuli za "*Here documents*", s pravom se možete upitati čemu što se tako čudno zove uopće može služiti. Riječ je, pojednostavljeno rečeno, o načinu kako običan tekst ubaciti unutar programskog koda, a pri tome se može sačuvati i formatiranje i izvršiti supstitucija varijabli. *Here documents* se rabe u programiranju u Unix shellu, Perlu, PHP-u, Windows PowerShellu i mnogim drugim jezicima. Kako je najlakše ovaj princip pokazati na primjeru, idemo odmah s jednim jednostavnim primjerom:

Ovaj primjer je pisan u Bash shellu, te će izvršavanje ove skripte ispisati ovakav ispis:

Umjesto naredbe **cat**, koja ovdje služi samo kao primjer, možete rabiti **mail**, **write** ili bilo koju drugu naredbu koja očekuje ulaz sa tipkovnice. Ovaj primjer je koristan za obavještavanje studenata (obično ćemo to činiti iz crona) da će im na određeni datum isteći korisnički račun ukoliko ne donesu potvrdu o upisu godine (u ovisnosti o politici na vašoj instituciji). No, vratimo se samom primjeru.

Znak "<<END" je oznaka gdje počinje tekst kojeg ne treba interpretirati kao naredbe, nego kao obični tekst. No, ipak se vrši substitucija varijabli pa možete rabiti varijable koje ste definirali u samoj skripti ili pak vanjske varijable iz okoliša (*environmenta*) skripte. Oznaka kraja teksta je ista riječ, samo bez znakova "<<"

Valja napomenuti da riječ ne mora biti "END", može biti bilo koja jedinstvena riječ, a najčešće se rabi "EOF", "END" ili kod nas "KRAJ" i slično. Jedna važna stvar, ukoliko oznaku početka *Here documenta* stavite u navodnike (bilo dvostruke, bilo jednostruke!), substitucija varijabli se neće obaviti (primjerice, cat <<"END" ili cat <<'END'):

I drugi programski jezici podržavaju *Here documents*. Evo primjera za programski jezik **Perl**:

Dakle, uporaba je slična. Bitna razlika je da ukoliko stavite jednostruke navodnike, substitucija

varijabli se neće obaviti (pisat će $posiljatelj i $primatelj).

Za popularni skriptni jezik **PHP** situacija je nešto drugačija, a i "<<" se mijenja u "<<<":

Također, moguće je cijelu konstrukciju dodijeliti varijabli, preko oblika "$varijabla = <<<EOF". Dalje s varijablom možete činiti što vam je volja.

Nadamo se da su primjeri bilo dovoljno jasni te da smo vam uspjeli razjasniti čemu *Here documents* služe, te da ćete ih moći ubuduće samostalno rabiti.

IP tuneli su objekti mrežnog sloja OSI modela koji nastaju postupkom tuneliranja - omatanja IP paketa u novi paket na izvorišnom kraju tunela, te izdvajanja unutrašnjeg IP paketa iz primljenog IP paketa na odredišnom kraju tunela.

Primjer korištenja ovakvog tuneliranje je tvrtka koja radi na dvije udaljene lokacije. Računala na obje lokacije treba smjestiti unutar iste privatne mreže (s IP adresama iz nekog od privatnih raspona adresa). Drugi primjer bila bi tvrtka na istoj lokaciji koja koristi dvije privatne mreže, a mi želimo da nam računalo iz one druge privatne mreže bude vidljivo. U ovakvim slučajevima oblikujemo tunel i definiramo krajeve tunela - čvorove koji na originalni IP paket dodaju novo zaglavlje ili s primljenog paketa skidaju vanjsko IP zaglavlje što omogućuje putovanje paketa kroz nekompatibilni adresni ili protokolni prostor.

Protokoli za tuneliranje dijele se na nespojno-bazirane (datagram-based; IP in IP, GRE, SIT, IPsec, PPTP, MPLS i dr.) i spojno-bazirane (stream-based; SSH, TLS). U ovom ćemo članku objasniti IP in IP, GRE i SIT tunele.

IPIP tuneli su najjednostavniji tuneli i podrazumijevaju enkapsulaciju IPv4 paketa unutar IPv4 paketa. To je moguće isključivo za unicast promet. IPIP ne možemo koristiti u multicast okolini, primjerice kod korištenja OSPF, RIP, BGP i sličnih protokola.

GRE (Generic Routing Encapsulation) tuneli (originalno razvijeni i široko upotrebljavani od Cisca), između ostalog, omogućavaju i prenošenje multicast i IPv6 prometa. Nazivaju se još i IPv6/IPv4 over IPv4 tuneli. Tipična je primjena ovih tunela u mrežama s dinamičkim usmjeravanjem.

SIT (Simple Internet Transition) tuneli nazivaju se još i IPv6 over IPv4 tuneli. Nalaze primjenu u povezivanju izdvojenih IPv6 mreža preko IPv4 područja.

Prije stvaranja tunela na Linuxu treba uključiti sljedeće module jezgre: za IPIP modul ipip, za GRE modul ip_gre i za SIT modul ipv6. Pogledajmo sada ispis podataka za sučelja:

Možemo uočiti da se učitavanjem svakog od modula pojavilo jedno pripadajuće sučelje. Uočimo iz ispisa overhead zbog dodavanja vanjskog zaglavlja: mtu vrijednost za eth0 sučelje pokazuje da su IP paketi veličine 1500 okteta. Kod IPIP i SIT tunela, maksimalna je veličina paketa 1480 okteta, dok je kod GRE tunela overhead najveći i iznosi 24 okteta.

Obratimo još samo pažnju na to da ukoliko konfiguriramo usmjernik uključimo odgovarajuće funkcije usmjernika:

Nećemo koristiti generička sučelja sit0, tunl0 i gre0, nego ćemo stvoriti vlastiti tunel. Sintaksa naredbe za dodavanje tunela je:

TUN_NAME je proizvoljan string (ime tunela), MODE je ipip, gre ili sit, a ADDR su adrese krajeva tunela (čvorovi koji dodaju ili odvajaju dodatno IP zaglavlje). Vidimo da niti jedan od krajeva tunela ne mora biti definiran (ili može biti podešen na default vrijednost any). Tunel s nedefiniranim izvorišnim krajem koristi se kod mreža s dinamičkim pridjeljivanjem adresa.

Ovisno o tome da li je definirana adresa udaljenog kraja tunela (remote ADDR), tuneli se dijele na dvije skupine:

Prvi navedeni, pointtopoint tuneli, nakon stvaranja se konfiguriraju kao i svako fizičko sučelje - dodaje im se adresa i potrebne rute:

Kod NBMA tunela sklopovlje ne zna kamo treba proslijediti paket te je zato potrebno unijeti rutu na drugačiji način (navodimo primjer za NBMA tunl0 sučelje):

Nakon ključne riječi via navodimo adresu udaljenog kraja tunela. U članku o ip route naredbi objasnili smo da nakon ključne riječi via možemo navesti jedino adresu sučelja usmjernika koji je izravno dostupan preko našeg sučelja (u ovom slučaju tunl0). Kako u ovom slučaju udaljeni kraj tunela nije vidljiv, koristimo ključnu riječ onlink kako bi jezgra ipak stvorila rutu.

Dodatna opcionalna polja kod stvaranja tunela su primjerice ttl ili dsfiled i sl, a njihove podrazumijevane (default) vrijednosti podešene su na inherit (što znači da se vrijednosti nasljeđuju od mrežnog uređaja kojem je tunel pridružen). Tuneli tipa GRE koriste i neke dodatne opcije koje niti IPIP niti SIT tuneli nemaju: npr. podešavanje ključa (običan ili dotted-decimal broj) s opcijom key, te stvaranje, odnosno provjeravanje, jednosmjerne sume opcijom csum.

Pokažimo sada primjer stvaranja IPIP tunela (stvaranje preostalih tipova tunela potpuno je analogno). Zamislimo slijedeće mreže:

Provala na računalne sustave je bilo i bit će. Efikasnog recepta kako ovo spriječiti "jednom za svagda" nema, no problem se može ublažiti konstantnim obrazovanjem kako sistem-inženjera, tako i krajnjih korisnika. Naravno da korisnike nije prihvatljivo učiti o kriptografskim metodama i *one-time* zaporkama, ali ih možemo podučiti kako odabrati dobru zaporku. Ni u kojem slučaju zaporka ne smije biti iz rječnika, bilo engleskog, bilo kojeg drugog jezika. Zaporke koje su poznati pojmovi iz okolice korisnika, imena bliže rodbine ili kućnih ljubimaca su jako podložne socijalnom inženjeringu i ne treba ih rabiti. Zapravo,
najbolje je rabiti nekakav generator slučajnih zaporku, poput pwgena. No, korisnici možda ne žele ili ne mogu zapamtiti ovakve zaporke i opet imamo problem.

Kad nemamo kontrolu nad zaporkama korisnika, ili smo naslijedili sustav pa ne znamo koliko su zaporke korisnika "snažne", jedino što možemo pokušati je poslužiti se alatima koji pokušavaju probiti zaporke korisnika, rabeći pri tome
različite tehnike. Nećemo previše ulaziti u te *password cracking* tehnike, spomenut ćemo samo rječničke napade (kada se rabe riječi iz različitih rječnika ili jednostavno zbirki riječi), te *brute-force* napade (kad se pokušava probiti zaporku nizanjem znakova po redu, "aaaa", "aaab", "aaac" itd).

No, dosta često se s nekog udaljenog (napadačkog) računala pokušavaju probiti korisnički računi, tako da se nižu već negdje snimljene zaporke. Obično se napdaju neki od poznatih serivsa, kao što su ftp, ssh ili SASL. Isto tako, razni trojanski programi na PC računalima znaju presresti korisničke zaporke i poslati ih na direktno napadaču. Sve to rezultira slanjem spama preko vašeg vlastitog poslužitelja, u zadnje vrijeme najčešće SASL servisa, koji je po *defaultu* uključen u CARNetovu distribuciju Debiana.

Podsjetimo se, SASL omogućava korisnicima da pošalju mail s bilo koje adresu na Internetu rabeći svoju zaporku, i na taj način zaobilazi "*relay denied*" poruke.

Ipak, dosta često je riječ o slabim zaporkama, pa nam ostaje zadatak kako ih pronaći. Upotrijebit ćemo jedan od češćih programa za *cracking* zaporki, John the Ripper. Instalacija je jednostavna:

Prvo što ćemo napraviti je spojiti /etc/shadow i /etc/passwd datoteku u jednu. Za to postoji alat "unshadow":

Ova novonastala datoteka sadrži i zaporke i korisnička imena, pa je stoga bitno da je zaštitite od čitanja bilo kome osim korisniku root.

Obećali smo da nećemo ulaziti previše u detalje, jer tražimo samo slabe zaporke pa nam naprednija podešavanja ni ne trebaju. U tome će nam pomoći i sam John the Ripper, koji, ukoliko ne navedete nijednu opciju, automatski rabi 3 najbitnija načina *crackiranja* zaporki: *single* način (pokušava pogoditi zaporku samo na osnovu podataka iz GECOS polja), rječnički napad (pokušava pogoditi zaporku na osnovu riječi iz rječničkih datoteka) i inkrementalni način (*brute force* napad s

kombinacijama bilo kojih znakova, a sama duljina nizova se s vremenom povećava).

Da vidimo kako to izgleda:

Svaki put kada stisnete tipku <enter>, ispisat će se statistika, te koja se zaporka trenutno pokušava pogoditi. Iz ovoga se jasno vidi na koji način john radi: dodaje razne dodatne znakove, uključujući i interpunkcije, pokušavajući emulirati način na koji korisnici biraju zaporku i tako ubrzati proces pronalaženja zaporke (ovo radi u bilo kojem načinu rada, *single*, *dictionary*...).

Iako se otkrivene zaporke prikazuju na ekranu, i naknadno možete vidjeti pogođene zaporke rabeći opciju "-show":

Iz ovoga je sasvim jasno da zaporke tipa "test123" ili "korisnik222" ne mogu biti dobre. Ovakve, ali i kompleksnije permutacije dolaze i s drugim načinima rada, stoga se za dobru zaporku, ponavljamo, obratite specijaliziranim programima.

Umjesto toga, možete upotrijebiti i jedan stari trik: uzmite neku rečenicu koju znate napamet (npr. poslovica), te odaberite samo prva slova. Primjerice:

Ovakve zaporke nema u nijednm rječniku, a *brute force* metodom će trebati previše vremena da se zaporka pronađe. No, vi ćete ovakvu zaporku moći lakše zapamtiti nego nekakav slučajni niz znakova. Također, ukoliko stavite pokoji interpunkcijski znak ili veliko slovo, dobit ćete dosta sigurniju zaporku, primjerice:

O ovoj temi se mogu napisati stranice i stranice teksta, ali vas nećemo previše zamarati s kriptografskim temama (osim ako sami ne pokažete zanimanje!). Vrijedi zapamtiti samo to da nikada nećete imati siguran sustav dok su korisničke zaporke slabe. Pomoću sigurnosne politike i u suradnji s upravom vaše institucije, naučite korisnike kako odabrati dobru zaporku. Svakako povremeno "provrtite" John the Ripper (primjerice preko noći) i provjerite ima li kakvih problematičnih zaporki na sustavu.

Sretno!

U prošlom članku smo opisali kako nepažnja može prouzročiti probleme na sustavu koje je ponekad teško locirati, a mogu biti izvor sporadičnih problema (članak se nalazi na adresi /sys-trek-objave/best-of-sys-help/#poruka-pam_radius_auth-failed-looking-up-ip-address-for-radius-server-other-server). Na naše iznenađenje, nakon primjene recepta iz članka situacija se nije popravila. Barem ne u potpunosti.

Naime, u logovima se (i dalje) javljala poruka:

Zaista, bilo je za očekivati da smo problem riješili u prvom koraku, ali očigledno nije tako. Nijedan se korisnik nije žalio, kako se i zašto onda pojavljuje ova poruka? U logovima se vide mnoge uspješne prijave korisnika, te samo poneka poruka da se radius poslužitelj ne javlja. Je li riječ o preopterećenju? Pogledajmo konfiguracijsku datoteku **/etc/pam_radius_auth.conf**:

Možemo vidjeti da treće polje određuje vrijeme čekanja da poslužitelj odgovori na upite. Vrijeme je postavljeno na 1 sekundu, što je sasvim dovoljno za današnje brze poslužitelje. Ili ipak nije? Povećali smo vrijednost na 5 sekundi:

Nakon te promjene, nikakvih problema niti poruka više nije bilo.

Čini se da je problem prouzročen presporim diskovnim podsustavom, ali do ove situacije može doći i na druge načine. No, popravak je vrlo lagan, a eventualno usporenje korisnicima neprimjetno (procjenjujemo da je u slučajevima kada se poruka pojavljivala, najdulje vrijeme čekanja bilo između 1 i 2 sekunde).

Kad bi barem svi problemi na poslužitelju imali ovako lako rješenje...

Često do nas dolaze pitanja kako instalirati CARNet-Etch distribuciju na "čisti" poslužitelj, dakle onaj na kojemu nije instalirana prethodna inačica Debiana (u ovom slučaju Sarge). Često se traži i lokacija .iso datoteke sa slikom instalacijsko CD-a. No, instalacijskog CD-a nema jer za njim više nema

potrebe. U staroj instalacijskoj proceduri išlo se na što višu razinu automatizacije, što za sobom povlači određenu nefleksibilnost. Instalacija nije tražila skoro nikakav input, pa je, primjerice, particioniranje diska bilo fiksno određeno i time suboptimalno za poslužitelje s većim diskom i slično. Na ovaj način je moglo doći do stvaranja premalih / i /var particija, što se kasnije ne može ispraviti na lak način.

Zbog sve većeg broja različitih poslužitelja postalo je teško pratiti različitosti u hardveru, stoga se prešlo na nativnu Debian instalacijsku proceduru. Ona kroz interaktivnost nudi fleksibilnost koju nijedna automatska procedura ne može imati.

Sve što vam treba je .iso image originalnog Debian instalacijskog CD-a. Možete birati između nekoliko opcija: malih 180 MB netinst CD slika, preko 1 CD-a s standardnom distribucijom i najčešćim paketima, sve do potpune distribucije na preko 20 CD-ova ili 4 DVD-a. Sasvim je dovoljno skinuti netinst sliku, a sve ostalo što vam treba instalirati na standradni način preko APT-a. Debian instalaciju za koju se odlučite možete skinuti sa adrese http://www.debian.org/distrib/ ili na ftp.carnet.hr.

Provjerite imate li CARNetove repozitorije u /etc/apt/sources.list datoteci. Ona mora izgledati (minimalno) ovako kako bi mogli doći do naših paketa (a to je ujedno i najbrži put do Debianovih paketa):

Nakon osnovnog podešavanja (IP adresa, ime i slično), sve što treba je skinuti skriptu sa adrese ftp://ftp.carnet.hr/carnet-debian/dists/carnet-etch/carnet-etch.sh i pokrenuti je kao root. Ona nema neku posebnu "inteligenciju", nego će samo povući i instalirati CARNet pakete. Sav posao će odraditi instalacijske procedure CARNet paketa i na taj način ćete dobiti standardni CARNet-Etch poslužitelj.

Imajte na umu da CARNet paketi podrazumijevaju da je poslužitelj na koji se instaliraju glavni poslužitelj ustanove i tako se konfiguriraju (npr. da zaprimaju poštu za cijelu domenu). Zbog toga će neke pakete trebati rekonfigurirati sa "dpkg-reconfigure paket" te još ručno u konfiguracijskoj datoteci ako ima potrebe.

S druge strane, ako samo želite dodatni web poslužitelj, CARNet paketi vam nisu ni potrebni, osim eventualno LAMP paketa (apache2-cn, php5-cn, suhosin-cn, squirrelmail-cn, webalizer-cn, mysql-cn i vjerojatno vsftpd-cn). Slično je i ako samo želite mail poslužitelj.

No, zato u svakoj situaciji preporučujemo instalaciju kernel-2.6-cn paketa, koji je redovito noviji i ažurniji od Debianovog.

Ukoliko želite što sličniji sustav nekom drugom poslužitelju, skinite popis paketa sa tog drugog poslužitelja:

U datoteci paketi.txt će se naći svi paketi s njihovim statusom (instaliran/nije instaliran). Datoteku jednostavno prebacite na novi poslužitelj te napravite:

Ova jednostavna procedura će dodati nove i obrisati stare pakete, ukratko, klonirati instalaciju sa starog Debian sustava.

Na gore opisani način dobijete "CARNet Etch" distribuciju na poslužitelju na kojem prethodno nije bila instalirana nikakva CARNet modificirana inačica operativnog sustava Linux.

Nakon prelaska na novu generaciju Debian distribucije, obično na sustavu ostanu neki programi i biblioteke i programi koje više ne trebamo ili su jednostavno zastarjele. Primjerice, možemo imati i po nekoliko inačica biblioteke libgnutls, libclamav, libdb i mnogo drugih na sustavu. Razlog je taj što novija inačica bilbioteke obično ne briše staru *major* inačicu, slično kao što možete imati dvije različite inačice Office paketa na računalu. Različitim inačicama biblioteka treba dodati i sve one programe koje ste probno instalirali i zaboravili na njih. Iako svaki paket posebno ne zauzima puno mjesta, kad se sve zbroji može se raditi o desecima, pa i stotinama megabajta prostora.

Kako ćemo znati koji su nam paketi višak? Ovdje nam može pomoći naredba **deborphan**:

Popis smo sortirali kako bismo vidjeli koliko starih inačica biblioteka može biti na sustavu, ali i

jednostavno radi preglednosti. No, ne možemo jednostavno uzeti sve ove bilioteke i programe i obrisati ih. Naime, naredba *deborphan* pronalazi sve pakete koji nisu referencirani u nijednom drugom paketu. Ovo ne znači automatski da nam navedeni paket nije potreban na sustavu.

Dakle, potrebno je provjeriti svaki paket pojedinačno treba li vam ili se može obrisati. Uzmimo na primjer paket libgnutls:

Vidimo da imamo noviju inačicu, te sve starije možemo sigurno obrisati:

No, imamo i ovakvih slučajeva:

Dakle, imamo samo jednu biblioteku *libident* o kojoj nije ovisan ni jedan drugi paket. Ovdje problem možete riješiti jedino vi sami, koji ste paket instalirali i u ovisnosti o trenutnim potrebama paket obrisati ili ga ostaviti. Imamo i treći, slučaj tranzicijskih paketa (ovdje shellutils, textutils i fileutils):

Iz ovoga možemo vidjeti da si možemo pomoći opcijom "-s" kako bismo vidjeli puni opis paketa, odnosno za što služi. Iz opisa vidimo da seu se ovi paketi zamjenili jednim (*coreutils*), te ih možemo slobodno obrisati:

Nisu svi tranzicijski paketi ovakvi (neki moraju ostati na sustavu), pa morate pročitati opis paketa kako bi došli do točne informacije.

Na kraju, ono što vam dosad nismo rekli je da naredba *deborphan* po *defaultu* ispisuje samo biblioteke, ne i programe u drugim kategorijama. Da bi vidjeli zaista sve pakete o kojima ne ovisi nijedan drugi paket, morat ćete upotrijebiti opciju "-a":

Iz ovog (jako skraćenog) popisa jasno je da je potreban daleko veći oprez, jer možete obrisati neki program koji je i dalje potreban. Zato, prije brisanja napravite *backup*, te u krajnjem slučaju ako niste sasvim sigurni ostavite program na sustavu.

Prije skoro dvije i pol godine smo pisali kako osloboditi "/" particiju (*root* particiju, nemojte pobrkati sa HOME direktorijem korisnika root, /root!) i time napraviti prostor za nove inačice jezgri sustava - kernel. No, od tada smo napustili LILO *bootstrap loader* u korist GRUB-a, te vlastiti kernel u korist originalnog Debianovog. Tako je članak "Kako osloboditi / (root) particiju za novi kernel?" postao pomalo zastario, a sudeći po upitima na SysHelpu, problem je i dalje ostao. Iz tog razloga, ponovit ćemo lekciju i nastojati pobliže objasniti problem, bez obzira rabili LILO ili GRUB, jedan ili više kernela.

U osnovi, problem je veličina samog paketa i odgovarajućih modula, koji su trenutno oko 60 MB i relativno male / particije. Dakle, veličinu / particije držite na barem 512 MB, a ukoliko je manja, morat ćete poduzeti neke korake da bi se problemi izbjegli (bilo brisanjem nepotrebnih datoteka, bilo reparticioniranjem diska).

Problem se manifestira kod instalacije paketa s novim kernelom, primjerice ovako:

Porukom "**No space left on device**" sustav nam javlja da na odgovarajućoj particiji nema dovoljno prostora za snimanje novih podataka. Direktorij "/lib/modules" se nalazi na / particiji, te je ujedno i glavni razlog zašto se particija zapuni (sam kernel je manji od 2 MiB).

Pogledajmo što se sve u /lib/modules (moduli jezge) i /boot direktoriju (jezgra i pomoćne datoteke) može nalaziti:

A sada i koliko mjesta zauzimaju same jezgre:

Može se vidjeti da jezgre zauzimaju relativno malo mjesta, no ukoliko ste isti poslužitelj nadograđivali, s vremenom će se to sve zbrojiti i nepotrebno zauzimati mjesto.

Prvo što možemo učiniti je bezopasno: provjerite ima li u / direktoriju nepotrebnih velikih datoteka i obrišite ih. Direktorij / treba izgledati otprilike ovako:

Dakle, nema nikakvih datoteka, nego služi samo kao *mount point* za ostatak datotečnog sustava i slično tomu. No, nekada se ovdje nenamjerno znaju naći datoteke koje samo zauzimaju prostor, bilo

vašom greškom, bilo da neki program rabi / direktorij ukoliko ne može naći svoj radni direktorij (a znaju ovamo zalutati i .avi, .mp3 i .pdf datoteke).

Ukoliko nema nepotrebnih datoteka, moramo preći na opasniju fazu: brisanje nepotrebnih jezgri i odgovarajućih modula. Skratit ćemo postupak i reći da možete obrisati sve jezgre ispod 2.6.26 ukoliko rabite Debian Lenny, te sve jezgre ispod 2.6.32 ukoliko rabite Debian Squeeze. Ukoliko niste sigurni koju jezgru rabite, poslužite se naredbom **uname**:

Također, u /boot/grub/menu.lst provjerite koja će se jezgra automatski učitati nakon restarta poslužitelja, odnosno koliko jezgri uopće imate u izborniku ("default 0" označava da će se učitati prva spomenuta jezgra):

Nadalje, provjerite koje su jezgre instalirane, jer ne moraju sve biti navedene u menu.lst.

Jezgre koje su instalirane, a ne rabe se, niti spominju u menu.lst, možemo odmah obrisati:

U nekim slučajevima (primjerice, netko je instalirao jazgru direktno iz koda, pa se ne pojavljuje u bazi paketa), potrebno je ručno brisanje. **Ovo nije bezazlena operacija, jer je greška lako moguće, stoga budite oprezni!**

Brisanje jezgri koje se ne pojavljuju u paketnom sustavu, niti u menu.lst se provodi ovako:

Nakon toga možete obrisati i njihove module:

Budite oprezni kod izvođenja naredbe "rm", provjerite u kojem ste direktoriju pomoću naredbe "pwd"!

Dalje je jednostavno:

ili

Sada bi se nove jezgre trebale bez problema instalirati, no nakon svake nadogranje na novu inačinu Debiana provjerite ima li nepotrebnih datoteka ili jezgri viška. Naravno, sasvim je u redu, dapače čak i pametno, ostaviti staru jezgru s kojom vam sve radi, kako biste mogli u svakom trenutku podići poslužitelj ukoliko se ustanovi problem s novom jezgrom. Naravno, ukoliko imate mjesta na / particiji :).

OVAJ ČLANAK JE ZASTARIO, POGLEDAJTE ČLANAK /sys-trek-objave/best-of-sys-help/#kako-osloboditi-root-particiju-za-novi-kernel-2**!**

Stariji poslužitelji imaju root particiju veličine 250 MB, što nakon nekoliko iteracija instaliranja kernel-cn paketa više nije dovoljno. Kernel-2.6-cn nikad ne briše starije inačice iz /boot direktorija, samo generira novi /etc/lilo.conf a kernele i njihove module ostavlja na miru. Kako donosi čak četiri inačice kernela, sveukupno zauzeće prostora je nešto malo preko 101 MB (iako svaka inačica kernela zauzima oko samo 1.5 MB).

Pokažimo to primjerom s jednog starijeg Compaq ML310 poslužitelja:

Na / particiji postoji samo 2.4 MB slobodnog prostora, što je očigledno nedovoljno za instalaciju novog kernela (potrebno je nešto malo preko 101 MB). Pogledajmo što se sve u /boot direktoriju nalazi:

Može se vidjeti da kerneli zauzimaju relativno malo mjesta. Daleko više mjesta zauzimaju kernel moduli:

Što smijemo brisati? U najkraćim crtama, možemo sve. Naravno, u tom slučaju moramo odmah instalirati novi kernel jer reboot poslužitelja u suprotnom više neće biti moguć. Dakle:

Gornja naredba će obrisati aktualni i eventualni zaostali 2.4 kernel, ali stariji kerneli (o kojima paketni sustav ne zna ništa) će ostati. Zato moramo ostatak pobrisati ručno:

Nakon toga možete obrisati i module:

Oprez kod izvođenja gornje naredbe, morate biti u /lib/modules direktoriju!

Dalje je jednostavno:

Ukoliko imate problema s apt-om, skinite kernel-cn sa ftp.carnet.hr i instalirajte ga ručno:

Gornje upute podrazumijevaju da je sa sustavom sve u redu. Ukoliko nije, nemojte ostavljati sustav bez kernela i ostavite trenutno radeći u /boot, uključujući i njegove module u /lib/modules.

Kernel koji će se startati sa svakim bootom je zapisan u /etc/lilo.conf datoteci:

Može se vidjeti da je labela "linux" navedena kao default kernel koji će se startati kod pokretanja sustava ("default=" redak).

Ukoliko default kernel nije oznacen kao "linux", nego nekako drugačije, jednostavno na sustavu ostavite kernel na koji pokazuje "default=" redak.

Moramo li napomenuti da novi kernel neće biti aktivan sve dok ne restartate poslužitelj?

Sigurno ste ponekad primjetili u kernel log datoteci zapise poput ovoga:

Radi se o SYN flooding napadu koji u pravilu nema neki utjecan na dobro konfiguriranu mrežu i poslužitelj. SYN flooding radi na način da poslužitelju pošalje SYN requests koji nikad ne završi sa ACK te na taj način poslužitelj koristi mrežni slot čekajući drugu stranu za odgovor. Šaljući sve više SYN flooding napada mrežni resursi postaju popunjeni, te poslužitelj nakon nekon vremena postaje neupotrebljiv.

Simptomi koji se javljaju sa strane krajnjeg korisnika su sporo i dugo učitavanje web stranice. Moguće je da se učita samo dio stranice.

Problem na koji sistemac može naići je da prilikom SYN flooding napada ne može primjetiti visoku opterećenost CPU-a.

Način na koji možemo primjetiti da se radi o SYN floodin napadu da iskoristimo naredbu time u kombinaciji sa wget i vidjeti koje je vrijeme potrebno za učitavanje.

Učitavanje je trajalo svega 7 milisekundi što je u ovom primjeru sasvim normalno.

U usporedbi kad se radi o SYN floodin napadu rezultat je puno drugačiji:

Ovdje vidimo da proces učitavanja nije još završio, poslužitelju treba dosta vremena da otvori mrežnu utičnicu
te moramo pričekati krajnji rezultat.

Nakon podužeg čekanja dobijemo rezultat:

Primjetit ćemo da je potrebno 52 milisekunde za (lokalno) učitavanje.

Pomoću naredbe netstat izvršit ćemo provjeru konekcija na poslužitelj:

Kao rezultat dobijemo hrpu SYN request zahtjeva sa iste IP adrese:

Način na koji se možemo zaštiti je uključivanje SYN cookies na poslužitelju pomoću naredbe sysctl.

U konfiguracijsku datoteku /etc/sysctl.conf moramo dodati redak:

kako bi ova funkcija radila i nakon ponovnog startanja poslužitelja. Na CARNetovim poslužiteljima je ovaj redak već dodan, ukoliko rabite paket kernel-2.6-cn.

Već smo objavili članke o tome kako se obraniti od prevedenih spamova, odnosno, u slučaju webmaila, kako detektirati preko čijeg se korisničkog računa šalje spam. Izravan povod su upiti na sys.help, na adresi syshelp@carnet.hr. Ovim člankom probat ćemo zaokružiti priču, opisati kako otkriti način "provale", kako se šalju spamovi i kako se skinuti sa crnih lista.

Upiti koji nam dolaze najčešće su potaknuti pritužbama korisnika da ne mogu slati mail na određene adrese, najčešće na velike mail providere tipa Gmail i Yahoo. U logovima se lako može pronaći URL na kojem se može zatražiti uklanjanje s crne liste, ali ako se ne pronađu krivci i ne ukloni uzrok vaše će računalo brzo biti vraćeno na crnu listu.

Iz vlastitog iskustva zaključili smo da su tri najčešća načina *spamiranja*i:

1. slanje spama preko webmaila

2. slanje spama uz pomoć SASL-a (SMTP AUTH)

3. slanje preko zaraženog računala iz lokalne mreže

Pa kako to spamer može slati mail iz naše mreže, na primjer preko webmaila? Lako, ukoliko otkrije korisnikovu zaporku. Najčešće je otkriva socijalnim inženjeringom, pomoću lažnih mailova za koje se čini da potječu od administratora sustava ili nekog servisa. Takvi su mailovi najčešće prevedeni, i to sve bolje i bolje. Njih smo obradili u prvom članku iz serije.

Drugi način zloporabe korisnikove zaporke je pomoću SASL mehanizma, koji omogućava korisnicima da šalju mailove iz vanjskih mreža. Uporaba je jednostavna, samo treba u mail klijentu označiti da želimo SMTP autentikaciju i mailovi već prolaze. Ovaj način autentikacije smo obradili u članku /sys-trek-objave/postfix-savjeti-i-trikovi-zeljko-boros/#sasl-smtp-autentikacija-u-postfixu.

Zadnji, i možda najjednostavniji način je slanje spama sa zaraženog računala u vašoj lokalnoj mreži. Za to nije ni potrebna zaporka, jer će mail poslužitelj bez autentikacije primiti mail s računala iz lokalne mreže i proslijediti ga. Današnji spamerski alati znaju pročitati postavke iz mail klijenta i ponašati se poput normalnih mail klijenata, umjesto da izravno šalju mailove na port 25, što je lako spriječiti na vašem usmjerivaču gdje se to dozvoli samo legalnom mail serveru.

Problem je kako detektirati što se događa, odnosno čija je zaporka provaljena. To nije trivijalno jer je, kako smo opisali, zloporaba moguća na više načina.

Najproblematičnijim se pokazao webmail, jer je logiranje rudimentarno. Problemu smo doskočili instalacijom dodatnog modula (Squirrel Logger), s kojim se definira što će se i kada zapisivati. Na taj način se "iz aviona" vidi tko je odgovoran za problem. Modul smo opisali u spomenutom članku.

Ukoliko su spameri posegnuli za SASL autentikacijom, zadatak će vam biti olakšan činjenicom da tu autentikaciju ne rabi mnogo korisnika, osim u slučaju da ste sustav tako konfigurirali. Zbog toga su unosi u logovima prilično uočljivi, te ih je lako usporediti sa spamovima. CARNetov CERT će vam poslati kopiju spama, zajedno sa svim zaglavljima.

Unosi u logovima izgledaju poput ovih:

Dakle lako je vidjeti koji korisnik se autenticirao, što nam olakšava posao.

Ostaje nam zadnji slučaj, kada imamo zaraženo računalo u mreži. U tom je slučaju situacija "šarolika" i ovisi o tipu programa/virusa/crva kojije zarazio računalo. Najlakši je slučaj kad računalo šalje mnogo mailova, što ga čini uočljivim u logovima. Tome se može doskočiti stavljanjem više primatelja u **To** ili **Cc** polje, što smanjuje broj konekcija prema Postfixu. Tu dolazimo do granice kada više ne možemo pružiti nekakav konkretniji savjet, nego moramo uopćiti cijelu priču. Pa krenimo...

U istraživanju si možete pomoći tako da pogledate vrijeme slanja mailova, kao i "*message id*" oznaku. Pronaći ćete ih u zaglavlju spama, kojeg će vam poslati druga strana ili CERT, ponekad i prije nego dospijete na crnu listu. Nemojte odmah tražiti po **From** polju, jer se to lako lažira. Za tim posegnite kada više nemate izbora i valja se hvatati za slamku.

Ukoliko i dalje ne možete pronaći ništa sumnjivo, vrijeme je za malo detektivskog rada. Kad nađete kandidate među korisnicima (bilo preko IP adrese, bilo preko *usernamea*), nazovite ih i pitajte jesu li slali kakve mailove, prema kome i koliko puta. Možda su u to vrijeme bili čak odsutni od računala, što je definitivno dokaz da je njihovoračunalo zaraženo.

Moguće je da i nakon toga ništa ne nađete. Spameri mogu na mail poslužitelju pomoću ranjivosti web sučelja instalirati programe koji služe za relay portova i time zaobići mnoge sigurnosne sustave. U ovakvim slučajevima pomažu vatrozidi i Intrusion Detection sustavi, ali to prelazi granice ovog članka.

Što uraditi nakon svega, kada se situacija vrati u normalu, a računalo je brisano s crne liste? Uložite

trud u obrazovanje vaših korisnika. Objasnite im da nikada nećete tražiti njihovu zaporku preko maila, da ne klikaju na razne linkove navedene u nemušto prevedenim mailovima. Održite seminar, pošaljite upozorenja, zalijepite obavijest na hodniku. Konkretni postupci ovise o lokalnim navikama i očekivanjima vaših korisnika, a bitan čimbenik je i veličina ustanove na kojoj radite.

Nadamo se da smo s ova tri članka bar malo pomogli rješavanju problema spama.

U proteklom tjednu smo na Helpdesku za sistemce zaprimili dosta upita zbog "kašnjenja", odnosno "vraćanja" mailova s popularnih besplatnih e-mail servisa Google i Yahoo. Problemi su bilo vrlo slični, ali su logovi pokazivali i određene razlike u konfiguracijama na poslužiteljima. No, svima je zajedničko to što SSL sloj nije bio dobro podešen u Postfixu, te primanje i slanje maila preko SMTPS protokola nije bio moguć, ili je bio otežan.

Probleme su uglavnom imali kolege s nepodešenim SSL/TLS slojem, ili je Postfix bio podešen da rabi krive certifikate. Oni su mogli dobiti poruke u logovima ovog tipa:

Oni koji su manje-više dobro podesili Postfix rabeći Comodo certifikate, mogli su usprkos tome dobiti ovakvu poruku o greški (njih ćemo svrstati u drugu skupinu):

Navest ćemo jednostavan recept za one iz prve skupine, kojima SSL zapravo uopće ne radi kako treba:

1. obrisat ćemo stare (pretpostavljamo da su neispravni ili zastarjeli) i napravimo nove samopotpisane (*self-signed*) certifikate:

2. u datoteci /etc/postfix/master.cf provjerimo postoji li redak:

Ukoliko ne postoji, trebate upisati točno kako piše umjesto postojećeg "smtps" retka (sve treba upisati u jedan redak!)

3. u datoteci /etc/postfix/main.cf redak

zamijenimo sa

4. još treba provjeriti jesu li u main.cf uključene dvije varijable

Oblik varijabli **smtp_use_tls** i **smtpd_use_tls** je od Postfixa 2.3 zastario, ali se još može rabiti.

5. sada možemo restartati Postfix:

ili pravilnije

6. provjerimo radi li SSL/TLS:

Rezultat bi morao biti sličan ovome:

Ostaje nam navesti rješenje za one koji imaju uredno podešen SSL (ili čak i Comodo certifikate), ali svejedno imaju problema sa Gmailom. U ovom slučaju problem je u tome što Postfix iz nekog razloga ne zna za root Certificate Authority (CA) certifikate za Equifax. Postfixu možemo reći gdje su certifikati na dva načina, a mi smo odabrali način kad su svi certifikati u jednoj velikoj datoteci, **ca-certificates.crt**. U main.cf upišite:

Nakon ovoga, kao i u prethodnom slučaju, treba restartati Postfix. Pošaljite testne mailove, a ukoliko se greške u logovima (**mail.log**, **mail.err**) i dalje javljaju, probajte skinuti root CA certifikat direktno:

I naravno, opet treba restartati Postfix. Sada bi trebalo raditi, a ukoliko to ipak nije slučaj, treba vidjeti što piše u logovima kod slanja maila, jer možda je kod vas potrebno još nešto podesiti.

Dakle, zaista je krajnje vrijeme da podesite svoj poslužitelj tako da ima ispravne certifikate, a time i zaštićene i vjerodostojne servise prema klijentskim računalima. U tome neka vam pomogne članak na adresi /sys-trek-objave/best-of-sys-help/#certifikati-za-servise, dok Comodo certifikat možete zatražiti preko stranice http://www.carnet.hr/sc_servis .

Logovi su nezamjenljivi dio gotovo svakog operativnog sustava. U njima se mogu pronaći sve relevantne informacije o prošlom i, bitnije, trenutnom stanju sustava, te je njihova kontrola jedna od

stvari koju svaki sistem-inženjer mora svakodnevno obavljati. U tome nam mogu pomoći alati poput OSSEC-a ili logchecka, ali ručni pregled se ne može ničim zamijeniti.

Na Linux sustavima log zapisi se nalaze u direktoriju /var/log (uz možda poneki izuzetak, no to nam u ovom trenutku nije važno). U slučaju da naiđete na bilo kakve probleme u radu sustava ili njegovih servisa, prvo mjesto gdje trebate pogledati je upravo tamo, i to u datoteci /var/log/syslog i /var/log/messages.

Iako su logovi obične tekstualne datoteke koje se mogu otvoriti u editoru ili naredbom less, vrlo često se one pregledavaju naredbom tail:

Naredba tail će prikazati zadnjih 10 redaka u datoteci. Ukoliko želite vidjeti manje ili više redaka (primjerice, pet), upotrijebite naredbu tail na ovaj način:

No, logovi su datoteke koje se konstantno "pune" s novim unosima. Tako ćete pregledavanjem logova uvijek biti u malom zaostatku za aktualnim zbivanjima na sustavu. U ovom slučaju može vam pomoći opcija "-f". Uporabom ove opcije tail će konstantno nadzirati datoteku, i ispisati svaki novi redak koji se upiše u datoteku dok je pregledavate:

Sličan rezultat možete postići i preko naredbe less, ukoliko pritisnete "<SHIFT> + f" dok pregledavate datoteku:

Izdvojit ćemo najbitnije log datoteke iz /var/log direktorija, te što se u njih upisuje:

1. **auth.log** - poruke vezane uz autentikaciju korisnika
2. **daemon.log** - poruke vezane uz daemon procese na sustavu
3. **debug** - poruke razine debug (vidi dolje)
4. **syslog** - većina drugih poruka se zapisuje i ovdje
5. **kern.log** - poruke vezane uz jezgru sustava

Ove datoteke su konstantno otvorene za pisanje, a u njih zapisuje daemon syslogd. Ponašanje ovog daemona možete kontrolirati preko njegove konfiguracijske datoteke, /etc/syslog.conf. Konfiguracija je malo specifična, jer uvodi pojam klase i razine.

Svaka moguća poruka koja ide prema syslogu mora imati svoju klasu ("facility") i razinu ("severity"). Klase na Linuxu su redom: **auth**, **authpriv**, **cron**, **daemon**, **ftp**, **kern**, **lpr**, **mail**, **mark**, **news**, **security** (isto što i auth), **syslog**, **user**, **uucp** i **local0** do **local7**.

Moguće razine su: **debug**, **info**, **notice**, **warn**, **err**, **crit**, **alert**, **emerg** i **panic**. Neke nećete nikada rabiti, a uporaba klasa mark i security, te razina warn, err i panic se više ni ne preporuča (ili su za internu uporabu, kao primjerice mark).

U /etc/syslog.conf se podešava u koje datoteke se poruke određene klasifikacije i razine upisuju, odnosno hoće li se i gdje zapisivati, jednostavno ignorirati, slati na druge syslog poslužitelje i slično. Primjerice, standardne postavke za Debian su:

Možemo vidjeti da je oblik retka u konfiguracijskoj datoteci:

Umjesto klasifikacije ili razine možete staviti zvjezdicu. Zvjezdica obuhvaća sve moguće klasifikacije za određenu razinu, i obrnuto.

Preko primjera je najlakše objasniti kako to sve funkcionira:

Ovim konfiguracijskim retkom smo odredili da se sve poruke klase "mail" upisuju u datoteku /var/log/mail/mail.log, bez synca. Ukoliko imate prometan mail poslužitelj, možete razbiti ovaj log na više manjih:

Na ovaj način možete brže doći do željenih informacija, ali u uobičajenom radu CARNetovih poslužitelja najčešće nije potrebno ovo raditi.

Razine u syslogu su hijerarhijske, dakle na najnižoj razini je "debug", a na najvišoj "emerg". Dakle

redak

zapravo znači "upiši sve razine od err i više: err, crit, alert i emerg"

Postoje još neke dodatne oznake. Crtica na početku imena datoteke se rabi kad ne želimo raditi sinkronizaciju datoteke sa stanjem na disku svakim upisom u nju (ne želimo raditi "sync", više informacija o ovome možete dobiti sa man syslog.conf).

Oznaka "=" označava da želimo filtrirati baš tu razinu, dakle ne i razine iznad navedene, dakle "=info" će zapisati samo poruke na toj razini. Ovdje se može dodati još i negacijski modifikator "!". Vjerojatno pogađate što će se dogoditi: bit će upisane sve razine osim navedene.

Dakle, "mail.!=warn" će upisati sve razine osim razine "mail.warn".

Ukoliko niste sigurni u koju datoteku neki servis zapisuje logove, pogledajte sljedeći redak:

Oznaka "*.*" znači da se sve klase i razine zapisuju i u datoteku /var/log/syslog, pa prvo pogledajte tamo. Ako ipak želite posebnu datoteku, nađite u manualu vašeg servisa (npr. tcpd/tcp_wrappers) koju klasu rabi, te upišite u /etc/syslog:

Još bolje, tcpd podržava promjenu klase i razine, pa možete rabiti "local0" do "local7" klase kako se logovi ne bi miješali (default je klasa "auth", pa će biti upisivani i drugi događaji u toj klasi). Primjerice, ako u /etc/hosts.allow imate:

a u /etc/syslog.conf:

onda će sve iz klase "local7.notice" biti zapisano u datoteku tcpd.log.

Ne zaboravite restartati syslog kako bi promjene bile pročitane.

Kako poslužitelji rade 24 sata dnevno, log datoteke mogu postati jako velike. Kako administrator ne bi morao ručno brisati, odnosno skraćivati log datoteke, obično se rabi alat logrotate.

Logrotate se može konfigurirati tako da rotira, odnosno sažima stare i kreira nove, prazne log datoteke. Također se može konfigurirati tako da čuva stare logove samo određeno vrijeme, rotira logove samo određene veličine i slično. Na taj način disk na poslužitelju nikada neće biti zapunjen starim logovima.

Logrotate smo detaljnije opisali u članku: Logrotate - zaboravljeni junak

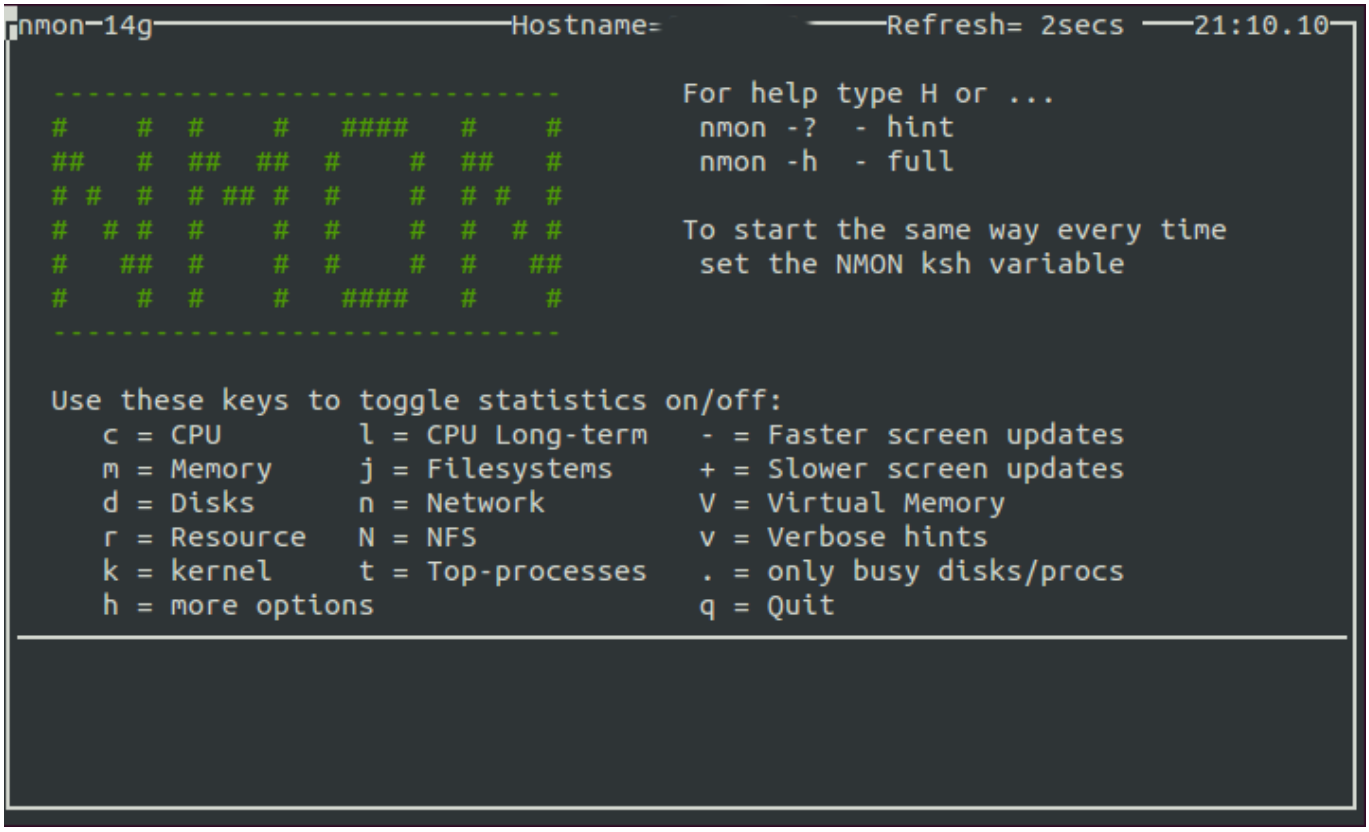
Syslog može još dosta toga, no o tome drugi put.

Postoje mnogi alati koji nam mogu prikazati stanje procesora i memorije, zauzeća diskova i ostalih parametara. Neke od njih smo već susreli u člancima na ovom Portalu, poput nethogs ili iftop. Nmon (nemojte ga brkati s nmap!) na prilično jednostavan način pokazuje gotovo sve parametre koji vam mogu zatrebati kada želite vidjeti ponašanje sustava zbog nekog problema ili slično.

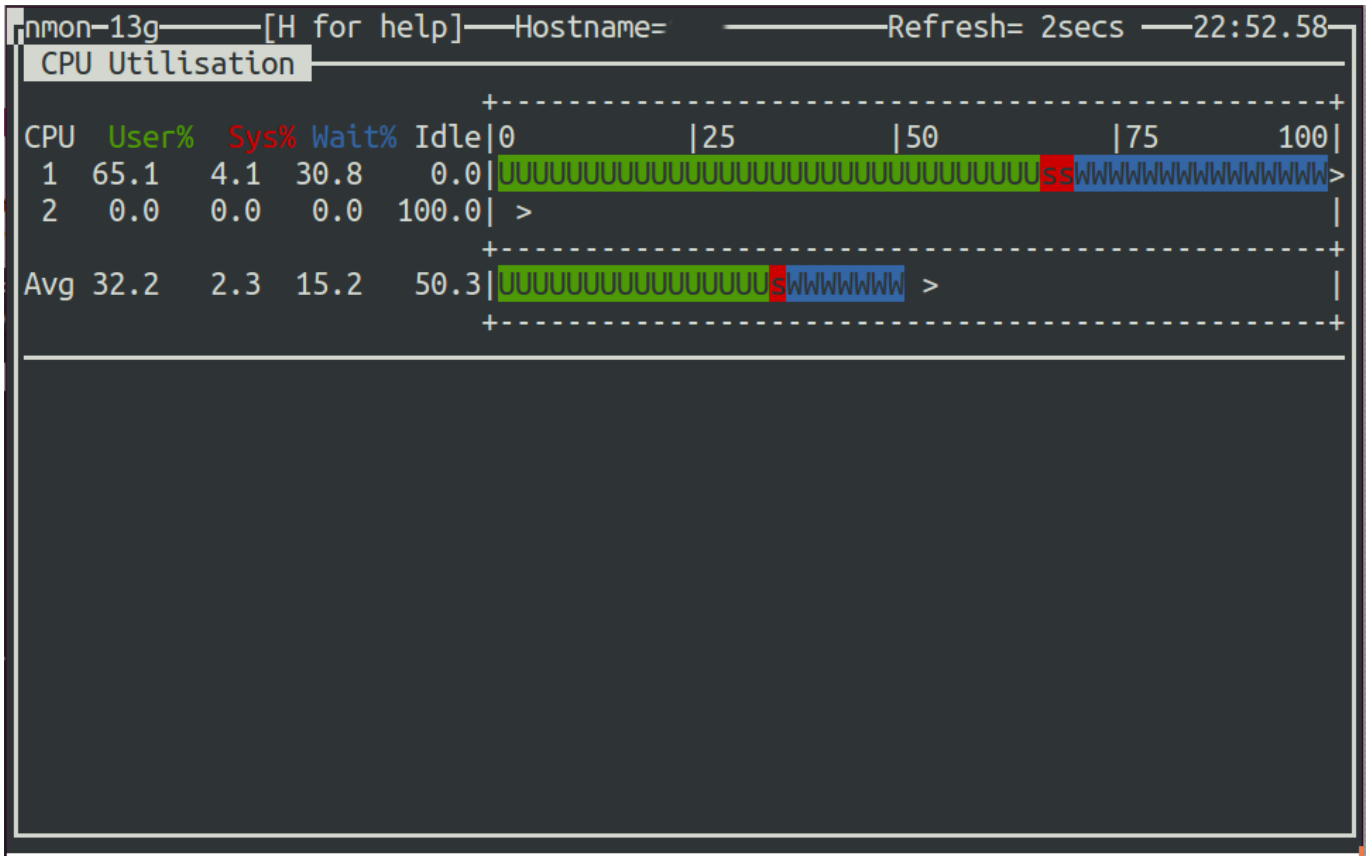
Iako je alat namijenjen sistemcima-početnicima koji ne vole previše komandnu liniju, koristan je svima. Preko ncurses sučelja pokušava vizualizirati podatke koje prikazuje, što je atraktivno (za komandnu liniju), ali i korisno.

Nmon može prikazati podatke o:
- zauzeću memorije
- procesoru
- top procesima
- mreži
- diskovima
- filesystemima
- općim resursima i verzijama

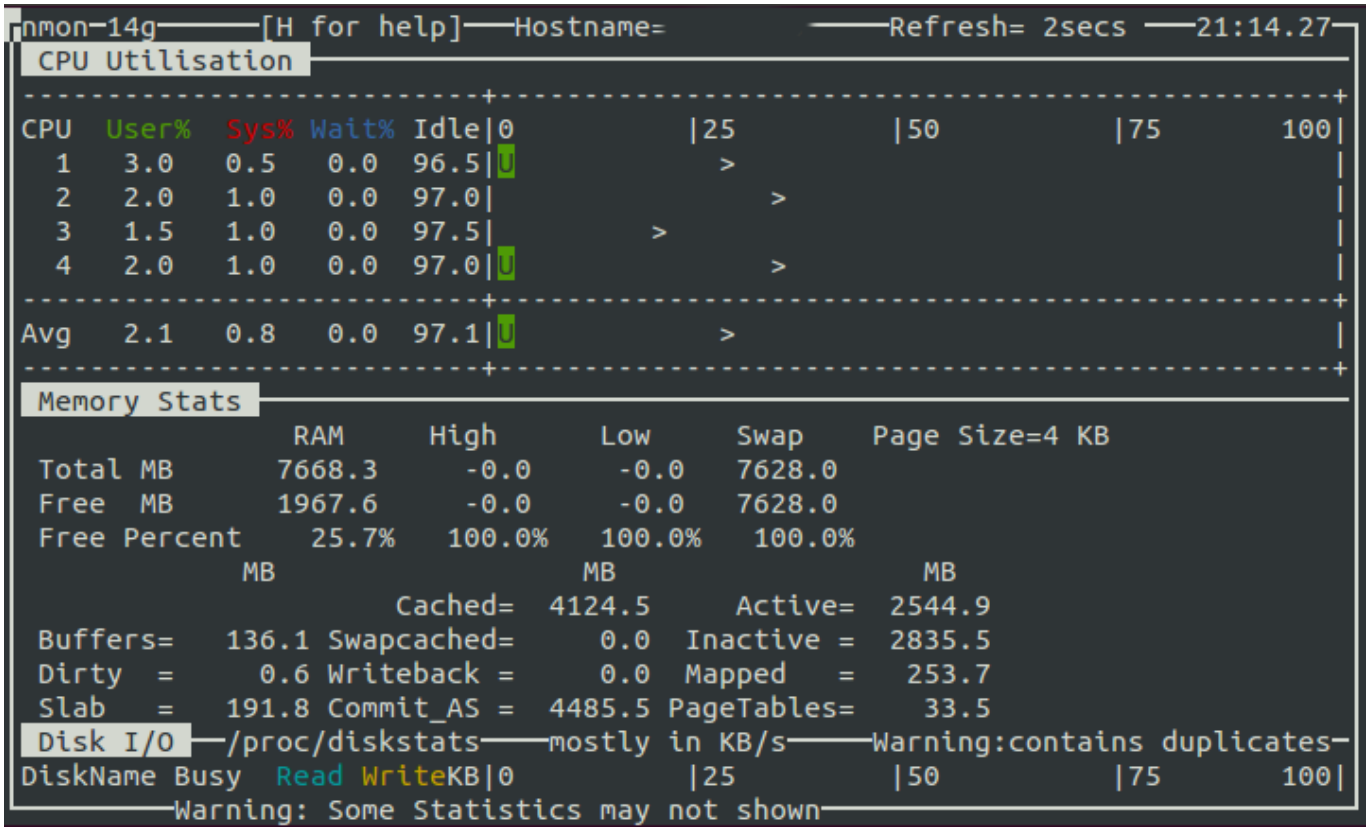
Do svake opcije se dolazi iz glavnog izbornika:



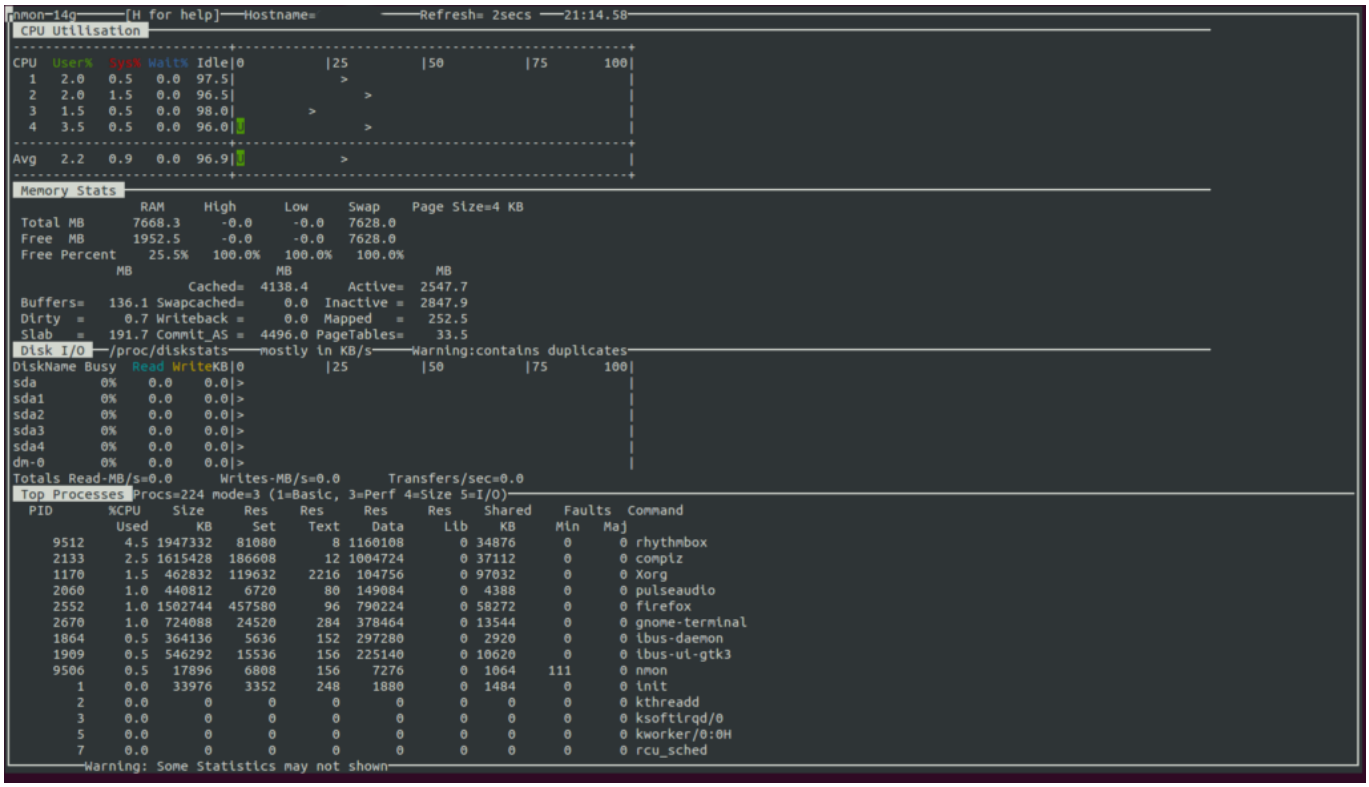
Kada odaberete opciju za, primjerice, podatke o CPU (c), pokazat će se podaci za svaki od procesora koje imate. U desnom dijelu ekrana pokazat će se i grafikoni, odnosno bar grafovi u boji.



Ukoliko odaberete neku drugu opciju, primjerice mrežu (m) i disk (d), blok podataka za CPU neće nestati, nego će se svaka nova opcija nadodati na dno.



Kako je očigledno da svi ovi podaci ne mogu stati na ekran veličine 80x24 karaktera, na dnu ekrana pojavljuje se poruka "*Warning: Some Statistics may not shown*". Problem se lako rješava maksimiziranjem prozora vašeg terminalskog emulatora.



Da ne morate pamtiti sve opcije, dovoljno je zapamtiti da tipka "h" (help) pokazuje sve tipke koje možete rabiti, no ovaj popis je malo proširen u odnosu na početni ekran. Sve module gasite ako ponovo pritisnete istu tipku s kojom ste ga uključili.

Defaultna brzina osvježavanja je 2 sekunde, a s tipkama "+" i "-" udvostručujete, odnosno prepolovljujete ovo vrijeme. S tipkom "t" ulazite u možda najpotrebniji modul, top procesi. Najsličniji je standardnoj naredbi top, a iako nema napredne mogućnosti sortiranja, ono je moguće po nekoliko kriterija. Sasvim dovoljno da se pogleda je li neki proces "podivljao".

Izlaz iz programa je naravno s tipkom "q".

Ukoliko ne želite svaki put gledati uvodni ekran, možete unaprijed upisati module koje želite automatski startati u varijablu NMON na ovaj način:

Dakle, ukoliko želite imati alat tipa "švicarski nožić", ovo je sasvim pristojan predstavnik tog tipa alata.

Dokumentacija u obliku man stranice zapravo i ne postoji, ali se pregled svih opcija može vidjeti s "**nmon -h**". Bolju dokumentaciju možete naći online na adresi http://nmon.sourceforge.net.

Jedna od novotarija koje smo pripremili za **tekuću** (verziju 2.x, baziranu oko Debian Sarge-a) i **novu** (verziju 3.x, baziranu oko Debian Etch-a) CARNet Debian distribuciju je monit-cn paket. Monit je sam po sebi iznimno koristan servis za nadzor drugih servisa -- on u predefiniranim intervalima prolazi po listi definiranih resursa (najčešće drugih servisa) te provjerava njihovu ispravnost i poduzima neke predefinirane akcije.

Primjerice ako imamo sljedeću konfiguraciju za Dovecot IMAP4/POP3 servis:

Monit će u tom slučaju svakih n sekundi provjeriti postoji li uopće navedena PID datoteka. Ako ne postoji servis će biti (za svaki slučaj) stopan pa startan koristeći definiranu init skriptu. Ako servis postoji te njegov PID u listi procesa odgovara onome iz datoteke, slijede ostali testovi -- monit ce popričati sa adresom 127.0.0.1 i portom 143 koristeći IMAP protokol, a zatim s portom 110 koristeći POP3. U slučaju da i jedan od tih protokola dobije grešku od nadziranog servisa ili se servis ne uopće odazove, monit će restartati servis. Na ovaj način je moguće definirati cijelu listu servisa i provjera ispravnosti, čime omogućavamo kvalitetniji aplikativni nadzor od onog sto nudi tipični System V init sustav. Da bi pojednostavili konfiguriranje liste postojećih servisa na CARNet-Debian posluzitelju, napravljen je **monit-cn** paket. On će prilikom instalacije provjeriti sto od tih servisa postoji (odnosno što je od njih instalirano i aktivno) na sustavu, popunit će /etc/monit.d/ direktorij sa konfiguracijama i startati monit. Osim toga, registrirati će ga kao poseban "respawnable" servis (koji ne može "umrijeti", već se uvijek sam sam ponovo pokrene) koji nadzire ostale detektirane servise na sustavu. Tipično podržani servisi koji se mogu nadzirati su:

Oko rekonfiguriranja monita (promjena monit init skripte, registracije u inittab i sl.) i njegovog načina startanja **nije** potrebno **ništa** ručno mijenjati, već automatika sve obavlja umjesto sistemca:

U slučaju da kasnije doinstalirate neki servis, detekciju podržanih servisa uvijek možete ponovo pozvati -- a ona će pametno zaključiti što je već bilo aktivno, što nije i da li je eventualno potrebno restartati monit:

Na ovaj način možete riješiti problem sa povremenim padovima amavisd servisa ili nekim ostalim standardno problematičnim servisima. Standardna monit-cn konfiguracija provjerava servise svakih 900 sekundi, te u slučaju detektiranih (i riješenih) problema šalje e-mail na root@localhost o tome. Provjeriti da li se monit ispravno pokreće možete s naredbom pidof:

U slučaju da pidof ne vrati nikakav ispis, znači da postoji neka greška u konfiguracijskim datotekama od monita, pa je najjednostavnije ustanoviti grešku na sljedeći način (npr. ovdje je nehotična greška to što ime protokola nije pop3, već samo pop):

U ovom slučaju vidimo da je problem do greške u dovecot.conf datoteci, pa je dovoljno rečenu obrisati (ili popraviti grešku u sintaksi), te poslati initu signal da ponovo pročita inittab:

OSSEC je multiplatformski sustav za detekciju napada otvorenog koda, i već dulje vrijeme je u ponudi CARNetovih paketa. U potpunosti je spreman za autonoman rad nakon instalacije, a za početak je dovoljno upoznati se s njegovim pravilima ("*rules*"), točnije kako ignorirati pravila koja

generiraju lažne pozitive, te dodati svoja vlastita pravila. Kako ignorirati određena pravila smo ovlaš spomenuli u članku /sys-trek-objave/bind-dns-savjeti-i-trikovi/#bind-ignoriranje-pojedinih-logova-u-lennyju, koji se tiče ignoriranja određenih unosa u logovima koje generira BIND. No, postoji mnogo više mogućnosti od ove.

U članku smo naveli ovaj primjer:

SID koji se spominje u direktivi **<if_sid>** se odnosi na broj pravila na koje želite utjecati, a njega ćete dobiti u mail poruci, npr. "Rule: 1002 fired" . Vi dodavanjem ovog pravila pravite "*child rule*", preko kojeg možete utjecati na krajnji rezultat (koji bi se dogodio da niste uveli svoje pravilo).

Što se tiče određivanja vašeg vlastitog broja, morat ćete poštovati odluku razvijatelja da korisnička pravila rabe brojeve veće od sto tisuća (100000).

Svoja pravila upišite u datoteku **/var/ossec/rules/local_rules.xml**. Kako je tamo već dodana CARNetova modifikacija ugrađenih pravila, nemojte upisivati ništa unutar bloka "**Begin update by CARNet...**" i "**End update by CARNet...**".

Svoja pravila stavite unutar **<group></group>** oznaka ispred ili iza CARNetovog bloka. Ukoliko grupa ne postoji, možete kreirati svoju:

Još samo trebate zapamtiti da brojevi pravila moraju biti jedinstveni, a dalje ih samo nanižite numerički jedan iza drugog, povećavajući broj za jedan, 100001, 100002 itd.

Isto kako možemo modificirati postojeća pravila unutar OSSEC-a, tako imamo i mogućnost da dodamo svoja vlastita pravila, u istoj datoteci gdje smo upisivali modifikacije postojećih pravila (local_rules.xml).

OSSEC zapise u logovima raščlanjuje na sljedeći način:

time -> Jun 5 11:42:39
hostname -> posluzitelj
program_name -> saslauthd
log -> pam_unix(smtp:auth): authentication failure; logname= uid=0
euid=0 tty= ruser= rhost=

Nakon ovoga, moguće je dodatno dekodiranje (definirano preko decoders.xml) pa možete dobiti još ključnih riječi, poput srcip, dstip, id, srcport i tako dalje. Za većinu slučajeva, bit će dovoljno i ovo što imamo do sada. Nakon rašćlanjivanja dobit ćemo ključne riječi, odnosno varijable s kojima možemo dalje raditi i rafinirati naše pravilo.

U navedenom primjeru imamo direktive **<program_name>** i **<match>**. S <program_name> možemo ograničiti djelovanje pravila na točno određeni program, u ovom slučaju saslauthd. <Match>, i njegov moćniji "partner" <regex> (koji omogućava uporabu regularnih izraza) se odnose isključivo na "log" dio, u ovom slučaju na dio retka "pam_unix(smtp:auth): authentication failure...".

Ovo je jako bitno zapamtiti kod pravljenja svojih pravila, da ne bi pokušavali "matchirati" od početka retka u logu (konretno, to je datum i vrijeme).

Ostalo je za objasniti još samo "**level**". OSSEC podržava petnaestak razina ozbiljnosti događaja ("severity"). Sve razine od 7 nadalje će generirati mail administratoru (ovo se može promijeniti u datoteci ossec.conf), dok će razina 0 poništiti bilo koju drugu razinu, ukoliko pravite "child" pravilo. To je upravo ono što želimo, ugasiti poruke za koje smo utvrdili da nam ne znače ništa i samo zatrpavaju sandučić elektroničke pošte.

Ukoliko želimo promijeniti razinu kod koje se šalju upozorenja mailom, u datoteci /var/ossec/etc/ossec.conf promijenite unos unutar oznaka **<email_alert_level>**:

u neku drugu, višu ili nižu, razinu. Ovo možda nije za preporučiti dok se bolje ne upoznate sa sustavom, pa jednostavno možete unutar vašeg pravila dodati unos

pa će vaše pravilo uvijek generirati upozorenje mailom, bez obzira na globalnu postavku razine

definirane unutar ossec.conf.

Kao primjer, navest ćemo rješenje za konkretan upit koji smo dobili od kolege sistem-inženjera kako da OSSEC šalje upozorenja ako se aktivira zaštita iptablesa, koju smo opisali u članku http://sistemac.carnet.hr/node/71.

Ovo je lako učiniti, sada kada znamo na koji način OSSEC raščlanjuje unose u logovima. Odgovarajući redak je:

Nakon raščlanjivanja, imamo:

time -> Jun 9 14:35:00
hostname -> posluzitelj
program_name -> kernel
log -> [1296546.520971] SSH_brute_force:IN=eth0...

Od ovih informacija, lako možemo napraviti pravilo:

Ovo pravilo upišite u svoju grupu pravila (između vlastitih <group></group> pravila), i restartajte OSSEC. Sada biste trebali početi dobijati obavijesti u trenutku kada se uključi zaštita zbog prekomjernih pokušaja spajanja na SSH daemon.

Naravno da ovakva pravila možete napisati za bilo koji servis, te ih možete imati neograničen broj. Nemojte ni pretjerivati, jer ćete početi ignorirati preveliku količinu ovakvih mailova sa sigurnosnim upozorenjima.

Ukoliko administrirate nekoliko poslužitelja, a za sistem-inženjera nije nemoguće da brine o nekoliko desetaka, pa i stotina poslužitelja, jasno je da s brojem poslužitelja klasičan način administracije neće biti optimalan. Postoje sustavi replikacije konfiguracijskih datoteka, kloniranja korisnika i slično, koji mogu pomoći u ovakvim situacijama. No, za manji broj poslužitelja bit će dovoljan dodatak za ssh koji paralelizira administrativne operacije. Poznat je kao "**pssh**", ili na Debianu i Ubuntuu "**parallel-ssh**" (paket se zove pssh, ali su naredbe preimenovane zbog konflikta s paketom putty).

Prije nego počnemo, ono što valja naglasiti je: ništa vam od ovoga što ćemo opisati neće raditi ukoliko niste ispravno podesili autenticiranje bez zaporke (kojeg smo opisali u članku "Bez zaporke uz pomoć ssh-agenta").

Paket se instalira kao i svi drugi, pomoću apt-get ili aptitude:

Ovime ste dobili nekoliko alata:

Parallel-ssh, naravno, izvršava istu naredbu na nekoliko udaljenih poslužitelja odjednom. **Parallel-slurp** može skinuti datoteke s udaljenih poslužitelja, dok **parallel-scp** radi obrnuto.

Naredba zanimljivog imena parallel-slurp "nemilosrdno - (**kill -9**)" ubija procese zadanog imena na svim poslužiteljima odjednom, dok **parallel-rsync** nudi mogućnosti alata rsync, ali naravno na više poslužitelja odjednom. Kako vidimo, ova dva zadnja alata se djelomično preklapaju sa ostalim alatima iz paketa, pa ih nećemo posebno obrađivati.

Ukoliko nešto "ne štima", susrest ćete se s ovakvom porukom:

Zašto je došlo do greške saznat ćete ukoliko upotrijebite prekidače "-o" i "-e". Prvi preusmjerava standardni izlaz (STDOUT), a drugi ispis grešaka (STDERR) u datoteku. Možda ćete nešto otkriti i u logovima udaljenog poslužitelja. Parallel-ssh sam po sebi nije baš informativan, čak i uz opciju "-v" (uključuje način rada *verbose*). Ipak, može vam pomoći opcija "-i" (*inline*) koja na ekranu ispisuje i poruke sa standardnog izlaza i standardne greške. Napomena, opcija "inline" **nije podržana u parallel-scp i parallel-slurp**.

U gornjem primjeru zadali smo da se na svakom od tri poslužitelja izvrši naredba "hostname --fqdn", čime smo naredili poslužitelju da se predstavi punim imenom i prezimenom. Može li se rabiti ime poslužitelja umjesto IP adrese, ako nam je tako zgodnije? Naravno...

Probajmo nešto prebaciti na te poslužitelje:

U primjeru smo prebacili datoteku **test.txt** u osnovni direktorij korisnika "korisnik".

Da ponovimo, u naredbama za prebacivanje datoteka (parallel-scp i parallel-slurp) nema opcije "-i (inline)" za ispis grešaka na ekran, pa u slučaju da vam prebacivanje datoteka ne radi, svakako uključite opcije "-o" i "-e".

Pokušajmo prebaciti te iste datoteke natrag. Ovaj put sintaksa je malo drugačija:

Kod skidanja datoteka potrebno je navesti u koji će se direktorij snimati datoteke, puna staza na udaljenom poslužitelju do datoteke, te naziv koji će datoteka dobiti u lokalnom direktoriju. Možda malo nelogično, ali "tako je to zamišljeno".

U praksi smo doživjeli da alati iz ovog paketa iz nepoznatog razloga ponekad odbiju raditi, a razlog je u jednom dijelu slučajeva bio na strani udaljenog poslužitelja. Naime, većina poslužitelja danas ima zaštitu od *brute-force* napada alatom fail2ban ili pomoću samih iptablesa, a te su se zaštite znale aktivirati i spriječiti prijenos datoteka ili izvršavanje naredbi. Imajte to na umu, uključite ispis STDOUT-a i STDERR-a i problema neće biti.

Ukoliko rabite paket pam_radius_auth za autentikaciju korisnika preko radiusa, možda ste u logovima ugledali sljedeću poruku (ili vam ju je proslijedio OSSEC):

Nadamo se da niste pomislili kako ovakve poruke treba ignorirati, ili još bolje, konfigurirati OSSEC da vam ih uopće ne šalje (upute su, naravno, tu *negdje* na Portalu). Ne, moramo problem riješiti.

Iz poruke se može zaključiti da se iz nekog razloga ne može saznati adresa za poslužitelj "**other-server**". Problem u DNS-u? Možda, ali koji je to poslužitelj "other-server"? Konfiguracijska datoteka za pam_radius_auth je **/etc/pam_radius_auth.conf**, pa pogledajmo prvo tamo. Ukoliko ste u editoru potražili niz "other-server", odmah ste mogli pronaći ovo:

Dakle, netko je (vi u brzini, neki vaš prethodnik ili zamjena) u konfiguraciji zaboravio zakomentirati **redak s primjerom** (iako se sjetio zakomentirati prvi). Sada je jasno otkuda "other-server" i zašto to neće nikada proraditi. Navedeni server nije konfiguriran niti će ikada biti pod tim imenom, te ga jednostavno treba zakomentirati:

Ovo je dovoljno da se problem riješi, i pouka kako treba pažljivo pregledati svaku učinjenu promjenu prije izlaska iz editora. Jedan apostrof ili zaboravljeni znak komentara može prouzročiti poteškoće ili prekide u radu servisa. U ovom slučaju radilo se "samo" o dodatnoj pauzi kod prijave na bilo koji servis. No, i ovo je dovoljno da se javljaju korisnici i tuže da im je "server spor" i slično.

Sigurno ste u ovo prazničnо vrijeme dobijali povećani broj spamova, lažnih pozitiva u spam mapama te lažiranih mailova na hrvatskom jeziku. Ovi potonji pokušavaju nagovoriti korisnike da im pošalju svoju zaporku ili drugu povjerljivu informaciju. Jednu takvu *phishing* poruku mnogi su dobili pred blagdane:

Iako je poruka očigleno loš prijevod sa stranog jezika, dovoljno je razumljiva da poneki korisnik u žurbi povjeruje u njenu vjerodostojnost i pošalje tražene podatke. Kada dobiju zaporku, spameri počinju rabiti vaš poslužitelj kao *relay* server, te ubrzo vaša domena dospije na crne liste poput onih koje sami koristite (bl.spamcop.net, dnsbl.njabl.org, zen.spamhaus.org i slične). Kad se to dogodi, vaši će se korisnici početi žaliti da više ne mogu slati poruke, jer ih odredišni serveri odbijaju.

Što učiniti? Jedino dugoročno rješenje je **preventivno** djelovanje i obuka korisnika. Naučite svoje korisnike da provjere takve poruke prije nego im povjeruju. Kada su u nedoumici, neka potraže savjet sistemskog inženjera. I naučite ih da nikad nikome ne otkrivaju svoju zaporku. Uostalom, to bi trebalo biti jasno navedeno u sigurnosnoj politici.

Dobro bi bilo da korisnicima održite barem jedan tečaj godišnje, koji bi odgovarao na njihove najčešće upite, te upoznavao nove zaposlenike i suradnike "kako stvari funkcioniraju".

A što učiniti *postmortem*, u situaciji kad je spam već došao u korisničke sandučiće? Najprije pošaljite

poruku svojim korisnicima i upozorite ih da je ta poruka lažna, da je obrišu i ne odgovaraju na nju (i nadajte se da nitko nije poslao zaporku). Da ovaj mail ne bi ponovo stigao na sustav, naučite SpamAssassin da se radi o spamu. "Trening" SpamAssassina je izvodi ovako: prvo izdvojite mail u datoteku, a potom pozovite naredbu sa-learn:

Vjerojatno to neće biti dovoljno, jer statističkom filteru je potrebno na stotine poruka (i *spam*a i *hama*) kako bi postao efikasan. Brže rješenje je blokiranje poruke po sadržaju. Upišite sljedeće u /etc/spamassassin/local.cf:

Ovime smo dodali ocjenu (SA *score*) 5.0 svakoj poruci koji sadrži pojam "MINIS" (od MINISTARSTVO FINANCIJA). Granica reza je obično na 6.31, no nikada ne dajemo ocjenu koja odmah prelazi granicu, jer time dajemo šansu SpamAssassinu da propusti mail ukoliko sadrži dovoljno drugih elemenata za prolaz. Primjerice, nije isto ako dobijete ovaj spam izravno ili vam ga proslijedi korisnik s upitom "što da napravim s ovim?". Želite dobiti upit, ali ne i spam, zato *score* treba biti nešto niži.

Mail je u HTML formatu, moramo ga tako i upisati jer SpamAssassin ne radi dekodiranje HTML-a. Radi jasnoće je stavljeno samo nekoliko slova, no možete staviti i više, dapače, preporučljivo je. Blokiranje po sadržaju nije jednostavno i traži određena predznanja. Treba pripaziti da ne omogućimo *false positive*, blokiranje dobrih mailova, upita vaših korisnika ili savjeta s lista o tome kako ukloniti problem. Stoga, oprezno s ovom mogućnošću, a najbolje je da ovakve *harcodirane* unose izbrišete nakon nekog vremena, kad poruke prestanu stizati. Spameri će već prijeći na neku drugu lažiranu obavijest.

A što učiniti ako su spameri počeli rabiti vaše računalo kao *mail relay*? O tome u drugom članku, u kojem ćemo opisati kako otkriti koji korisnici i preko kojih servisa šalju spamove, te kako popraviti nastale rupe u sustavu.

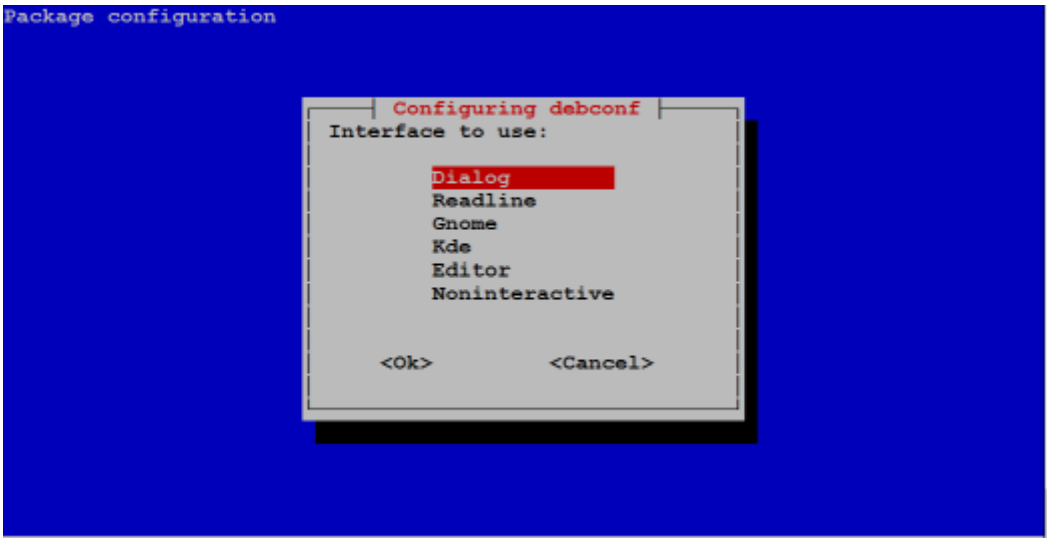
Nakon nadogradnje sustava na izdanje Squeeze, neki su kolege primjetili da im se pojavljuju poruke o greškama, odnosno upozorenja o neinicijaliziranim vrijednostima u modulu Priority.pm. Poruke se pojavljuju prilikom uporabe alata apt-get, a izgledaju ovako:

Ne morate previše brinuti, rješenje problema je vrlo lako.

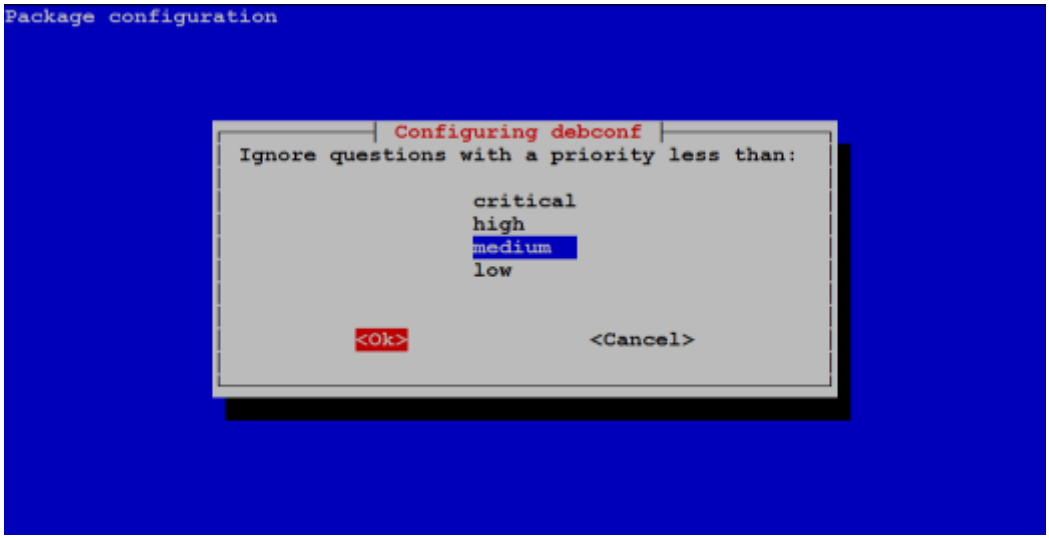
Problem je u jednoj zvjezdici viška, koja se pojavi unutar datoteke /var/cache/debconf/config.dat:

Problem je lako rješiti. Pokrente naredbu:

Na postavljena pitanja odgovorite najprije sa "Dialog":



i zatim s "Medium":



Možda bi bilo dobro da objasnimo druge opcije koje se mogu vidjeti u izborniku. Debconf je sustav za konfiguraciju paketa putem upita korisniku. Te upite debconf može postavljati na nekoliko načina.

Prvi je preko programa "Dialog", koji iscrtava svima poznate pregledne menije u boji. Za uporabu Dialoga nije potrebno grafičko sučelje (GUI).

Ukoliko je konfiguracija vašeg terminala oštećena pa se umjesto lijepih grafičkih elemenata pojavljuju nečitljive "črčke", možete odabrati opciju "Readline". Ova opcija rabi najosnovniji način rada u naredbenom retku, koji odgovara bilo kojem terminalu. Zato je dosta zgodan ukoliko često imate potrebu za radom preko mobitela ili preko nekog drugog ograničenog sučelja.

Druge dvije opcije, "Gnome" i "Kde", zahtijevaju grafičko sučelje, pa se njima ovdje nećemo ni baviti.

Sljedeća opcija je "Editor", koja će pokrenuti vaš standardni program za uređivanje tekstualnih datoteka. U njemu tada možete uređivati opcije na najkomforniji način, ali sumnjamo da će vam to biti često potrebno (ili poželjno).

Zadnja opcija je "Noninteractive". Ako je odaberete, nikakva se pitanja neće postavljati, konfiguracija će biti prepuštena paketu, što znači da servis najvjerojatnije neće proraditi kako treba ili neće uopće proraditi.

Drugi parametar koji možemo promijeniti je prioritet poruka, odnosno upita koje želimo vidjeti. Ponuđeni su prioriteti "**critical**", "**high**", "**medium**" i "**low**". Ovo može izazvati zabunu ako mislite da postavkom "low" dobivati manje upita. Zapravo je logika suprotna: s postavkom "low" dobit ćete **sva** pitanja, a s postavkom "critical" (najvišom) **samo** pitanja **kritična** za rad sustava (jedan paket može imati pitanja na više razina).

Preporučujemo postavku "medium", koja pruža optimalan skup upita.

Iako je većini godišnji odmor već prošao, program '**vacation**' će svakako pomoći i kasnije, ili najkasnije za prve sljedeće blagdane ili odmor. Program vacation ne skriva svoju svrhu iza čudnog imena, nego je sve odmah jasno: vacation umjesto vas odgovara na mailove dok ste vi odsutni s predefiniranom porukom. Pri tome pamti kome je sve poruku poslao, i u sljedećih 7 dana neće istoj osobi poslati poruku o odsutnosti.

Program se mora prvo inicijalizirati, što se može učiniti jednostavnim pozivanjem samog programa i odgovaranjem na pitanja:

Nakon ovoga, bit će kreirane tri datoteke u vašem $HOME direktoriju: **.forward**, **.vacation.msg** i **.vacation.db**.
Ukoliko već imate .forward datoteku, bit ćete upitani što s njom. Ukoliko odgovorite "Y", bit će napravljena kopija i nećete izgubiti sadržaj te datoteke.

Datoteka .vacation.msg sadržava poruku koja će se slati, i inicijalno sadržava tekst:

Datoteku treba prilagoditi po vašim željama. Varijabla $SUBJECT sadržava Subject polje originalne poruke, što možete, ali i ne morate rabiti.

Datoteka .vacation.db sadržava bazu e-mail adresa s kojih ste primili mail, sa svrhom da svaki pošiljatelj primi samo jednu poruku vašem odsustvu, odnosno jednu poruku tjedno.

Kad odgovorite sa 'Y', datoteka će biti preimenovana u **.forward~vacation~backup**, a novi sadržaj će biti:

Po sintaksi koja se rabi u .forward datotekama, može se vidjeti da će svaki mail stići i korisniku, ali i biti proslijeđen programu vacation. Da bi mogli primati mail i na aliase, modificirajte datoteku .forward ovako:

Od opcija (sve ih možete vidjeti sa "man vacation"), napomenut ćemo najzanimljivije i najkorisnije.

Opcija "**-r**" mijenja broj dana nakon kojeg će se opet slati poruka odsutnosti na istu adresu. Tako će "**vacation -r 15**" promijeniti broj dana sa osnovnih 7 na 15, pa ukoliko u tom razdoblju pošiljatelj pošalje više mailova, samo će jednom dobiti poruku od vacationa.

Zanimljiva je i opcija "**-l**", koja će ispisati sve unose u bazi, dakle sve adrese s kojih ste primili e-mail. Zgodno je za znati, a i korisno ukoliko mislite rabiti sljedeću opciju.

Opcija "**-x**" je najvažnija, jer pomoću nje možete unaprijed upisati adrese na koje ne želite slati obavijest. Opcija prima adrese na standardnom ulazu, pa je najjednostavnije da sve adrese upišete u datoteku i napravite redirekciju:

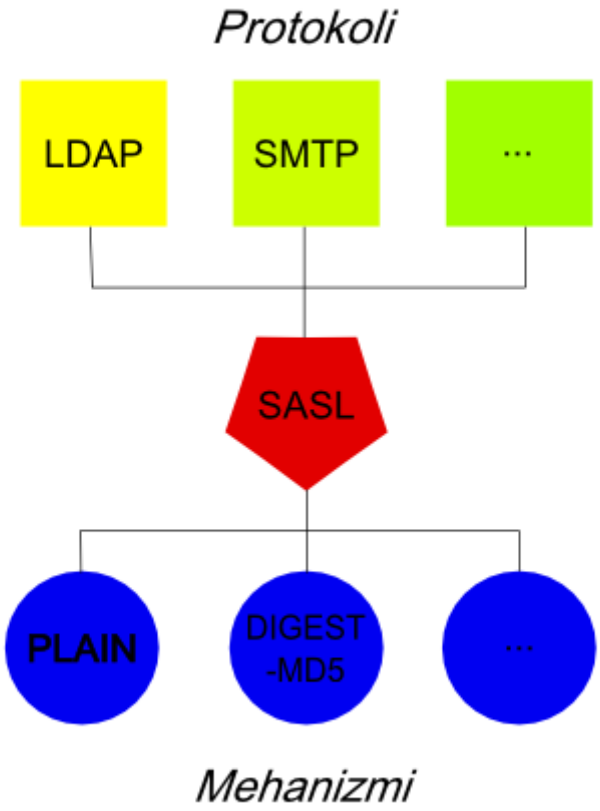
Cijele domene možete ubaciti preko sintakse: @domena

Vacation pazi da ne odgovara na poruke od sustava ili na liste, ali uvijek može promaći koja poruka, pa s vacationom oprezno (istestirajte na nekom drugom korisničkom računu), da ne biste dobili pokoji *hate-mail*.

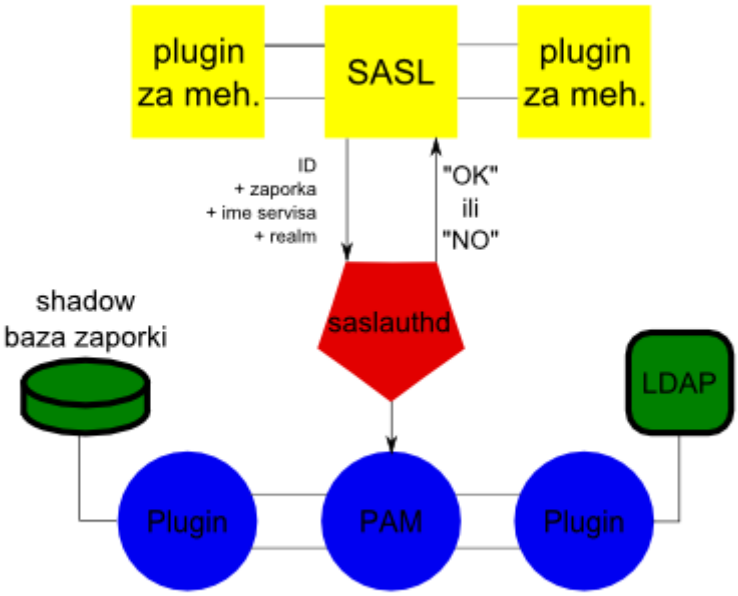
Na kraju, ne zaboravite vratiti .forward na početno stanje nakon povratka, dok ostale datoteke možete sačuvati za sljedeći put.

SASL (Simple Authentication and Security Layer) rabite već godinama, a da to možda i ne znate. Radi se o načinu sigurne autentikacije korisnika prema svim mrežnim servisima, no oni moraju imati ugrađenu podršku za SASL. Na ovaj način se izbjegava da svaki servis posebno radi autentikaciju, što može rezultirati sigurnosnim propustima zbog grešaka i propusta u kodu.

SASL, slično PAM-u, može raditi autentikaciju na mnogo načina. Tako korisnike možete autenticirati direktno iz /etc/passwd, odnosno /etc/shadow datoteke (getpwent i shadow metode), Kerberosa, IMAP-a, PAM-a ili LDAP-a. Također, podržava i svoju vlastitu bazu (nalazi se u /etc/sasldb), no uz sve ove mogućnosti, ova zadnja se čini izlišnom.



SASL čini program, odnosno daemon **saslauthd**, i moduli koji omogućavaju komunikaciju daemona i mrežnog servisa. Moduli definiraju različite *chalenge-response* mehanizme, kao što su EXTERNAL, ANONYMOUS, PLAIN, OTP (*one time password*), NTLM, DIGEST-MD5 i druge. Sve ovo zapravo i nije toliko zanimljivo za uobičajenu uporabu, dovoljno je znati da morate imati instaliran paket **libsasl2-modules** kako bi podržavali sve ove mehanizme.



Saslauthd nema standardnu konfiguracijsku datoteku, nego se podešavanje obavlja u datoteci /etc/default/saslauthd. Postoji nekoliko varijabli koje nam mogu biti zanimljive. No, sa samo dvije možemo puno postići: START i MECHANISMS.

Varijabla START određuje hoće li se saslauthd uopće pokrenuti prilikom pokretanja sustava. Jasno je da bi vrijednost trebala biti:

Varijabla MECHANISMS određuje koje ćemo prije spomenute mehanizme rabiti. Uobičajeno je staviti "pam", jer na taj način rabimo sve načine autentikacije koje ste možda prethodno postavili i prilagodili svojim potrebama:

Ispravno konfiguriranu SASL autentikaciju možete provjerite s alatom **testsaslauthd**:

Budite oprezni kod izvršavanje ove naredbe, jer će se zaporka vidjeti u popisu procesa (pomoću naredbi "ps" i "top") za vrijeme izvršavanje naredbe testsaslauthd.

Postfix koji dolazi u CARNet paketu eksplicitno zahtijeva da imate instaliran SASL.

SSL (Secure Sockets Layer) vam je možda najpoznatiji od prije kao SSLeay, sustav koji je omogućavao sigurni, odnosno enkriptirani promet između klijenta i poslužitelja. Najčešće ga se instaliralo kao dodatak u Apache u prvim on-line web trgovinama. Prvu inačicu SSL-a je napisao Netscape, i ona nije ugledala svjetlo dana, nego je vrlo brzo zamijenjena inačicom 2.0 u veljači 1995. godine. SSL (i TLS) se danas bazira na OpenSSL biblioteci otvorenog koda.

SSL je u svojim inačicama 2.0 i 3.0 zaživio i u mnogim drugim primjenama, koje do tada nisu imale podršku za siguran prijenos podataka preko TCP/IP mreže (ili su imale neka vlastita rješenja).

Par godina kasnije, protokol je dodatno standardiziran i opisan u RFC-u 2246, te je preimenovan u TLS (Transport Layer Security). Može se reći da je TLS 1.0 nadgradnja SSL-a 3.0, te se ponegdje može pronaći da je TLS 1.0 zapravo SSL 3.1, no uglavnom ćete sresti da je TLS == SSL3. No, ova dva protokola nisu u potpunosti identična, pa to treba imati na umu. Iako se protokoli pomalo razlikuju, aplikacije će znati prijeći sa TLS-a na SSL3

Danas nije poželjno rabiti bilo koju inačicu SSL-a nižu od 3. Aktualna je inačica TLS/a 1.2, izašla 2008. godine.

Dolazimo do pojma STARTTLS. STARTTLS nije nikakav poseban protokol, nego se upotrebljava kada želimo neenkriptirani link pretvoriti u enkriptirani. Ovo se najčešće rabi kod mail klijenata i poslužitelja, primjerice u Postfixu. Ovo je drugačije ponašenje nego primjerice kod HTTPS-a, koji odmah dogovora enkriptirani kanal (dakle, radi na drugom OSI sloju).

Konkretno, kako ćemo znati da određeni SMTP poslužitelj podržava TLS? Spojit ćemo se na port 25 i vidjeti koje opcije poslužitelj nudi:

Iz ispisa možemo saznati mnoge zanimljive stvari (npr. poslužitelj podržava poruke do 20 MiB i SMTP AUTH autentikaciju). No, nas zanima samo redak STARTTLS ukucavanjem ključne riječi "starttls" pokrećemo enkripciju, što možemo vidjeti po poruci:

Sada će prijenos maila biti enkriptiran i siguran, naravno, sve dok ne dođe u korisnikov mailbox i time izgubimo uvid što se dalje s porukom događa.

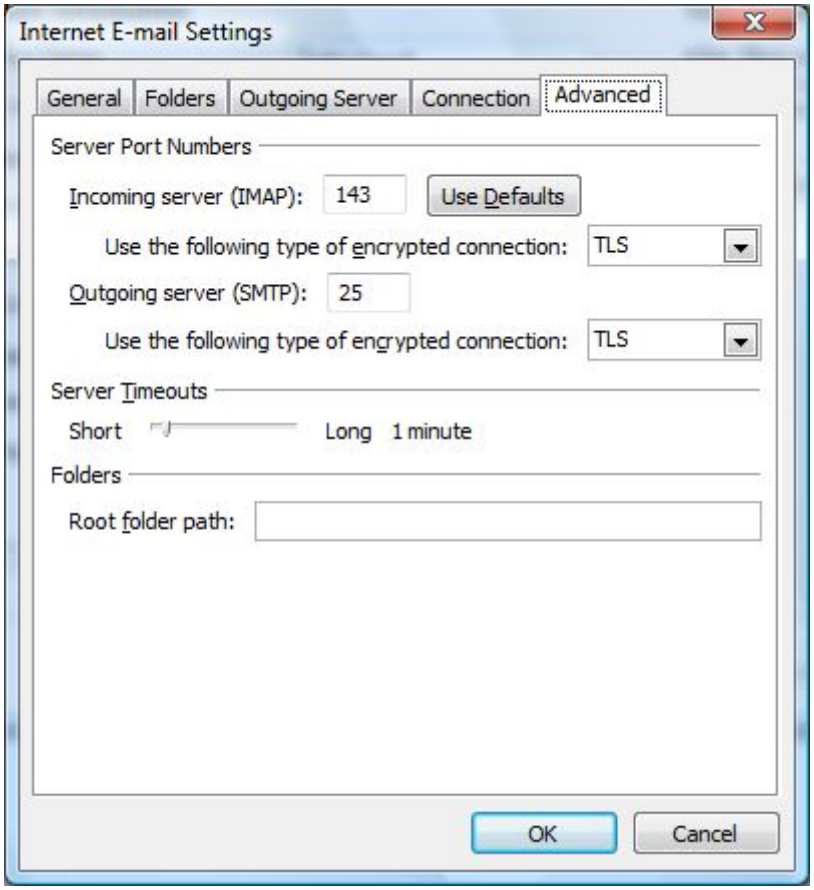
Do ovdje je vjerujemo bilo sve jasno. No, kako svaki protokol koji rabi TLS dobiva novi port, te da se u nekim slučajevima promet može odvijati preko "starog" porta (npr. 25). Na kraju, pojedini softveri mogu imati malo drugačiju terminologiju, što na kraju može dovesti do zbunjenosti kako korisnika, tako i sistemaca.

Za početak, navest ćemo uobičajene portove i njihove "sigurne" inačice (svi portovi su TCP):

Prvo stvar, svaki ne-enkriptirani protokol ima svoj port, a svaki enkriptirani drugačiji, preko kojeg ide identičan promet, samo je razlika u tome što je enkriptiran.

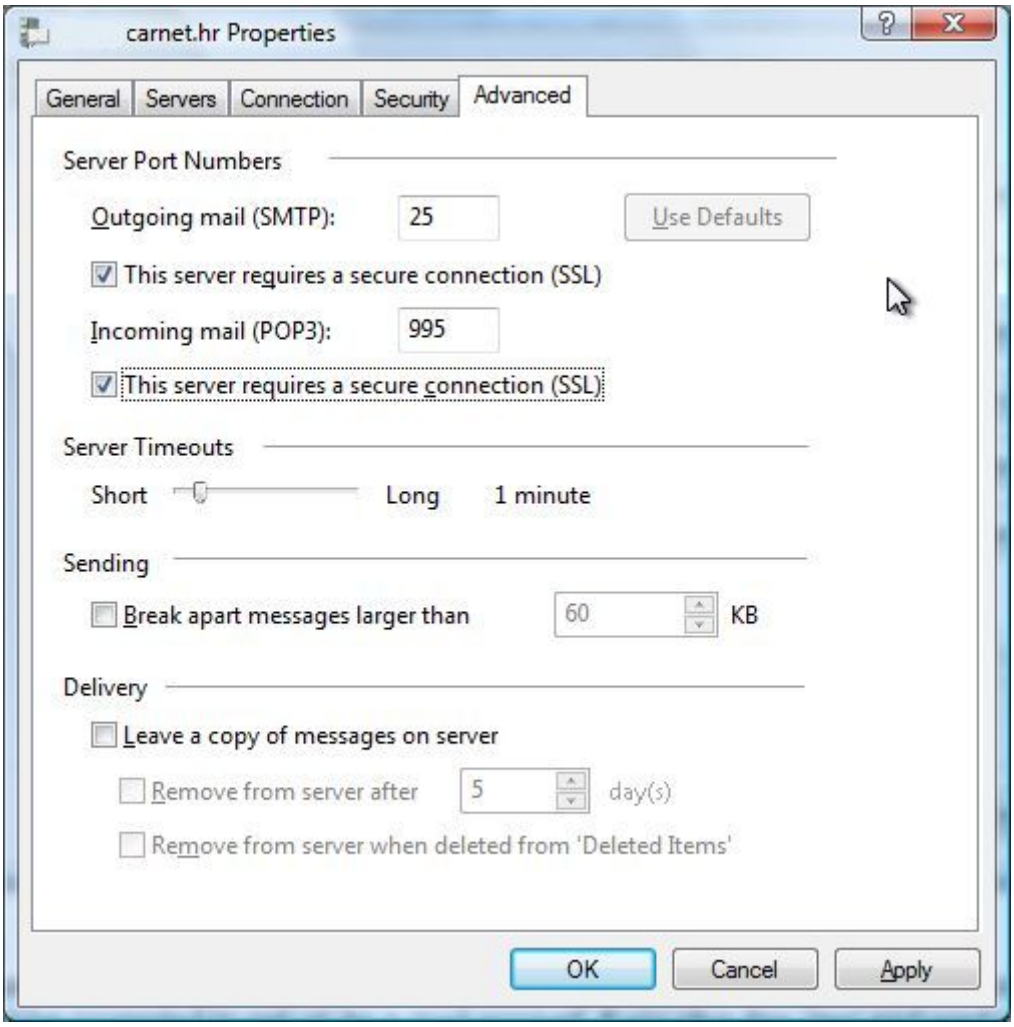
Drugo, enkripcija može ići i preko starog porta, ako to podržava poslužitelj, ali i klijent (naveli smo primjer starttls naredbe koja radi upravo to).

Kako onda uključiti ove enkripcijske protokole u najpopularnijim programima?



Za **Outlook**, odite na Tools -> Account Settings -> <dvoklik na profil> -> More Settings -> Advanced -> <odaberite SSL, TLS ili AUTO> za "Incoming" i "Outgoing" poslužitelje.

Ukoliko odaberete SSL, port za dolazni IMAP poslužitelj će se promijeniti na 993 (imaps), dok TLS podržava i enkripciju preko starog porta (slično je i za POP3 poslužitelj). Ovo je važno ukoliko ne želite u vatrozidu otvarati nove portove. Također, Outlook odmah nudi da testirate navedene postavke, pa to učinite. U logovima na poslužitelju ćete vidjeti zašto ne radi, ako se pojavi problem (ne zaboravite prilagoditi vatrozid između klijenta i poslužitelja!).

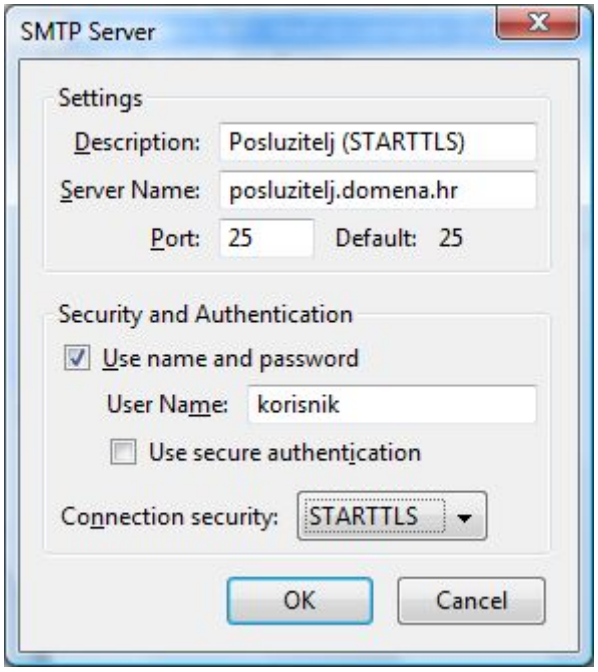


Za **Windows Mail** (gornji primjer je za POP3 protokol), odite na Tools -> Accounts -> <dvoklik na profil> -> Advanced -> Uključite "This server requires secure connection (SSL)" za "Outgoing server (SMTP)" -> Uključite "This server requires secure connection (SSL)" i za "Incoming mail (SSL)" (port će se promijeniti na 995, pop3s).

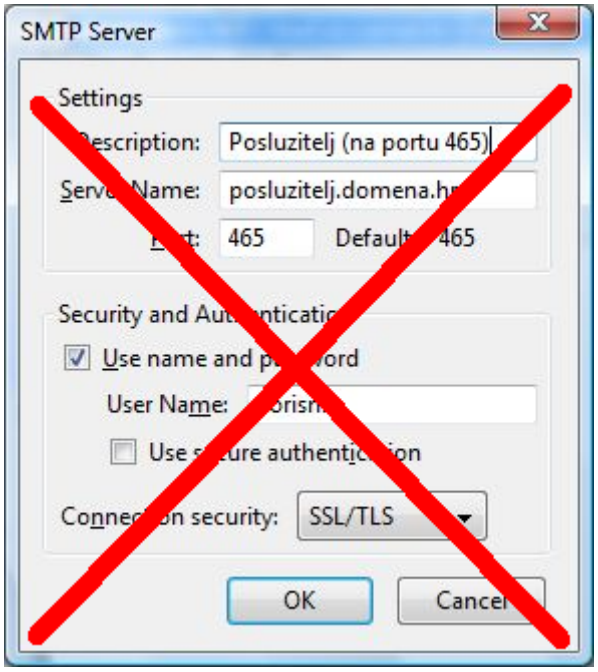
Za **Mozilla Thunderbird** dolazni poslužitelj, odite na: Tools -> Account Settings -> Server Settings željenog poslužitelja -> Security Settings -> Odaberite STARTTLS ili SSL/TLS

Ukoliko odaberete SSL/TLS, port će se promijeniti na 993 ili 995, u ovisnosti je li POP3 ili IMAP tipa.

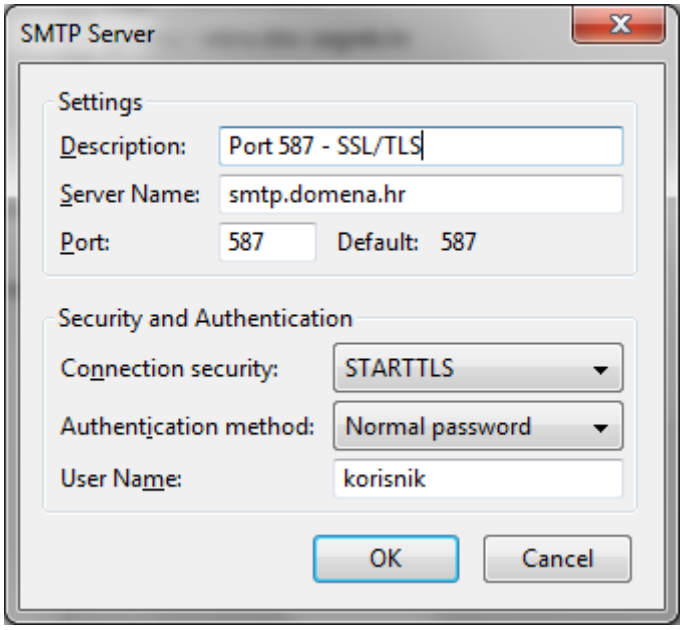
Za odlazni poslužitelj, u opcijama s lijeve strane odaberite "Outgoing Server" -> <dvoklik na poslužitelj> -> odaberite STARTTLS ili SSL/TLS.



Ukoliko odaberete STARTTLS, port će ostati 25, a odaberete li SSL/TLS, port će se promijeniti na 465.



No, **port 465 se više ne rabi u ove svrhe**, štoviše, već je prenamijenjen za druge svrhe. **Port koji bi trebali rabiti je 587**:



Nadzor rada informatičkih sustava oduvijek je bio problem, uglavnom zbog različitih potreba tvrtke u kojoj se sustav treba nadzirati.

Danas se najviše koriste gotovi proizvodi, kompleksni i prilagodljivi sustavi nadzora, poput poznatih programa iz svijeta open sourcea, *mona* i *nagiosa*.

No, ako se dosad nise susreli s *monom,* ili jednostavno ne želite instalirati i podešavati cijeli sustav, možete posegnuti za dobrim starim skriptama koje će brzo riješiti problem, a ako ste s njima zadovoljni, mogu se koristiti i trajno.

Ova skripta napisana u perlu izuzetno je jednostavna i prilagodljiva. Objašnjavamo kako je konfigurirati i rabiti na unixu.

Skripta obično radi iz crona i pokreće se svakih nekoliko minuta kako bi pružila što ažurniju sliku stanja mrežnih servisa. Prima dva parametra, --mail i --forcemail. Prvi parametar šalje mail svakih $PERIOD minuta u slučaju da neki servis ne radi i služi za uporabu iz crona. Drugi šalje mail svaki put kad se skripta pokrene i služi isključivo za testiranje i podešavanje. Za uporabu iz naredbenog retka nije potrebno navoditi nikakve opcije.

Upisano u cronu može biti otprilike poput ovog:

Akone radite pod linuxom, možete pokušati ovako:

U prvom slučaju skripta se pokreće svakih 5 minuta, a u drugom svakih 10 minuta. Skripta ne mora raditi pod root ovlastima, čak se to ne preporuča jer skripta ne pruža zaštitu od pokušaja zloupotreba.

Skripta se konfigurira samo s nekoliko parmetara.

$MAILTO = **"nickname\@provider.hr"**;

$MAILTO parametar služi za definiranje email adrese na koju će se obavijesti slati. Po zamisli, to je alias koji preusmjerava mail na SMS gateway. Oba mobilna operatora podržavaju tu mogućnost. Obratite pažnju na obvezni backslash u email adresi.

$HOST = "neki.server.hr";

$HOST definira poslužitelj gdje se servisi nalaze.

$PERIOD = 120;

$PERIOD je naravno razdoblje, u minutama, u kojemu skripta ponavlja slanje obavijesti za pojedini

servis.

$services = '21 22 25 53 80 110 111 113 139 143 389 587 3306';

$services = 'ftp ssh smtp domain http pop3 auth 139 imap2 ldap submission';

Varijabla $services sadržava sve servise koje želite nadzirati. Oni se mogu napisati numerički ili simbolički.

$nc = "/bin/nc";

$nc = "/usr/local/bin/nc";

Za pregled otvorenih portova rabimo popularni alat netcat (nc), pa ovdje možemo podesiti stazu do njega ako se iz sigurnosnih razloga ne želite pouzdati u PATH. No, skripta se ne vrti pod root korisničkom oznakom, pa velika zaštita skripte ili environmenta nije nužna.

$timestamp = "$ENV{HOME}/.check_services";

$timestamp određuje gdje će se snimati podaci o trenutačnom stanju servisa.

Default je $HOME korisnika pod kojim se skripta pokreće. Možete ovo i promijeniti, samo treba pripaziti da nitko drugi ne može pisati u tu datoteke zbog eventualnog race conditiona.

$mail = "/usr/bin/mailx";

I na kraju, varijabla $mail označava mail klijenta kojeg želite rabiti za slanje maila, i ako ga želite promijeniti, pripazite na ispravnu sintaksu u programskoj petlji na mjestu u skripti gdje se šalje mail.

Ugodna uporaba skripte check_services!

Squirrelmail je jedan od najpopularnijih webmail sustava, vjerojatno ga imate instaliranog zahvaljujući tome što dolazi u obliku CARNetova paketa. Postoji nekoliko desetaka plugina za Squirrelmail. Upoznat ćemo vas sa "Squirrel Logger" pluginom, jer on upotpunjava niz od nekoliko članaka koje smo planirali oko problema zloporabe mail poslužitelja.

Dakle, ponekad želimo znati tko to točno i kada šalje mailove iz Squirrelmaila, jer on te informacije sam "nerado" prikazuje. Razlog zašto bi željeli znati tko šalje mailove preko Squirrelmaila je najčešće provala u sustav, odnosno činjenica da je netko otkrio zaporku korisnika i počeo zlorabiti vaš sustav za spamiranje. Drugi razlozi mogu uključivati discipliniranje i praćenje korisnika koji ne poštuju dogovore o načinima uporabe sustava i slično.

Squirrelmail Logger plugin se instalira kao i svaki drugi plugin za Squirrelmail. Potrebno je skinuti arhivu na adresi:

http://www.squirrelmail.org/plugin_view.php?id=52

Arhivu squirrel_logger-2.3.1-1.2.7.tar.gz raspakirajte u direktoriju /usr/share/squirrelmail/plugins, te iskopirajte demo konfiguraciju:

Sa 8 odaberite "Plugins", te potom plugin "squirrel_logger". Snimite sa "s" i izađite sa "q".

Za početak, dovoljno je da odkomentirate sljedeće unose:

Prvo polje ($sl_log_events) određuje koje ćete događaje logirati, a drugo polje ($sl_logs) kako će se ti događaji bilježiti (syslog, datoteka, sql baza) i što će se u njih upisivati.

Gornje promjene će biti dovoljne da u logovima (/var/log/syslog, /var/log/mail.log) dobijete unose poput ovih:

To nije sve, jer je moguće logirati i druge stvari. Najzanimljivija je mogućnost logiranja mailova ako broj primatelja prelazi određeni (konfigurabilni) iznos. U terminologiji ovog plugina, to se zove "*mass mailing*":

'MASS_MAILING'   => "Possible outgoing spam: by %2 (%3) at %4 on %6: %7",

Kao što možete pretpostaviti, sve je dosta konfigurabilno, te možete promijeniti što će se logirati pomoću parametara:

%1 - ime događaja
%2 - ime korisnika
%3 - naziv domene
%4 - adresa udaljenog računala
%5 - timestamp (vrijeme događaja)
%6 - datum
%7 - neki komentar koji sami definirate

No, to nije sve (znamo, znamo, zvučimo kao u TV reklami). Pomoću ovog plugina možete, umjesto zapisivanja u logove, poslati mail na neku e-mail adresu. Kako ovo može izazvati dosta maila, predviđeno je da se šalju samo kritične obavijesti:

Adresa na koju se šalje pošta je u standardnoj vrijednosti naslovljena na postmastera, što zapravo ne treba mijenjati, jer bi alias postmaster trebao biti preusmjeren na root korisnika, a rootov mail bi trebali dobijati vi. Naravno da možete promijeniti ovu vrijednost ukoliko želite mailove slati na, primjerice, GMail. Obavijesti izgledaju ovako:

Possible outgoing spam: by korisnik (server) at 161.53.XX.YYY on 12/29/2011 15:45:05: Total 27 recipients

VAŽNO: za slanje pošte morat ćete instalirati "Compatibility Plugin", inačice barem 2.0.11 (aktualna je 2.0.16)

Ostale postavke su manje-više same po sebi razumljive. Možete podesiti što će logovi i izvješća sadržavati (koja zaglavlja), na koga će glasiti mail obavještenja, koji će se mail poslužitelj rabiti i slično.

Osim preko *syslog* mehanizma, moguće je zapisivati i direktno u SQL bazu. Za ovu se brinu polja

i

A ostalo je na vama...

Unatoč pažljivim kontrolama, u desecima tisuća Debian paketa ponekad se dogode greške, manje ili veće. Zbog velikog broja održavatelja (*maintainera*), različitih nepredviđenih okolnosti, te napuštenih (*orphaned*) paketa, moguće je da paket, u manjoj ili većoj mjeri, ne radi kako bi trebao, odnosno pokazuju se problemi u radu s njim ili nekim vezanim paketom. Pozabavit ćemo se problemom "visećih simboličkih veza" (*Dangling symlinks*).

Iako potpuno bezopasna, ova pojava se očituje tako da svaki put kada se iz crona osvježava baza man stranica, na mail dobijete poruku sličnu ovoj:

Iako iz same poruke to nije vidljivo, radi se o grešci prilikom registriranja "alternativnih" naredbi. Ne ulazeći ovom prilikom u detalje, spomenut ćemo samo da se radi o sustavu koji omogućuje da na sustavu imate više različitih inačica istog programa, primjerice naredba "editor" može pozivati "vi", "emacs", "joe" ili nešto sasvim drugo.

Problem je, kako to sam naziv i upućuje, u simboličkoj vezi (linku) koja ne pokazuje nigdje. Kako to znači da nam nije od nikakve koristi, slobodno možemo te veze obrisati. Za to postoji nekoliko načina:

Isto to možete napraviti i ručno:

Na kraju, kao alat opće namjene, možete instalirati i rabiti program *symlinks*:

Uporaba je jednostavna, recimo da samo želite saznati ima li visećih veza:

Sve ovo možete popraviti sa naredbom:

Kako simboličke veze ne zauzimaju puno prostora, nije nužno brisati ih sve sa sustava. Za rješenje

ovog problema bit će dovoljno skoncentrirati se samo na direktorij /usr/share/man i pokrenuti naredbu "symlinks -d -r /usr/share/man".

Kako bi se uvjerili da je sve u redu, ručno pokrenite skriptu koja se pokreće iz crona (i koja zapravo prijavljuje poruke o nedostajućim man stranicama):

Ukoliko je sve u redu, ne bi se trebalo ništa ispisati, a time ni generirati svakodnevni mail na roota.

Ova zaista čudna greška javlja se u /var/log/auth.log prilikom pokušaja startanja SSH daemona. Sasvim je moguće da će neki drugi servis otkazati poslušnost, uz neku sličnu poruku, primjerice apache (koji će čak eksplicitno spomenuti /dev/null).

Uzrok ovoj poruci je promjena tipa datoteke /dev/null, koja je vrlo bitna za rad sustava. Datoteke /dev/null i /dev/zero su posebne (character-based, nebufferirane) datoteke, a još se zovu i "data sink".

Čitanje iz /dev/null uvijek daje EOF (End of File), dok čitanje iz /dev/zero uvijek daje ASCII NULL karakter (\0). Pisanje je u njih, naravno, moguće, ali sam naziv "data sink" sugerira da se podaci tamo mogu jedino izgubiti (i ovo ima svoju primjenu).

Ako napravimo "ls -l /dev/null /dev/zero", dobit ćemo:

Ukoliko se stanje na vašem sustavu razlikuje, potrebno je napraviti slijedeće:

Nakon toga imate ispravno podešene te dvije datoteke, te će servisi proraditi nakon restarta (bilo problematičnih servisa, bilo cijelog sustava). Više o tim datotekama i naredbi mknod piše, a gdje drugdje, nego u manualu: "man null" i "man mknod".

Na pitanje zašto su se tako ključne datoteke "same od sebe" promijenile teško je dati odgovor, ali najvjerojatnije je riječ o lošoj instalacijskoj proceduri nekog od paketa, ili nestanku struje ili pak nekoj specifičnoj situaciji kod vas.

Unix time, odnosno *epoch (Unix time)* je, jednostavno rečeno, broj sekundi proteklih od ponoći 1. siječnja 1970. godine po UTC-u (Universal Time Code). Ovaj se broj rabi interno na svakom Unix/Linux sustavu u razne svrhe, ali je prilično skriven od pogleda. Češće će ga susresti programeri, nego sistem administratori, ali je korisno znati što predstavlja ovaj broj i kako ga rabiti.

S Linuxom dolazi GNU naredba date, koja podržava *epoch* format vremena. Kako ćemo saznati koliko je sekundi proteklo od ponoći 1. siječnja 1970? Vrlo jednostavno, samo treba upisati:

Kao što se može vidjeti, radi se o 10-znamenkastom broju, iako sam broj znamenki nije fiksan. Broj je s 9-znamenkastog na 10-znamenkasti broj prešao 9. rujna 2001. godine. Kako je interna reprezentacija 32-bitna (*signed*), a maksimalni pozitivni broj koji se može zabilježiti 32-bitno je 2147483647, proizlazi da će krajnji datum koju ovaj broj može prikazati biti 19. siječnja 2038. godine. Poslije toga, slično problemu Y2K, brojač se vraća na nulu i vrijeme 1. siječnja 1970. Više o tom problemu možete pogledati na Wikipedijinim stranicama: http://en.wikipedia.org/wiki/Year_2038_problem.

Iako se *epoch* ne susreće često, treba imati na umu da je interno itekako u uporabi, te ga rabe neki servisi za logiranje. Može se naći i u logovima ADSL i kabelskih "modema", dakle korisno je znati o čemu se radi.

Relativno je lako prikazati trenutno vrijeme u *epoch* vremenu, ali konverzija u ljudima razumljivo vrijeme nije toliko lagana kako bi trebala biti. Prikazat ćemo (zahvaljujući Unix i Linux guruima na Usenet grupi hr.comp.os.linux) kako konvertirati Unix vrijeme u ljudima razumljiviji oblik:

Na kraju, postoji čitav niz web stranica koje mogu ovu konverziju izvesti bez previše kompliciranja, a jedna od njih je http://www.esqsoft.com/javascript_examples/date-to-epoch.htm.

Koji ćete pristup rabiti, ostaje jedino vama na volju. Više informacija o *epochu*, kao i uvijek, možete saznati na Wikipedijinim stranicama: http://en.wikipedia.org/wiki/Unix_time

Prije nekoliko dana je jedan upit kolege razotkrio stari problem koji se često javljao prilikom razmjene datoteka s Unix/Linux na DOS/Windows računala, ali i obrnuto. Originalni upit glasi otprilike ovako: "Imam skriptu koja odbija raditi, a ne vidim zašto. Kad izvršavam te iste naredbe iz skripte jednu po jednu, sve radi."

U nastavaku je bio priložen ispis greške, a između ostalog je bila navedeno:

Upravo je znak "**\r**" ukazao na problem. Ovaj znak u programskom jeziku C obično označava karakter CR (carriage return, ASCII kod 13), koji određuje da se kursor u terminalu (ili glava pisača) vrati na početak retka.

Kao što znate, u tekstualnim datotekama se za kraj retka postavlja znak newline. Problem je u tome što se na različitim operativnim sustavima ovaj znak drugačije interpretira.

Na Unixima, pa time i Linuxu, u tu svrhu se rabi LF (linefeed, ASCII kod 10). Na DOS-u, odnosno Windowsima, rabe se dva znaka: CR+LF.

Ukoliko učitate datoteku na "pogrešnom" OS-u, dobite ćete cijelu datoteku u jednom retku (ako učitate Unix format na DOS-u), ili čudne artefakte na krajevima redaka (ako datoteku u DOS formatu učitate pod Unixima).

Dakle, na Unixima znak CR postaje višak na kraju svakog retka. Što je najzanimljivije, taj karakter je nevidljiv ukoliko datoteku učitamo u neki noviji editor teksta (vim, joe...), jer oni automatski prepoznaju o kojem se formatu radi i u tom formatu mogu transparentno snimiti datoteku.

Naravno, postoje načini kako bi vidjeli te "nevidljive" znakove.

Možda je najjednostavnije vidjeti tip datoteke pomoću naredbe **file**:

Jasno je naznačeno da je u pitanju datoteka s CRLF tipom završetaka redaka.

Možete i pregledati datoteku sa naredbom cat:

Ukoliko se bude pojavio niz znakova "**^M**" (cat-ova interpretacija CR karaktera), to znači da je datoteka u DOS formatu. Naravno, ne mora biti da je cijela datoteka u DOS formatu, nego samo jedan znak ili paragraf, što se može dogoditi ukoliko ste rabili editore sa obje platforme na toj istoj datoteci. Kolega je u ovom slučaju skriptu napisao na Windowsima, i prebacio na Linux, pa je cijela datoteka bila u DOS formatu.

Možemo postaviti pitanje možemo li konvertirati datoteku u pravilan format na neki jednostavan način? Naravno, i kao i uvijek na Linuxu, na više načina.

Vjerojatno najpopularniji je način uporabom specijaliziranih programa **unix2dos** i **dos2unix**. Na Debianu, te se naredbe nalaze u paketu **tofrodos**, a paket donosi i naredbe **todos** i **fromdos**, za slučaj da vam je tako lakše zapamtiti. Uporaba je jednostavna pa naredbe nećemo detaljno opisivati:

Ukoliko ne želite instalirati nikakve nove pakete, možete se poslužiti i jednostavnom naredbom **tr**, koja dolazi gotovo sa svim Linuxima (nalazi se u **coreutils** paketu):

Više informacija o ovome možete naći na wikipedijinim stranicama, uključujući i sve potankosti koje smo pokušali "sakriti" od vas u članku:

http://en.wikipedia.org/wiki/Newline

Za kraj, jedan mali kviz: znate li zašto se kolegi iz primjera ispisalo između ostalog i ovo:

Molimo, ukoliko znate odgovor, upišite ga u komentar članka.

Prije nekog vremena opisali smo kako putem alata grepmail pronaći mail prema određenim kriterijima u vašim mailbox datotekama. Iako se samom alatu nema što prigovoriti, ovaj način pretraživanja kod velikih mailbox datoteka može biti spor, te ne može rekurzivno pretraživati mailbox datoteke u drugim direktorijima. Sve ovo rješava **mairix**, alat koje će sve vaše mailbox foldere rekurzivno indeksirati, te će pretraživanje višegigabajtne arhive biti gotovo u samo nekoliko

sekundi.

Mairix može indeksirati mbox, Maildir i MH formate zapisa, tako da je pokrivena i situacija na tipičnom CARNet poslužitelju. Instalacija je jednostavna:

Sljedeći korak je, kao i uvijek, malo konfiguracije. U paketu dolazi datoteka /usr/share/doc/mairix/dotmairixrc.eg iz koje možete vidjeti kako bi trebala izgledati tipična konfiguracijska datoteka. No, za početak možete upotrijebiti ovu konfiguraciju:

Objasnit ćemo svaku stavku. Kao što se može pretpostaviti iz samog njenog naziva, opcija **base** određuje gdje se nalaze vaše mailbox datoteke. Ovo uključuje i sve eventualne direktorije ispod toga. Opcija **database** određuje lokaciju indeksne datoteke, i možete je ostaviti kako je predloženo.

Opcija **mfolder** određuje gdje će se snimiti mailovi koji odgovaraju kriteriju pretraživanja. Ukoliko je navedeno samo ime datoteke, njena staza se dodaje stazi zadanoj u opciji base, u ovom primjeru to će biti $HOME/mail/Search. Ukoliko navedemo apsolutnu stazu "/tmp/Search", rezultati pretraživanja će se tamo i snimati, bez obzira na vrijednost u opciji base (iako, ne preporučujemo snimanje u /tmp!).

Mformat određuje o kojem se formatu zapisa radi, a u našem slučaju je to klasični mbox. Ukoliko rabite Maildir, tako i upišite.

Najvažnija opcija je **mbox**. Ukoliko navedete samo tri točke "...", to označava da želite rekurzivno indeksiranje svega u $HOME/mail. Ukoliko želite indeksiranje samo određenih datoteka u $HOME/mail, primjerice samo mailboxova koji pocinju sa 2008-*:

U slučaju da u $HOME/mail imate mailboxove raspodjeljene po godina, 2006, 2007, 2008, i želite ih indeksirati, a ostalo ne, stavite:

Dakle, trostruka točka određuju rekurziju, odnosno da ukoliko neki direktorij počinje s 200, i njega će se indeksirati. Ukoliko želite indeksirati samo određene foldere, navedite ih, odijeljene dvotočkama:

Za ostale mogućnosti pogledajte u dokumentaciju, mada je indeksiranje svih mboxova u potpunosti realna i korisna opcija. Vrijedi napomenuti da mairix ne radi locking mboxova, što znači da ukoliko želite indeksirati INBOX, možda neće biti indeksirane sve poruke. No, indeksiranje INBOX-a i nije tako korisna opcija.

Ukoliko postoje samo neki mailboxovi koje ne želite indeksirati, postoji i opcija **omit**. Njena uporaba je malo drugačija, primjerice:

će indeksirati sve, osim mboxova koji počinju sa spam (primjerice spam, spam2008, spam2009...). Ukoliko je spam direktorij, pomoći će:

što će indeksirati sve, osim mailboxova i direktorija koji počinju sa spam, primjerice i spam2009 i spam/bilosto.

Nakon konfiguracije, slijedi indeksiranje. Jednostavno pokrenite mairix:

Ova će operacija potrajati neko vrijeme, i možda će ispisati pokoju grešku, no nemate neki izbor osim zanemariti ih. Kako bi indeks bio aktualan, ovu operaciju stavite u cron, npr:

Najzad, uporaba. Ona je različita nego ona kod grepmaila, ali nije ništa teža za zapamtiti. Primjerice,

će generirati mailbox kojeg ste definirali u mfolder opciji, a koji će sadržavati mailove koji u zaglavlju ili tijelu poruke sadržavaju taj pojam. Kako to može dati previše rezultata, ograničit ćemo pretraživanje:

će dati sve mailove, isfiltrirane po kriteriju pošiljatelja (f=From). Slično tome:

će pronaći sve mailove koji su poslani na adresu "netko@domena.hr". Vodeći se tom logikom, "c" označava Cc: polje, dok "a" označava "Address" dakle sva adresna polja: From:, To:, Cc:. Nadalje, "s" označava "Subject:" polje:

Analogno tome, "b" označava "body", dakle tijelo poruke. Moguće je i kombinirati:

ili

će pronaći sve mailove gdje se pojam "Pero" može pronaći u tijelu poruke ili u From: zaglavlju. Moguće su i negacije:

će pronaći sve mailove koji ne potječu od korisnika "pero@pero.hr". Zanimljiva je opcija koja omogućava približno pretraživanje:

koje će pronaći sve retke Subject iz zaglavlja koji imaju najviše jednu grešku od navedenog pojma, dakle uvjete zadovoljava i "pojam" i "dojam". Ukoliko vamje potreban samo dio pojma (*substring*), upotrijebit ćete:

koji će pronaći sve pojmove u Subjectu koji sadržavaju "ivati", npr. "otkrivati", "prikrivati" i slično. Vremenske odrednice možete upotrijebiti na način:

će pronaći sve mailove unazad jedan mjesec do danas, sve mailove stare između 3 i 6 tjedana, te sve mailove starije od 6 godina (respektivno). U zadnjem slučaju, mairix će pronaći sve mailove od 1. prosinca 2009. do danas, i koji u Subjectu imaju "pojam". Ukoliko vas zanima veličina mailova, upotrijebit ćete sličnu sintaksu:

što će pronaći sve mailove veće od jednog kilobajta, sve mailove između jednog i pet kilobajta, te sve mailove manje od 5 megabajta.

Čitanje mailboxa koji se stvori pretraživanjem je lako, u ovisnosti o mail klijentu kojeg rabite.

Dokumentacija također dolazi u paketu, i nalazi se u HTML formatu u datoteci /usr/share/doc/mairix/mairix.html. Službeno web-sjedište projekta možete naći na adresi: http://www.rpcurnow.force9.co.uk/mairix/.

Snaći se u sve većoj količini maila kojeg primamo je sve teže. Ukoliko pri tome napomenemo kako mnogi imaju višegodišnje arhive, sasvim je jasno da "stog sijena" iz naslova svake godine biva sve veći. Većina korisnika, pretpostavljamo, svoj mail čuva na klijentskoj strani (Thunderbird, Outlook), te se pouzda u ugrađene tražilice kako bi našli neki stari mail koji im je potreban. Ovaj pristup ima sve mane koje se pojavljuju kad se podaci nalaze na korisničkom računalu: kvarovi diskova, virusi, krađe laptopa - sve to može dovesti do gubitka vaših podataka. Pri tome je sasvim uobičajeno da korisnici nerado rade zaštitne kopije, pa se ne može doći do podataka iz zaštitne kopije (*backupa*), jer *backup* jednostavno ne postoji.

Čuvanje maila na samom poslužitelju, s druge strane, ima smisla. Poslužitelji imaju kakav-takav sustav zaštitnih kopija, dostupni su s bilo kojeg drugog računala (bilo preko korisničke ljuske, bilo POP3/IMAP protokolima ili webmailom). Današnje brzine procesora i jeftin diskovni prostor ne sprječavaju čuvanje čak i velikih korisničkih pretinaca e-pošte. Snalaženje pri tome olakšavaju neki alati, poput grepmaila.

Radi se o alatu kojem ime govori sve (kao i mnogim drugim alatima): pretražuje pretince e-pošte u potrazi za zadanim kriterijima. A grepmail podržava zaista sve: pretražuje po datumu, pošiljatelju, primatelju, svim zaglavljima, sadržaju i tako dalje. Pri tome mu ne smeta ni ako je pretinac komprimiran sa gzip-om...

Grepmail se poziva slično kao i druge naredbe:

U ovisnosti o odabranim opcijama, grepmail će isfiltrirati odabrani pretinac i rezultat ispisati na standardni izlaz. Kako je ovo nepraktično, najbolje je izlaz preusmjeriti u datoteku, sa standardnim znakom ">", npr:

Gornja će naredba potražiti riječ "upgrade" u svim vašim porukama u dolaznom pretincu i kreirati datoteku "search" sa isfiltriranim porukama. Ona je u standardnom mbox formatu i može se pročitati sa bilo kojim programom za čitanje pošte, uključujući alpine i mutt. Potrebno je samo pokrenuti vaš program sa odgovarajućom opcijom:

Nakon što ste pronašli što vas zanima, potrebno je jednostavno obrisati datoteku s rezultatima. No, postoje i dva alata koji dodatno olakšavaju uporabu grepmaila, na način da primaju sve parametre grepmaila, provode pretragu, kreiraju datoteku s rezultatima, pokreću odgovarajući mail klijent, te nakon uporabe brišu privremenu datoteku s rezultatima.

Za korisnike programa mutt, možete skinuti skriptu "grepm" na adresi http://www.barsnick.net/sw/grepm.html, dok za korisnike programa pine možete rabiti "grepine" na adresi http://www.dfki.de/~pietsch/software. U trenutku pisanja teksta skripta je bila nedostupna, pa je zato možete skinuti ovdje (možete maknuti ekstenziju .txt).

Uz paket grepmail dolazi uobičajena dokumentacija u /usr/share/doc/grepmail i standardna stranica manuala. Ipak, kroz primjere je, vjerujemo, najlakše učiti, pa ćemo navesti nekoliko češćih načina primjene grepmaila.

Ispis broja poruka u odabranom pretincu:

Ispis broja poruka, ali rekurzivno kroz sve direktorije (rekurzija se, naravno, može primjeniti i na druge načine pretrage, ne samo na ispis broja poruka):

Ispis svih poruka od korisnika "Pero" (prezime nije bitno), rekurzivno kroz cijeli mail direktorij (pretražuju se samo zaglavlja zbog opcije -h):

Ispis svih poruka koje su poslane ili primljene s neke e-maila adrese, rekurzivno (opcija -Y omogućava prepoznavanje i razlomljenih zaglavlja i automatski uključuje opciju -h):

Grepmail ima velike mogućnosti filtriranja po datumu (opcija -d), zahvaljujući Perl modulu Date::Manip. Također, grepmail možete ulančavati (*pipeati*), isto kao i većinu drugih naredbe.

Ispis svih poruka poslanih jučer, ali koje **ne sadržavaju** (opcija -v) riječi "rezultati" **ili** "mjerenja" (pretraga je *case-insensitive* zbog opcije -i):

Ispis svih poruka primljenih između 1. prosinca i 10. prosinca 2008:

Ispis svih poruka koje sadrže riječ "izvješće" i poslane su poslije 8 sati na određeni datum:

Ispis svih poruka pristiglih od korisnika "Marko" nakon drugog ponedjeljka u prosincu 2008. godine:

Svaki servis kojeg instalirate na poslužitelj, osim izvršnih datoteka, sadrži i pomoćne datoteke (manual, popis promjena i slično), ali i inicijalizacijske skripte, koje nas u ovom trenutku najviše i zanimaju. Debian rabi inicijalizacijske skripte u Unix System-V stilu, a s njima je najlakše upravljati pomoću malog uslužnog programa - update-rc.d.

Ruku na srce, za postojanje ovog programa gotovo i ne morate znati, jer sve što on može, možete napraviti i ručno. Update-rc.d se rabi najviše u instalacijskim skriptama (preinst, postinst, prerm, postrm) koje se nalaze u direktoriju /var/lib/dpkg/info. Razlog zašto ga spominjemo je problem kojeg smo zaprimili na SysHelpu, gdje se jedan servis nije pokretao nakon restarta.

Ukratko, svi servisi koji se moraju pokrenuti prilikom pokretanja poslužitelja, moraju imati simbolički link na inicijalizacijsku datoteku, /etc/init.d/<servis>. Linkovi se nalaze u direktorijima /etc/rcX.d, gdje je X broj od 0-6, odnosno slovo "S". Ovi brojevi odgovaraju trenutnoj razini rada, odnosno u originalu ***runlevel***.

Srećom, Debian ima jako pojednostavljen sustav, pa je:

0 ili N	sustav se gasi
1 ili S	jednokorisnički način rada
2 do 5	višekorisnički način rada, mreža, GUI
6	sustav se restarta

Simbolički linkovi na startup datoteku se nalaze u tim direktorijima, a nazivaju se ili "S" (start) ili "K"

(kill), primjerice:

Dakle, vsftpd će se pokretati u runlevelima 2 do 5, a gasiti u runlevelima 0, 1 i 6. Upravo je to pretpostavljena vrijednost naredbe update-rc.d:

Bez ikakvih opcija, update-rc.d će staviti "S" linkove u rc.d direktorije od 2 do 5, a "K" linkove u rc.d direktorije 0, 1 i 6.

Brojevi iza slova omogućavaju finu gradaciju redosljeda pokretanja servisa, pa će se servis vsftpd (S20) pokrenuti poslije web poslužitelja apache2 (S09). Ukoliko želimo da se neki servis starta prije ili poslije nekoga drugoga, update-rc.d ćemo upotrijebiti u ovom obliku:

Linkovi će tada imati imena S24vsftpd, odnosno K24vsftpd.

Update-rc.d prima i druge parametre, pa tako možemo osigurati da se servis A starta prije i gasi poslije servisa B:

Servis A će se tako postati S80servisA i K20servisA, pa će se startati prije servisa B (K80 < K90), ali i gasiti poslije njega (K20 > K10).

Pomoću ove naredbe moguće je brzo onemogućiti neki servis:

Naravno, postoji još opcija, pa je tako moguće zasebno za svaki runlevel odrediti kada će se neki servis pokretati, odnosno gasiti. Sintaksa je sljedeća:

Servis će se startati vrlo rano, a gasiti vrlo kasno, što je pogodno ukoliko se radi o nekom bazičnom servisu o kojem ovisi drugi servisi (daemoni). Obično se uzima da je vrijeme gašenja "100 minus vrijeme starta" (start 10, stop 100-10=90), ali to nije uvijek slučaj.

Ukoliko vas zanima što se kolegi koji je prijavio problem dogodilo, probajte sami zaključiti (rabio je jedan drugi alat za upravljanje):

Ukoliko pobliže pogledate, **svi** su linkovi "K" linkovi, dakle jadni dovecot nije imao šanse za start niti u jednom runlevelu. Rješenje je jednostavno, može se riješiti preko update-rc.d, ali i ručno:

ili

Poslije toga treba pozvati naredbu update-rc.d, najbolje u obliku u kojem se nalazi u postinst skripti (kod vas je broj možda drukčiji, ali sam broj nije od presudne važnosti):

I problem je riješen.

Ukoliko ste zaboravili root zaporku i nemate drugog načina da je pronađete ili promijenite, ne treba brinuti, jer postoji jednostavno rješenje. Kod nekih je distribucija Linuxa dovoljno sustav podignuti u *single user* načinu rada, no kod Debiana je i tu potrebno unijeti rootovu zaporku. Kod Debiana je potrebno iskoristiti mogućnosti koje nam boot procedura omogućava, kao to je prenošenje parametara inicijalizacijskom procesu.

Koji će točno postupak biti primjenjen, ovisi o *bootloaderu* koji rabite: LILO ili GRUB. Standardno je na CARNetovim poslužiteljina instaliran LILO. No, GRUB je ugodnijeg sučelja i standardan je na mnogo drugih distribucija, pa odluka koji ćete rabiti ovisi o vama i vama bitnim detaljima. Za standardnu uporabu, obadva *bootloadera* su sasvim zadovoljavajuća.

Prvo ćemo navesti postupak za LILO, koji je nešto jednostavniji nego postupak za GRUB.

LILO

Kod pokretanjaposlužitelja morate držati stisnutu tipku Shift, kako biste dobili prompt **boot:**. Nakon pojave prompta, otkucajte "linux init=/bin/sh" pa će cijeli redak izgledati ovako:

Nakon pojave rootovog prompta #, možete nastaviti po uputama.

GRUB

Ukoliko imate instaliran GRUB, postupak je sljedeći:

1. na GRUB promptu pritisnite tipku "e" (ovim ćete pokrenuti "edit" način rada). Ako se GRUB izbornik ne pojavljuje, probajte stisnuti tipku ESC.
2. sa strelicama odaberite kernel koji želite bootati
3. opet pritisnite tipku "e"
4. otipkajte "init=/bin/sh" na kraju retka (Enter se podrazumijeva)
5. pritisnite tipku "b" kako bi bootali poslužitelj s navedenim kernelom i opcijama

Nakon što ste dobili standardni rootov prompt, postupak je potpuno jednak za LILO i GRUB. Rootova ljuska u kojoj se nalazite ima vrlo limitirane mogućnosti (nema skoro nijedne naredbe na koju ste navikli). No, sasvim je dovoljna za ono što nam treba.

POSTUPAK

Prvo moramo mountati / (root) particiju tako da možemo po njoj pisati (ona je u ovom trenutku mountana samo za čitanje):

Kako se naredba "passwd" nalazi na /usr particiji, u slučaju da nije mountana, mountajte je:

Dalje je lagano:

Možete također s nekim editorom (vi, vim) direktno urediti /etc/shadow datoteku. Dovoljno je u njoj obrisati sve znakove iz drugog polja u retku za root korisnika, pa će cijeli unos izgledati otprilike ovako:

Nakon reboota (/sbin/reboot), root korisnik će imati zaporku koju ste mu dodijelili, ili praznu ako ste direktno editirali /etc/shadow datoteku (naravno da ćete root korisniku dodijeliti novu zaporku čim se sustav podigne).

Dovršimo operaciju:

Ovo nije jedini način promjene zaporke. Uvijek ostaje mogućnost podizanja sustava s neke *rescue* CD distribucije, poput Knoppixa. Kako ove distribucije mogu pomoći i kod puno složenijih situacija, obradit ćemo ih drugi put.

Još prije smo u ovom članku na portalu za sistemce opisali upotrebu pGina-e s freeradius pluginom za autentikaciju korisnika u javno dostupnim računalnim učionicama. Budući je u međuvremenu tvorac pGina-e napustio projekt, preuzela ga je open source zajednica. Razvoj projekta može se pratiti na sourceforge-u.

Trenutno je aktualna verzija 2.0 Alpha u kojoj je napokon uspostavljena funkcionalnost s Windows 7 OS-om, ali ima manje mogućnosti od posljednje stabilne verzije, 1.8.8. Za upotrebu na Windows XP klijentima, za sada preporučujemo samo stabilnu verziju, koju možete preuzeti ovdje. No, ako prelazite na Windows 7, osuđeni ste na alpha verziju.

Po instalaciji pGina-e 2.0 Alpha, mogu se konfigurirati sljedeće postavke:

Na žalost, još nije podržana interakcija s windows domenom, već se svi korisnici stvaraju isključivo kao lokalni. Nadamo se da će se u budućnosti ovaj projekt i dalje razvijati i da će mu se dodavati funkcionalnosti kako bi se proširile ove, za sada rudimentarne, mogućnosti.