PERTEMUAN 13
PROXY SERVER

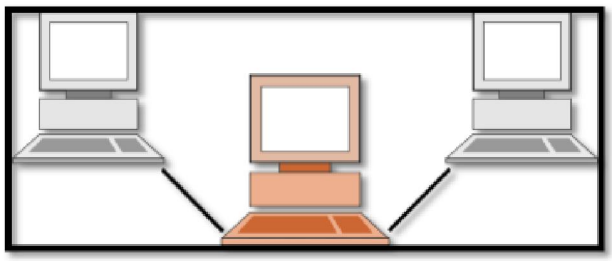
A. TUJUAN PEMBELAJARAN

Pada bab ini akan dijelaskan mengenai Proxy Server

B. URAIAN MATERI

1. Pengenalan Proxy

Dalam jaringan komputer, *server proxy* adalah *server* (sistem komputer atau aplikasi) yang berfungsi sebagai perantara permintaan dari klien untuk mencari sumber daya ke server lain. Sebuah komputer klien terhubung ke *server proxy*, meminta beberapa *service*, seperti file, koneksi, halaman web, atau sumber daya lainnya yang tersedia dari server yang berbeda. *Server proxy* mengevaluasi permintaan menurut aturan penyaringan. Sebagai contoh, menyaring *traffic* dengan alamat IP atau protokol. Jika permintaan divalidasi oleh *filter*, proxy menyediakan sumber daya dengan menghubungkan ke *server* yang relevan dan meminta layanan atas nama klien. Sebuah server proxy opsional dapat mengubah permintaan klien atau respon *server*, dan kadang-kadang mungkin melayani permintaan tanpa menghubungi *server* yang ditentukan.



Sebuah proxy server memiliki berbagai macam tujuan potensial, seperti :

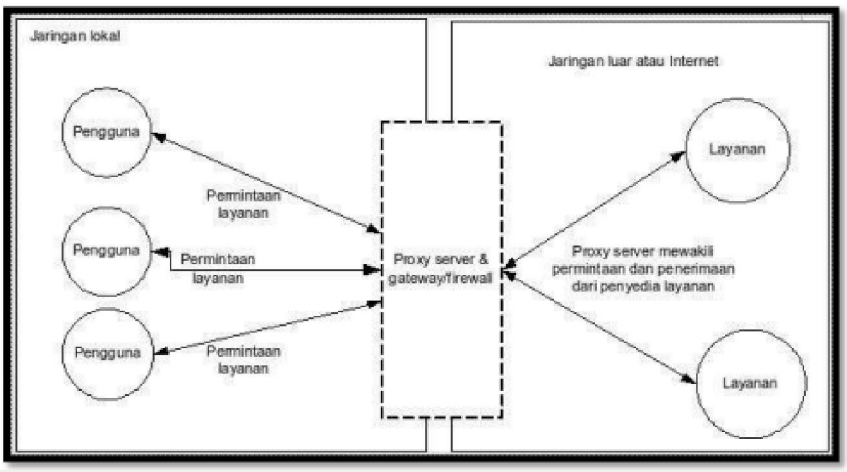
- Untuk menjaga mesin di baliknya menjadi anonim (terutama untuk keamanan).
- Untuk mempercepat akses ke sumber daya (menggunakan cache). Web proxy biasanya digunakan untuk cache halaman web dari server web.
- Untuk menerapkan kebijakan akses ke layanan jaringan atau konten, misalnya untuk memblokir situs yang tidak diinginkan.
- Untuk log/penggunaan audit, yaitu untuk menyediakan pegawai perusahaan pelaporan penggunaan internet.
- Untuk bypass keamanan/kontrol orangtua.
- Untuk memindai konten menular malware sebelum pengiriman.

- Untuk memindai konten outbound, misalnya untuk perlindungan data kebocoran.
- Untuk menghindari pembatasan regional.

2. Proxy Sebagai Gateway

Dalam suatu jaringan lokal yang terhubung ke jaringan lain atau internet, pengguna tidak langsung berhubungan dengan jaringan luar atau internet, tetapi harus melewati suatu *gateway*, yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. *Gateway* ini sangat penting, karena jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari internet, dan hal tersebut akan sulit dilakukan bial tidak ada garis batas yang jelas jaringan lokal dan internet. *Gateway* juga bertindak sebagai titik dimana sejumlah koneksi dari pengguna lokal akan terhubung kepadanya, dan suatu koneksi ke jaringan luar juga terhubung kepadanya. Dengan demikian, koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh *gateway* secara bersama-sama (*connection sharing*). Dalam hal ini, *gateway* adalah juga sebagai *proxy server*, karena menyediakan layanan sebagai perantara antara jaringan lokal dan jaringan luar atau internet.

Diagram berikut menggambarkan posisi dan fungsi dari *proxy server*, di antara pengguna dan penyedia layanan :



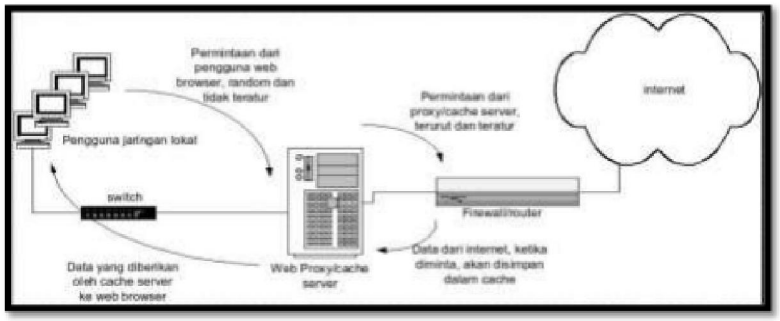
3. Jenis-jenis Proxy Server

Seperti yang telah disebutkan di atas bahwa sebuah *proxy server* memiliki berbagai macam fungsi atau tujuan potensial. Berikut ini akan dijelaskan proxy server yang berfungsi sebagai *cache proxy* (mempercepat akses ke sumber daya), *filter proxy* (memfilter akses ke situs-situs tertentu), dan juga jenis proxy yang lain.

a. Cache Proxy

Salah satu fungsi dasar dan sangat penting dari suatu proxy server adalah caching. Proxy server memiliki mekanisme penyimpanan obyek-obyek yang sudah pernah diminta dari server-server di internet, biasa disebut caching. Karena itu, proxy server yang juga melakukan proses caching juga biasa disebut cache server.

Diagram berikut menggambarkan proses dan mekanisme caching :



Gambar 4.3 Mekanisme *Caching* Pada Proxy

Sebuah *cache proxy* dapat mempercepat permintaan layanan dengan mengambil konten yang disimpan dari permintaan sebelumnya yang dibuat oleh klien yang sama atau bahkan klien lain. *Cache proxy* menyimpan salinan setempat dari sumber daya yang sering diminta, yang memungkinkan organisasi besar untuk secara signifikan mengurangi penggunaan *bandwidth*, dan meningkatkan kinerja. Sebagian besar ISP dan bisnis besar memiliki *cache proxy*. *Cache proxy* adalah jenis pertama dari *server proxy*.

b. Filter Proxy

Sebuah *content-filtering web proxy* memberikan kontrol administratif terhadap konten yang mungkin disampaikan melalui proxy. Dengan ini kita bisa membatasi akses komputer klien ke situs-situs atau konten tertentu. Hal ini umumnya digunakan baik di organisasi non-komersial maupun di organisasi komersial (terutama sekolah-sekolah) untuk memastikan bahwa penggunaan internet sesuai dengan kebijakan penggunaan yang diterima.

Beberapa metode yang umum digunakan untuk konten penyaringan meliputi: URL atau *blacklist* DNS, URL regex penyaringan, MIME penyaringan, atau kata kunci penyaringan konten.

c. Transparent Proxy

Dengan menggunakan *transparent proxy* maka kita tidak perlu menyetting proxy pada web browser klien, sehingga browser akan otomatis melewati proxy pada saat mengakses web. Jadi transparent proxy ini sangat bermanfaat untuk memastikan bahwa semua klien pasti melewati proxy.

d. Socks Proxy

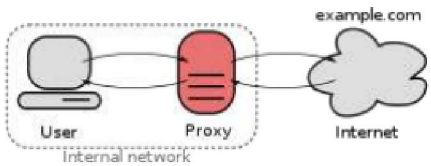
Secure Socket (SOCKS) adalah internet protokol yang rute paket jaringan antara klien dan server nya melalui proxy server. Socks5 menyediakan layanan tambahan yakni otentikasi sehingga hanya pengguna yang sah dapat mengakses server. Praktis, server SOCKS akan memperantarai koneksi TCP ke alamat IP yang berubah-ubah serta menyediakan sarana untuk paket UDP agar dapat diteruskan. SOCKS terdapat pada Layer 5 dari model OSI (lapisan perantara antara lapisan presentasi dan lapisan transport).

SOCKS adalah standar *de-facto* untuk *circuit-level gateway*. Penggunaan lain dari SOCKS adalah sebagai alat pengelakan, yang memungkinkan untuk melewati penyaringan Internet untuk mengakses konten jika diblokir oleh pemerintah, tempat kerja, sekolah dan layanan web negara tertentu.

Beberapa klien SSH mendukung *port forwarding* dinamis yang memungkinkan pengguna untuk membuat SOCKS proxy lokal. Hal ini dapat membebaskan pengguna dari keterbatasan menghubungkan hanya ke remote port yang telah ditetapkan oleh server.

e. Forward Proxy

Forward proxy merupakan proxy yang paling umum, dan ditemukan online sebagai *open proxy*. Forward proxy meneruskan (*forward*) sebuah request dari komputer pada sebuah website, dan kemudian mengakses server untuk mengambil informasi. Forward proxy memiliki kemampuan untuk mengakses lebih banyak website dibandingkan reverse proxy.



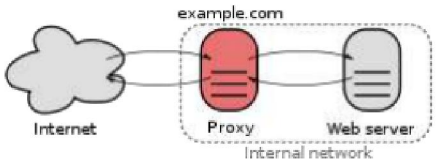
Gambar 4.4 Forward Proxy

Pada konfigurasi forward proxy, request berasal dari sebuah komputer dalam bentuk percobaan pengguna mengakses website. Request disaring melalui forward proxy dan kemudian melalui sebuah firewall. Firewall memastikan request bersifat legal atau sah, atau dari pengguna sebenarnya dan bukan dari program jahat (*malicious program*). Jika request-nya merupakan request yang benar, maka proxy akan meneruskannya (*forward*). Namun jika tidak, maka request ditolak (*request denied*).

Setelah mendapatkan informasi dari server website, maka proses akan membalik sehingga informasi akan tertuju pada komputer yang membuat request. Forward proxy dibuat untuk meneruskan *traffic* dari server ke tahapan berikutnya.

f. Reverse Proxy

Pada Reverse Proxy ini, Proxy berada di garda depan menerima permintaan HTTP Request (umumnya diport 80). Seperti Forward Proxy, salah satu tugas dari Reverse Proxy ini yaitu untuk melakukan caching halaman-halaman web yang pernah di-request sebelumnya.



Gambar 4.5 Reverse Proxy

Reverse proxy berjalan di port 80 untuk melayani request Http. Di port 80 Reverse Proxy tidak menggantikan fungsi Web Server, melainkan dia akan melanjutkan request Http tersebut ke Web Server untuk diolah. Dan apabila Web Server telah selesai mengolah permintaanya tersebut, Web Server akan mengembalikan kembali ke Reverse Proxy. Sebelum Reverse Proxy mengirim kembali request Http tersebut ke client sebagai respons (HTTP Response), Reverse Proxy akan menyimpan respon Http tersebut kedalam media penyimpanan sekunder. Sehingga, apabila ada request Http yang sama kembali, Reverse Proxy akan mengambil langsung response Http tersebut tanpa meneruskan request Http tersebut ke Web Server.

Keuntungan penerapan Reverse Proxy ini, apalagi di Web Server dengan traffic yang tinggi yakni memberikan nilai plus di sisi user-experience. Client akan mendapatkan response dari halaman yang direquest lebih cepat ketimbang merequest ke Web Server yang tidak menggunakannya. Dan keuntungan dari sisi server yaitu load server akan turun karena tugas dari Web Server akan lebih ringan dengan sedikitnya request yang diterimanya.

Sebagai catatan, Request Header yang diterima oleh Web Server adalah Request Header dari Proxy, bukan dari client. Buat yang melakukan analisa statistik web (Urchin, AwStat) maupun trace/debug log Web Server, perlu dilakukan setting tambahan di sisi Proxy dan Web Server.

4. Proxy Squid

Salah satu contoh aplikasi proxy/*cache server* adalah Squid. Squid adalah sebuah daemon yang digunakan sebagai *proxy server* dan *web cache*. Squid memiliki banyak jenis penggunaan, mulai dari mempercepat server web dengan melakukan *caching* permintaan yang berulang-ulang, *caching* DNS, *caching* situs web, dan *caching* pencarian komputer di dalam jaringan untuk sekelompok komputer yang menggunakan sumber daya jaringan yang sama, hingga pada membantu keamanan dengan cara melakukan penyaringan (*filter*) lalu lintas. Meskipun seringnya digunakan untuk protokol HTTP dan FTP, Squid juga menawarkan dukungan terbatas untuk beberapa protokol lainnya termasuk *Transport Layer Security* (TLS), *Secure Socket Layer* (SSL), *Internet Gopher*, dan HTTPS.

Squid umumnya didesain untuk berjalan di atas sistem operasi mirip UNIX, meski Squid juga bisa berjalan di atas sistem operasi Windows. Karena dirilis di bawah lisensi GNU *General Public License*, maka Squid merupakan perangkat lunak bebas.

a. ACL (Access Control List) pada Squid

Selanjutnya konfigurasi-konfigurasi lanjutan squid, selain sebagai cache server, squid yang memang bertindak sebagai "*parent*" untuk meminta object dari kliennya dapat juga dikonfigurasi untuk pengaturan hak akses lebih lanjut, untuk pertama kali yang dibicarakan adalah ACL (*Access Control List*), ACL sendiri terdiri dari beberapa tipe antara lain :

1. **src** - IP Address asal yang digunakan klien
2. **dst** - IP Address tujuan yang diminta klien
3. **myip** - IP Address local dimana klien terhubung
4. **srcdomain** - Nama domain asal klien
5. **dstdomain** - Nama domain tujuan klien
6. **srcdom_regex**- Pencarian pola secara string dari nama domain asal klien
7. **dstdom_regex** - Pencarian pola secara string dari nama domain tujuan klien
8. **time** - Waktu dinyatakan dalam hari dan jam
9. **proto** - Protokol transfer (http, ftp, gopher)
10. **method** - Metode permintaan http (get, post, connect)
11. **url_regex** - Regex yang cocok di URL secara keseluruhan
12. **cache_dir** – Mendefinisikan suatu direktori *cache*
13. **delay_pools** - Menspesifikasikan jumlah pool yang digunakan untuk membatasi jumlah bandwith dari ACL
14. **delay_class** - Menspesifikasikan kelompok dari masing-masing pool yang telah didefinisikan pada opsi delay_pools
15. **delay_parameters** - Menspesifikasikan rumus bandwidth yang akan didapatkan oleh ACL yang akan memasuki delay_pools
16. **delay_access** - Mendefinisikan ACL yang akan dimasukkan ke pool tertentu untuk mendapatkan "perlambatan" bandwidth
17. **deny_info** – Mendefinisikan output halaman HTML pada Squid untuk ACL tertentu

Berikutnya adalah *control list* yang akan digunakan untuk mengatur kontrol dari ACL, *control list* tersebut antara lain :

1. **http_access** - memperbolehkan *access* http
2. **icp_access** - memperbolehkan *peer* untuk mengirimkan icp untuk men-query object
3. **miss_access** - memperbolehkan klien meminta object yang belum ada (miss) didalam *cache*
4. **no_cache** - object yang diminta klien tidak perlu disimpan ke harddisk
5. **always_direct** - permintaan yang ditangani langsung ke server origin
6. **never direct** - permintaan yang ditangani secara tidak langsung ke server origin

5. Sistem Autentikasi pada Squid

Squid dapat memproteksi suatu jaringan dengan sistem autentikasi. Contohnya di kampus Gunadarma, setelah laptop kita terkoneksi dalam jaringan *hotspot* maka kita harus login terlebih dahulu menggunakan *email* dan *password* studentsite agar kita bisa mengakses situs yang lain. Nah, hal yang seperti ini bisa ditangani menggunakan Squid.

Squid mengenal beberapa macam skema autentikasi seperti berikut :

1. *Basic Authentication*

   Ini adalah skema autentikasi yang didukung oleh semua peramban (browser) utama dan berfungsi dengan baik di semua OS. Sayangnya skema autentikasi basic ini memiliki satu kelemahan utama, yaitu proses pengiriman data user dan password dikirim dalam format *plain text*. Jadi sangat rentan terhadap proses *sniff* atau penyadapan saat proses autentikasi berlangsung. Contoh program bantu untuk skema autentikasi basic ini adalah LDAP.

2. *Digest Authenthication*

   Skema ini lebih aman, karena pada saat autentikasi, data username dan password tidak dikirim dalam format *plain text*. Secara umum, kelebihan skema autentikasi digest dibandingkan skema autentikasi basic, yaitu lebih aman. Tapi sayangnya tidak didukung oleh beberapa browser, yakni Internet Explorer 5 & 6.

3. *NTLM Authentication*

   Dengan menggunakan skema autentikasi NTLM, semua user yang sudah login ke domain, ketika mengakses squid tidak akan diminta lagi username dan password. Ini yang kita kenal sebagai proses *Single Sign On*. Jika sudah sukses autentikasi di satu layanan, ketika ingin menggunakan layanan lain tidak perlu memasukkan login dan password lagi, proses autentikasi berlangsung secara transparan. Sayangnya skema ini hanya berfungsi dengan baik di sistem operasi Windows dan hanya mendukung browser Internet Explorer dan Firefox.

4. *Negotiate Authentication*

   Skema ini bisa dianggap sebagai *wrapper* (atau alat bantu) untuk menggunakan salah satu dari autentikasi ke Kerberos atau NTLM. Kelebihan skema ini, jauh lebih aman bila dibandingkan dengan skema autentikasi NTLM. Kelemahannya, lagi-lagi hanya berfungsi dengan baik di lingkungan OS Windows.

C. SOAL LATIHAN/TUGAS

D. DAFTAR PUSTAKA

Buku

Link and Sites: