O'REILLY ONLamp.com BSD DEVCENTER

Mrezno Filtriranje na osnovu Operativnog Sistema

by Avleen Vig
02/16/2006

Odrzavate heterogenu mrezu I zelite da omogucite razlicite Quality of Service agreements I mrezne restrikcije na osnovu klijentovog operativnog sistema. Sa pf i altq, mozete ograniciti kolicinu protoka dostupnu korisnicima razlicitih operativnih sistema, ili da naterate da izlazni web saobracaj ide preko transparentog proxy-ja koji filtrira saobracaj. Ovaj clanak objasnjava kako da instalirate pf, altq, i Squid na vasem FreeBSD ruteru i web proxy-ju da biste postigli ove ciljeve.

Cilj Misije

U idealnom okruzenju, nebi postojala potreba za kontrolom protoka, filtriranje zanovano na OS fingerprintu, ili cak Quality of Service (QoS). Nekoliko faktora u stvarnom zivotu zahtavaju promenu plana. Protok nije besplatan, I mnogi Internet Provajderi naplacuju korisnicima na osnovu kolicine protoka koji koriste. Crvi, virusi, I zarazeni sistemi mogu dovesti do povecanih troskova za protok. Kada se crv W32.Slammer pojavio, koji je preopterecivao konekcije zarazenih mreza, mnoge kompanije su videle da se njihovi mesecni racuni za konekciju naglo skocili zbog zaobracaja koji je crv pravio.
Filtrirajuci vase konekcije na osnovu operativnog sistema mozete delimicno pomoci da se takve situacije vise ne ponavljaju. Dok cu se ja fokusirati na filtriranju sapbracaja koji dolazi od Windows sistema, ovaj process se moze primeniti jednako I na BSD, Linux, Mac OS, ili ostale operativne sisteme izlistane u pf.os fajlu na vasem sistemu. Ovo narocito moze biti korisno osobama koje koriste starije verzije OS-a koje nisu ili se nemogu patch-ovati ali I dalje zahtevaju neku mreznu konekciju.
Kao nastavak transparentnog filtriranja, moguce je I filtriranje prema sadrzaju, sa alatima kao sto su npr squidGuard dozvoljavajuci deci I desktopima u firmama I slicnim da relativno sigurno surfuju.

Alati za posao

Tokom mog istrazivanja za ovaj clanak, nekoliko osoba me je pitalo zasto sam izabrao da koristim BSD, pf, altq, i Squid za ovaj zadatak. Ostali alati su blizu da omoguce zahtevanu funkcionalnost, ali ni jedna ne ispunjava zahteve tako dobro kao ove. Linux i iptables mogu da rade sa Squid-om da omoguce transparent proxy ali nemogu da filtriraju konekcije na osnovu operativnog sistema. Iako postoje I ostali proxy serveri, Squid je jedan od najboljih koji danas postoje.
Vazno je napomenuti da OS fingerprinting radi samo sa TCP SYN paketima, koji zapocinju TCP komunikacije, a ne sa trenutno uspostavljenim konekcijama ili UDP komunikacijama. Dok ovo nece biti problem za vecinu sistemskih I mreznih administratora, mozda cete zeleti da obratite vise paznje na vasa pravila za UDP filtriranje.

Instaliranje pf i altq

pf i altq omogucavaju filtriranje paketa I kontrolu protoka. Njihova veza je slicna kao kod IPFIREWALL i DUMMYNET, gde isti fajlovi sa pravilima konfigurisu I pf i altq.
Dok se pf moze koristiti univerzalno, altq zahteva podrzanu mreznu karticu. Dobre vesti su da je vecina mreznih kartica koje se danas koriste podrzana. Pogledajte man 4 altq odeljak Podrzani Uredjaji da biste nasli listu podrzanih mreznih kartica.
Jednom kada utvrdite da je vas uredjaj podrzan, dodajte pf i altq vasem kernelu. Moracete da ponovo kompajlirate vas kernel kao sto je to objasnjeno u FreeBSD Handbook-u. Prvo, dodajte nekoliko opcija na kraju vaseg kernel konfiguracionog fajla:

```
device pf
options ALTQ
options ALTQ_CBQ
options ALTQ_RED
options ALTQ_RIO
```

```
options ALTQ_HFSC
options ALTQ_CDNR
options ALTQ_PRIQ
```

Beleska: Ako instalirate altq na viseprocesorski sistem, dodajte options ALTQ_NOPPC vasoj konfiguraciji pre nego sto rekompajlirate kernel.
Nakon sto ste kompajlirali kernel I ponovo pokrenuli sistem, testirajte pf da biste bili sigurni da se pravilno instalirao komandom pfctl -s rules. Ako vidite gresku pfctl: /dev/pf: No such file or directory, pf se nije pravilno instalirao. Ako vidite gresku No ALTQ support in kernel ALTQ related functions disabled, pf radi ali altq ne radi. U ovom zadnjem slucaju, jos uvek cete moci da naterate korisnike da idu preko transparentnog proxy-ja, ali necete moci da ogranicite protok koristeci altq.

Instaliranje Squid-a sa Podrskom za Transparentno Filtriranje

Instalirajte Squid komandom
:

```
% cd /usr/ports/www/squid && make config install clean
```

Ovo ce vam dati listu mogucih opcija za kompajliranje Squid-a. Da omogucite podrsku za transparent proxy , odaberite SQUID_PF. Mozete I izabrati ili ponistite ostale opcije. Cesto nalazim da je SQUID_SNMP koristan za prikupljanje I graficko prikazivanje statistike koristeci RRDTool. Jednom kada se Squid instalira, editujte */usr/local/etc/squid/squid.conf*. Izaberite sledece opcije:

```
http_port YOUR_PROXY_IP:3128
http_access deny to_localhost
acl our_networks src YOUR_NETWORK/24
http_access allow our_networks
visible_hostname YOUR_HOSTNAME
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Zamenite *YOUR_PROXY_IP* sa IP adresom na kojoj ce vas proxy server da slusa, YOUR_NETWORK/24 sa rasponom adresa vase interne mreze (na primer, 192.168.0.0/24), i *YOUR_HOSTNAME* sa imenom hosta koji zelite da prikazete korisnicima u error porukama. *YOUR_HOSTNAME* nije obavezan ali je veoma koristan ako imate klaster proxy servera koji dele zajednicki front end kao npr load balancer.
Iako mozete poceti menjanjem samo ovih opcija, trebalo bi da odvojite malo vremena I za ostatak vaseg *squid.conf* fajla i da ga prilagodite vasim potrebama. Tokom vremena, mozda cete morati da prilagodite razne ostale opcije kao sto su velicina cache-a ili vreme prekida konekcija. Ako potrosite sat vremena na upoznavanju sa raznim opcijama Squid konfiguracionog fajla to vam moze ustedeti vreme I trud u buducnosti.

Filtriranje Sadrzaja sa squidGuard-om

Ovo je opcioni korak za one koji zele da koriste filtriranje sadrzaja. Ja koristim squidGuard da filtriram sadrzaj na mojoj kucnoj mrezi, da bi onemogucio mojoj deci da idu na sajtove za koje ja mislim da nisu odgovarajuci. Aplikacija za ovo ima mnogo, posto squidGuard nudi crne liste za reklame, sadrzaj za odrasle, droge, kockarske sajtove, sajtove mrznje, I ostalo.

Insltaliranje I Konfigurisanje squidGuard-a

Insltalirajte squidGuard komandom:

```
% cd /usr/ports/www/squidguard && make install clean
```

Kao i Squid, I squidGuard je veoma lak za konfigurisanje—sto je dobra strana, zato sto ste zadnja dva sata potrosili na konfigurisanje Squid-a, zar ne?
Kopirajte */usr/local/etc/squid/squidGuard.conf.sample* u */usr/local/etc/squid/squidGuard.conf* I otvorite ga u editoru po vasem izboru. Ako zelite da filtrirate prema odredjenom vremenu u toku dana, procitajte vodic za Konfigurisanje squidGuard-a. Za sada, filteri uvek trebaju biti omoguceni. Uklonite postojecu source sample-clients grupu, I kreirajte novu grupu na njegovom mestu sa vasim rasponom mreze:

```
source localnet {
      ip 192.168.0.0/24
}
```

Na kraju fajla, zamenite postojecu acl grupu novom grupom:

```
acl {
   default {
         pass !ads !drugs !gambling !porn all
         redirect http://YOUR_WEBSERVER/cgi-bin/squidGuard.cgi? \
            clientaddr=%a&clientname=%n&clientuser=%i&clientgroup=%s&url=%u
   }
}
```

Beleska: Ne stavljajte kosu crtu u redirect liniji.
Ovaj ACL ce zabraniti pristup bilo kojem URL-u izlistanom u bazi podataka za reklame, droge, kockanje, I porno bazi podataka. I ostale baze podataka su dostupne; Postoji lista u konfiguracionom fajlu pre acl grupe koje mozete da odaberete. Posebno obratite paznju na redirect deo, koji daje reference CGI-ju. U slucaju da korisnik poseti zabranjeni sajt, squidGuard ce preusmeriti zahtev na ovaj URL. Primer squidGuard.cgi je dostupan za download. Stavite ovo na web server I izmenite redirect da promeni YOUR_WEBSERVER u ime vaseg web servera.
Morate konfigurisati i Squid da koristi squidGuard sa redirect_program direktivom u */usr/local/etc/squid/squid.conf*. Otvorite ovaj fajl jos jednom I potrazite liniju:

```
#  TAG: redirect_program
```

Ispod ovog, dodajte komandu:

```
redirect_program /usr/local/bin/squidGuard
```

Konfigurisanje pf-a i altq-a

Doslo je vreme da se instalirani programi I promene zajedno uklope tako sto cemo konfigurisati paket filrer i kontrolor protoka. Kao default, FreeBSD cuva pf konfiguraciju u */etc/pf.conf*. Primerak konfiguracionog fajla koji postoji je vaoma dobro dokumentovan. Za pocetak, uzmite u obzir primer filtriranja sa Mreznim Prevodjenjem Adresa (NAT). Ovaj primer pretpostavlja da imate internet konekciju sa dostupnim protokom od 3Mb downstream i 512Kb upstream:

```
ext_if       =  "fxp0"
external_addr =  "1.2.3.4"
int_if       =  "fxp1"
internal_net  =  "192.168.0.0/24"
proxy_server  =  "192.168.0.10"

altq on $ext_if bandwidth 512Kb cbq queue { windows_out, trusted_out }
queue windows_out bandwidth 20%
queue trusted_out bandwidth 80%
altq on $int_if bandwidth 3Mb cbq queue { windows_in, trusted_in }
queue windows_in bandwidth 20%
queue trusted_in bandwidth 80%

rdr on $int_if inet proto tcp from $internal_net os "Windows" to any \
  port www -> $proxy_server port 3128
nat on $ext_if from $internal_net to any -> ($ext_if)

pass out quick on $ext_if inet proto tcp from $proxy_server \
  to any port www keep state queue windows_out
pass out quick on $ext_if inet proto tcp from $internal_net os "Windows" \
  to any keep state queue windows_out
pass out quick on $ext_if inet proto tcp from $internal_net os "unknown" \
  to any keep state queue windows_out
pass out on $ext_if inet proto tcp from $internal_net \
  to any keep state queue trusted_out
```

```
pass out quick on $int_if proto tcp from any to $proxy_server \
  queue windows_in
pass out on $int_if proto tcp from any to $internal_net queue trusted_in
```

Prvih pet linija odredjuju varijable koje ce skup pravila cesto koristiti. Varijable su veoma korisne u konfiguracionim fajlovima. Ako ih koristite pravilno, Ustedecete dosta vremena u buducnosti ako vam zatreba da promenite IP adrese, mrezne interfejse, protokole, ili bilo sta drugo. Kada odredite varijable u vasim pravilima, kasnije im pristupate kao $variable_name.

Sest altq konfiguracionih linija koje slede postaljaju vrednosti za kolicinu dostupnog protoka. Protok se kontrolise na interfejsu gde paketi napustaju ruter. Paketi koji idu od mreze prema internetu napustaju ruter na $ext_if, tako da postavljamo $ext_if na 512Kb. Isto tako, paketi koji dolaze na nasu mrezu od interneta napustaju ruter na $int_if, tako da postavljamo pravila $int_if na 3Mb. pf razume b, Kb, Mb, i Gb, koji predstavljaju bitove, kilobite, megabite, i gigabite u sekundi.

altq linija onda odredjuje tip rasporeda (scheduler) koji ce se koristiti da se paketi stave u niz (queue). Ovaj primer koristi cbq. Prema rasporedu se odlucuje kojim ce se redom obradjivati nizovi. Class Based Queuing (CBQ) deli dostupan mrezni protok izmedju dva ili vise niza, gde se svakom nizu dodeljuju paketi prema izvornoj ili odredisnoj adresi, portu ili prema nekom drugom indentifikujucem faktoru (u ovom slucaju operativni system). Nizovi mogu imati I prioritete da bi se neki paketi obradili pre drugih. OpenBSD Packet Queuing stranica sadrzi vise detalja o razlivitim tipovima rasporeda.

Kraj ove linije odredjuje koji ce se nizovi ograniciti.

Sledece dve linije odredjuju same nizove kojima ce altq rukovoditi. Format je veoma razumljiv I izricito kaze koliko je protoka dostupno za svaki niz. Ovo mozete odrediti u procentima ili fiksnom vrednoscu. Ako gledate na vasu internet konekciju kao jedan put, niz definise na koliko traka razliciti paketi mogu da putuju. Sto vise traka koriste, vise ce se podataka prenesti. Ova konfiguracija se onda ponavlja za dolazeci protok.

rdr linija je kljucna za filtriranje. Ona odredjuje da sav TCP saobracaj (proto tcp) od interne mreze ($internet_net) koji potice od Windows sistema I ide na bilo koju drugu adresu na port 80 (os "Windows" to any port www), bude preusmeren na proxy server na port 3128 (-> $proxy_server port 3128).

nat linija postavlja mrezno prevodjenje adresa, sto dozvoljava internoj mrezi da komunicira sa internetom.

Primetite da se svaka linija pf pravila zavrsava sa windows_in, windows_out, trusted_in, ili trusted_out. Ovo su cetiri niza predhodno postavljenih kao deo altq pravila; ovo omgucava pf-u da zna koje ce nizove pf i altq da koriste kada obradjuju pakete. Ovi nizovi su potpuno opcionalni. Ako ih izostavite iz pf pravila altq nece moci da ogranicava protok za pakete koji se podudaraju sa tim pravilima.

Najjednostavniji format pf pravila je:

```
<pass|block> <in|out> on <interface> from <src> to <dst>
   [keep state] [queue <queue_name>]
```

Mozete zadavati I brojeve portova I protokole.
Ovaj skup pravila sadrzi I specijalno pravilo za saobracaj sa nepoznatog OS-a. Patch-evi za operativni system i IP stacks mogu da promene fingerprint paketa. Filtrirajuci sav nepoznati saobracaj preko Windows nizova pomaze da se izbegnu buduci problemi.
Zadnja dva pravila kontrolisu ulazni saobracaj. Zbog toga sto je nemoguce znati dali je odredisni sistem Windows, pf ne moze da filtrira dolazni saobracaj na osnovu operativnog sistema koji ne koristi proxy. U praksi, ovo ne predstavlja neki veliki problem zato sto web saobracaj sa ovih sistema vec putuje kroz proxy koji ogranicava protok. Ako Windows sistemi imaju peer-to-peer softver kao npr BitTorrent, taj saobracaj nece ici kroz proxy. U ovom slucaju, stavite takav saobracaj u niz tako sto cete odrediti portove koji ce se koristiti, ovako

```
bittorrent_ports = "6881:6999"
pass out quick on $int_if proto { tcp, udp } from any to any \
  port $bittorrent_ports queue windows_in
```

~Dalibor Gudzic@soxxx