O'REILLY ONLamp.com BSD DEVCENTER

Proxy Terminologija jedan-na-jedan

by Dru Lavigne
06/19/2003

Ako ste ikada pristupili internetu iz kancelarijskog okruzenja, verovatno je vasa komunikacija prosla kroz proxy. U sledecih nekoliko clanaka, diskutovacu o prednostima koriscenja proxy-ja i demonstrirati konfiguraciju nekoliko proxy-ja dostupnih iz FreeBSD kolekcije portova.

Mozda jos uvek neznate sta ustvari proxy radi. Odvojite malo vremena i idite posetite http://www.freebsd.org/ports/, i u "Search for:" polju ukucajte rec "proxy". iznanadicete se koliko je proxy-ja dostupno, mozda i malo uplasiti od svih onih termina koji ih opisuju: reverse proxy, arp proxy, transparent proxy, itd. Ostanite samnom i opisacu vam vecinu najcescih proxy termina. Onda ce terminologija poceti da dobija smisao kada pogledamo konkretne primere.

Proxy Osnove

U svojoj najednostavnojoj formi, proxy je parce softvera koje "radi u ime" mreznog klijenta. imajte na umu da u mrezi, klijent je entitet koji kreira mrezni zahtev i server je entitet koji odgovara na taj zahtev. Na primer, vas web citac je klijent koji zahteva web sadrzaj od web servera.
U zavisnost od proxy-ja, postoji nekoliko nacina u kojima on moze da "radi u ime" klijenta. Prvi je da zauzme mesto klijenta, sto znaci da klijent nikada ne komunicira direktno sa serverom. Umesto toga, klijent kreira konekciju do proxy-ja i onda proxy kreira konekciju do servera, dobija neki odgovor od servera, i prenosi ga nazad do klijenta. Ovo je cest slucaj sa web citacima i izgleda ovako:

```
web browser ----------------> proxy ---------------> web server
            <---------------       <--------------
```

Sledeci put kada idete na web stranicu, pogledajte pri dnu vaseg GUi web citaca. Ako kaze "Waiting for www.google.ca", vas web citac se konektuje direktno do odredjenog web servera. Ali, ako kaze nesto kao "Connecting to 192.168.1.1", vas zahtev ide kroz proxy koji je lociran na toj adresi.
Koriscenje prixy-ja daje mrezi nekoliko prednosti. Prvo, jedini kompjuter u mrezi koji zahteva javnu iP adresu je onaj nak kome se proxy softver nalazi. Ovo znaci da cela mreza moze imati pristup internetu, cak iako ste u mogucnosti da dobijete samo jednu iP adresu od vaseg internet Provajdera. Osim sto stedite, ovim isto tako dobijate i malo na sigurnosti zato sto sakriva vasu mrezu od interneta. Jedina iP adresa oju internet host vidi je iP adresa proxy-ja.
Postoje i druge prednosti koriscenja proxy-ja koje se ticu bezbednosti. Posto svi internet zahtevi prozale kroz proxy, vecina proxy-ja vam omogucava da konfigurisete koji zahtevi su dozvoljeni a koji su zabranjeni. U stvari, mogucnosti i lakoca konfigurisanja su obicno ono sto trazite kada ocenjujete koja proxy aplikacija je najpogodnija za vasu mrezu.
Proxy ce isto tako sadrzati i cache ranijih zahteva koji moze da ustedi protok. Ovo je slicno kesu vaseg web citaca, osim sto cela mreza moze imati predonst kesiranog sadrzaja. Ako je jedan korisnik vec zahtevao URL, proxy ce kopirati sadrzaj u njegov kes. Kada sledeci zahtev za taj URL stigne do proxy-ja, vratice kesirani sadrzaj umesto da ide ponovo na internet da bi preuzeo zahtevanu web stranicu. imajte na umu da zasticeni sadrzaj nece biti kesiran. Na primer, ako date informacije o vasoj kredintoj kartici na stranici ciji URL pocinje sa "https://", ta informacija nece biti kesirana od strane proxy-ja.
Proxy koji radi kesiranje ce koristiti algoritam da odredi koliko cesto da "obnavlja" sadrzaj svog kesa. Kes je odlican za smanjenje internet protoka, ali korisnici ne zele da dobiju stranicu koja je bila sacivana u kesu vise

od mesec dana, posebno ako se originalna stranica promenila od tada. Neke stranice su i dinamicnije od ostalih. Na primer, Slashdot cesto tokom dana menja svoj sadrzaj dok iANA retko menja sadrzaj. Algoritam sadrzi kriterijume koji ce pomoci proxy-ju da odredi kada da osvezi svoj kes i koje stranice prvo da osvezi. Najcesce korisceni algoritmi su iCP (internet Cache Protocol), CARP (Cache Array Routing Protocol), i HTCP (HyperText Caching Protocol). Mozete se vise informisati za sva tri protokola na iCP stranici. Ovi protokoli imaju dodatnu prednost u tome da oni dozvoljavaju deljenje informacija svog kesa izmedjuvise proxy-ja. Ovo omogucava vecoj mrezi da ima raspodeljeni kes i onda internet zahtevi mogu biti izbalansirani.
Postoji i losa strana koriscenja proxy-ja: klijent se mora unapred konfigurisati da bi ga koristio. Ovaj proces je poznat kao "modifikacija klijenta". Na primeru web citaca, on zahteva od korisnika da ide u "Preferences" ili "Options" deo njegovog web citaca, da nadje "Proxy" odeljak, i ukuca iP adresu proxy-ja i broj porta na kojem proxy aplikacija slusa. Ostale aplikacije mogu zahtevati posebab proxy klijent softver instaliran i konfigurisan na svakoj masini kojoj treba pristup do proxy-ja.
Ovo nas dovodi do drugog nacina na koji proxy moze da "radi u ime" klijenta: kao "transparentni" proxy. Transparentno znaci da nista nije predhodno konfigurisano na klijentu; u stvari, korisnici mozda nece ni znati da njihovi zahtevi idu kroz proxy. Transparnetny proxy ce presresti klijentov zahtev, proveriti dali je dozvoljen, i onda ga prosledi do servera. Ovaj tip proxy-ja je cesto integrisan u zastitni zid koji vam dozvoljava da konfigurisete proxy kao deo polise sigurnosti mreze.

Proxy-ji konkretne aplikacije

Sada je dobro vreme da se pomene da se vecina proxy-ja smatra "application-specific". Pogledajte pazljivije rezultate vase pretrage iz FreeBSD liste protova. Primecujete da tamo postoje RealAudio proxy-ji, iRC proxy-ji, HTTP proxy-ji, FTP proxy-ji, SMTP proxy-ji, i tako dalje. Za svaku internet aplikaciju, postoji odvojeni proxy softver. Ovo je veoma vazno; ovo oznacava pravu snagu i konfigurabilnost proxy softvera.
Zamislite na trenutak tipicnu mrezu zasticenu zastitnim zidom. Korisnici iza zastitnog zida zele da surfuju internetom i da salju i primaju e-mail poruke. Zastitni zid je konfigurisan da dozvoli izlazne portove 25 (SMTP), 80 (HTTP), i 110 (POP3) i da dozvoli odgovore na te pakete nazad do mreze. Kada paket dodje do zastitnog zida, njegov header ce se uporediti sa pravilima zastitnog zida da bi se proverilo da li je broj porta i izvorna/odredisna iP adresa dozvoljena.
To zvuci prilicno sigurno, zar ne? Dokle god je podatak u paketu ono sto kaze da jeste, onda je bezbedno. Jedno od ogranicenja baze pravila zastitnog zida je da je ogranicen na informacije sadrzane u header-ima paketa. (Pogledajte Objasnjenje Slojeva TCP Protokola.) Da bi zastitni zid mogao da proveri podatke paketa, on mora da *razume* te podatke. Ovo je poznato kao "provera sadrzaja" i zahteva dodatni softver koji razume sadrzaj koji se proverava. Kao sto ste mozda vec pogodili, taj dodatni softver ce biti aplikacioni proxy.
Uzmimo HTTP paket kao primer. Jedan od korisnika salje van paket namenjen za port 80. Dolazi do zastitnog zida koji dozvoljava paket, zato sto baza pravila dozvoljava izlazni port 80. Ali, taj paket nije sadrzao HTTP podatke. Umesto toga, korisnik je knofigurisao njegovu p2p aplikaciju da koristi port 80, znajuci da je taj port otvoren na zastitnom zidu. Deljenje fajlova je verovatno zadnja stvar koju je mrezni administrator zeleo da dozvoli u njegovim bezabednosnim polisama, ali i pored toga baza pravila zastitnog zida nije zaustavila nezeljeni paket.
Cak i da je palet sadrzao legitimne HTTP podatke, kako ce zastitni zid znati da li sadrzao i virus, malicioznu ActiveX komponentu ili JavaScript? ili tonu dosadnih pop-up reklama? Zastitni zid to nemoze znati, ali dobar HTTP proxy moze.
Na kraju, tu je i problem autorizacije. Zastitni zid jedino moze doneti autorizacijske odluke na osnovu iP adrese, ali iP adrese mogu biti lazne i vise od jednog korisnika moze sedeti za istim kompjuterom. Nemozete napisati pravilo za zastitni zid koje kaze "Gwendolyn moze da surfuje sa 10.0.0.1 ali Martin nemoze". ipak, proxy se moze konfigurisati da primora korisnika da se autorizuje pre nego sto im se dozvoli pristup internetu kao i da cuva listu dozvoljenih korisnika i njihove dozvoljene lokacije.
Proxy zvuci odlicno, ali zasto vam treba poseban proxy za svaku aplikaciju? iz jednostavnog razloga sto svaka aplikacija koristi razlicite komande. Mozete se setiti koriscenja SMTP i POP3 komandi. Ako ste povezani na SMTP server i pokusate da koristite LiST komandu, dobicete gresku. To je zato sto je LiST POP3 komanda, a ne SMTP komanda. Slicno tome, paket koji sadrzi SMTP podatke ce sadrzati SMTP komandu. SMTP proxy moze traziti validne SMTP komande u podacima samog paketa. U slucaju da nenadje nijednu, paket verovatno ne sadrzi SMTP podatke.
Ako ikada trebate konfigurisati aplikacioni proxy, dobro je znati gde mozete da nadjete komande koje se koriste za svaku aplikaciju, zajedno sa objasnjenjem sta koje komanda radi. Najbolje izvor informacija je RFC

za taj odredjeni protokol. Sada cu vam dati neke reference za najcesce koriscene aplikacije; dok pregledjujete svaki RFC, potrezite odeljak "Commands". Primeticete da se HTTP komande ustvari nazivaju "Methods".

- HTTP, RFC 2616
- FTP, RFC 959
- SMTP, RFC 2821
- POP3, RFC 1939
- iMAP4, RFC 2060
- iRC, RFC 2812

Ponovo cemo se vratiti na skupove komandi za aplikacije, kada budemo izgradili i konfigurisali neke od aplikacionih proxy-ja koji su dostupni u kolekciji portova.

Napredni Proxy Termini

Postoji jos nekoliko drugih proxy termina koje bih pomenula pre zavrsetka ovog clanka. Prvi je "reverse proxy". Setite se definicije proxy-ja: softverska aplikacija koja radi u ime mreznog klijenta. Reversni proxy je suprotnost tome: on radi u ime mreznog servera. Najcesce koriscenje reversnog proxy-ja je da se zastiti web server. Kada korisnik na internetu zahteva podatke od web servera zasticenog reversnim proxy-jem, reversni proxy presrece zahtev i proverava dali su podacu zadrzani u zahtevu prihvatnjivi. Na primer, da podaci ne sadrze neke ne-HTTP podatke ili neke maliciozne HTTP komande. Ako su podaci prihvatljivi, reversni proxy ce primiti zahtevani sadrzaj od web servera i proslediti ga do originalnog korisnika. Na ovaj nacin, korisnici na internetu nikada ne pristupaju direktno vasem web serveru.
Jos jedan tip proxy-ja je "ARP proxy". ARP se koristi uvek kada TCP/iP host treba poslati paket. Pre nego sto interfejs hosta moze da kreira okvir, koji ce biti poslat ka mrezi, on mora znati hardversku adresu hosta koji ce primiti okvir. Posto sam paket sadrzi samo iP adresu, ARP se koristi da odredi koja hardverska adresa je dodeljena toj iP adresi.
Ponekad, ipak, taj ARP nece moci da pronadje hardversku adresu. Pogledajmo jednostavan primer:

```
web server ----- firewall/NAT device ------ internet router ----- web browser

|-----DMZ-------|                 |------------internet-----------------|

|--- 10.0.0.0 --|                   |-- 1.2.3.0 --|
```

Ovde se web server nalazi u DMZ koja je zasticena zastitnim zidom. Web serveru je dodeljena privatna adresa 10.0.0.1. NAT uredjaj je staticki dodelio toj privatnoj adresi pravu adresu 1.2.3.4. DNS server sadrzi podatke koji vode do 1.2.3.4 tako da ceo svet moze naci pravi iP dodeljen web serveru. Zastitni zid/NAR uredjajisto ima javni iP 1.2.3.100.
Sta se desava kada web browser zeli da pristupi sadrzaju na tom web serveru? Web browser ce upitati DNS da dobije adresu web servera 1.2.3.4. Onda ce poslati web zahtev van na internet gde ce ruteri proslediti paket do 1.2.3.0 mreze. internet ruter prikacen na mrezu 1.2.3.0 ce poslati van ARP zahtev trazeci hardversku adresu za 1.2.3.4.
Ali ne postoji fizicku interfejs dodeljen 1.2.3.4. Umesto te adrese je samo logicka asocijaciaj koja govori zastitnom zidu da bilo koji paket namenjen za tu adresu treba biti poslat na web server koji se nalazi na 10.0.0.1. Zbog toga sto ne postoji fizicki interfejs, ne postoji fizicka adresa i nijedan host nece odgovoriti na ruterov ARP zahtev. Bez odgovora, ruter nemoze da posalje paket na mrezu.
Dobrodosti u ARP proxy. Ovde host (u ovom slucaju zastitni zid) odgovara na ARP zahtev sa njegovom licnom hardverskom adresom. Pretpostaqvka je da jednom kada primi okvir, znace sta da radi sa njim. Vas FreeBSD sistem ima ugraqdjeni ARP proxy (arp komanda). Recimo da je hardverska adresa zastitnog zida AA:BB:CC:11:22:33. Da bi konfigurisali tal zastinti zid da prima okvire i za njegovu iP adresu i iP adresu web servera, koristite ovu komandu kao superuser:

```
% arp -s 1.2.3.4 AA:BB:CC:11:22:33 pub
```

Da bi ste se uverili:

```
% arp -a

(1.2.3.100) at aa:bb:cc:11:22:33 on ed0 [ethernet]
(1.2.3.4.) at aa:bb:cc:11:22:33 on ed0 permanent published [ethernet]
```

pub ili "published" dodatak je taj koji poziva ARP proxy.
Ovaj clanak je pokrio vecinu uobicajenih proxy termina. U sledecem clanku, videcemo neke od ovih termina u akciji dok instaliramo i konfigurisemo jedan od proxy-ja koji se nalaze u FreeBSD kolekciji portova.

~Dalibor Gudzic@soxxx