Proxying

Why and How

Alon Altman

alon@haifux.org

Haifa Linux Club

Definition

proxy \Prox"y\, n.; pl. Proxies. The agency for another who acts through the agent; authority to act for another, esp. to vote in a legislative or corporate capacity.

— Webster's Revised Unabridged Dictionary

In computer networking...

Proxy refers to a special kind of server that functions as an intermediate link between a client application (like a web browser) and a real server. The proxy server intercepts requests for information from the real server and whenever possible, fills the request. When it is unable to do so, the request is forwarded to the real server.

— http://www.oit.ohio-state.edu/glossary/gloss3.html

HTTP Proxying

What is HTTP

- HTTP, or Hyper Text Transfer Protocol, is the standard protocol used to access web pages, and interact with web-based applications.
- It’s newest version HTTP/1.1 is defined by RFC2616.

Why Proxy?

- **Cache** — Keep frequently visited web pages available without need to re-download.
- **Maintain privacy** — Strip or manipulate identifying information before it's sent to the server.
- **Audit** — Keep a log of all web access from your network.
- **Filter** — Prevent access to certain web sites from your network.

How to proxy?

The most advanced and free proxy server is *squid*, downloadable from http://www.squid-cache.org/.

squid is straightforward to set up:

- Install the RPM and then edit the configuration file `/etc/squid/squid.conf`.
- Initialize your cache directory using `squid -z`.
- Install the *squid* service to be run at startup

The HTTP Protocol

To explain HTTP proxying we must first explain the standard HTTP protocol.
The HTTP protocol usually works as follows:

- Client connects to server port 80.
- Client sends request and headers.

```
GET /lectures/084/ HTTP/1.1
Host: www.haifux.org
User-Agent: Mozilla/5.0 (X11; U; Linux i686;
en-US; rv:1.4) Gecko/20030624
...
Referer: http://www.haifux.org/future.html
```

The HTTP Protocol (cont.)

- Server replies with status code, headers, and data.

```
HTTP/1.1 200 OK
Date: Thu, 08 Jan 2004 16:29:56 GMT
Server: Apache/1.3.28 (Unix) PHP/4.3.3
Last-Modified: Sun, 04 Jan 2004 11:39:38 GMT
...
ETag: "1d8802-1d5-3ff7fb7a"
Content-Type: text/html

<HTML>
...
```

- Both client and server close the connection.

HTTP Protocol with a proxy

The HTTP protocol via proxy works as follows:

- Client sends request and headers to proxy including full URL of the resource.
- Proxy checks its cache.
    - Cache hit
        - If needed, Proxy validates data from original host.
        - Proxy returns data to the client.
    - Cache miss — Proxy retrieves page from original host and returns data to the client.
- Proxy stores page in cache if possible, and depending on configuration.

When *not* to cache

Caching is not always a good idea:

- Dynamic Sites — You want to see current news.
- Query Results — You want current search results.
- Pages with side effects — You want your daily donation at http://www.hungersite.com/ to be counted each day.

Sometimes, cached data should not be served:

- Expiration — Even static sited are updated sometimes.
- User Request — If the user pressed *Reload*, you'd better fetch a fresh page.

Cache control

The caching behavior of proxies (and also your browser's cache) is primarily controlled in HTTP 1.1 by the `Cache-Control` header, using information from the `Date` and `Modification-Time` headers. Values of this header include:

- `max-age=`n — Return pages not older than n seconds, or cache response for up to n seconds. `max-age=0` is used by *Reload* to request revalidation of the page from the source.
- `no-cache` — Do not use a cache for satisfying the request, or do not cache the result. Used by the server to mark uncachable pages, or to strongly force reload when cache is corrupt.

Notes about caching

- A proxy server may return a stale (expired) response if it cannot contact the source of the data.
- Some proxy servers (including *squid*) cache errors in addition to normal responses.
- Web authors should take care to include the appropriate `Cache-Control` header in dynamic pages, and to use different URLs for different versions of cachable resources to allow for efficient caching.

Proxy headers

Proxy servers add (if configured to do so) special headers about the nature of the server and the client:

- `Via` is a standard header which lists the name(s) of the proxy server(s) the request has passed through.
- `X-Forwarded-For` is a header *squid* adds to identify the client originating the request.

These headers allow the web server to know about the proxy and the user behind it.

Header manipulation

Every time you request a page, your browser sends the following information in the request headers:

- `User-Agent` — Your browser and OS version.
- `Referrer` — The page that linked to the requested page, or if using IE — the last page visited in the window with the page.
- `Cookie` — Small piece of information used to uniquely track visitors.
- … and cache information as seen before.

A proxy server is in the unique position to manipulate request headers in order to protect your privacy or to skew server statistics.

Filtering and Auditing

- Proxies usually keep detailed logs of all requests.
- A proxy can also deny certain requests.
- This can be used to require authorization or to restrict users' web access.
- To make these restrictions effective, direct connection to the web should be blocked.
- *squidGuard* (from http://www.squidguard.org/) could be used to block access to questionable sites in conjunction with *squid*.
- **Warning:** sites such as Google's cache and open proxies may be used in certain occasions to bypass your proxy's filters.

Transparent Proxies

A transparent HTTP proxy is a proxy server that simulates the actual web in a form transparent to the clients.
Transparent proxies have the following advantages over standard proxies:

- No need to reconfigure each and every application with proxy information.
- Bypassing the proxy is blocked, but standard browsing still works.
- Users don't realize that there is a proxy at all.

Transparent proxy detection

Many ISPs set up transparent HTTP proxies in order to reduce their bandwidth costs by caching requests made by their clients.
To check if your ISP is running a transparent proxy, visit http://www.whatismyip.com/ and compare the result with your actual IP address from the output of `/sbin/ifconfig ppp0`.
If the IP addresses are different, you are behind a NAT or a transparent proxy.

Transparent proxy config.

- To set up a transparent proxy, all web requests must be redirected to the proxy. This is done using `iptables` in the `nat` table.
- If the server is a NAT or a firewall, first we should setup the redirection for requests originating from the internal network, using the following command:
  ```
  iptables -t nat -A PREROUTING -p
  tcp --dport 80 -i eth1 -j
  REDIRECT --to-ports 8080
  ```
  (assuming the internal network is on `eth1`)

Transparent proxy config.

- If you have web servers on your internal network routed by the firewall, you should specifically allow access to the server by issuing a command such as:
  `iptables -t nat -A PREROUTING -d 192.168.0.0/16 -j ACCEPT`
  before the former command.

Local transparent proxy

- If you wish to apply the transparent proxy to the local machine as well, use the following commands:
  ```
  iptables -t nat -N proxy
  iptables -t nat -A OUTPUT -p tcp
  --dport 80 -j proxy
  iptables -t nat -A proxy -m owner
  --owner-uid squid -j RETURN
  iptables -t nat -A proxy -p tcp
  -j REDIRECT --to-ports 8080
  ```

squid configuration

- The `REDIRECT` target in `iptables` only redirects the packets towards the proxy, but does *not* modify the request to include the entire URL.
- Therefore, the proxy must find the designated host name by looking at the `Host` header in the HTTP request.
- All browsers send the `Host` header as it is also used to allow several virtual hosts with the same IP address.
- To trigger this behavior, the squid configuration option `httpd_accel_uses_host_header` must be set to *`on`*.

Transparent proxies and DNS

- The behavior of a transparent proxy requires it to perform a DNS lookup for each missed request, which may slow down the transaction.
- It is suggested to run a caching DNS server for the network as well as the proxy server.
- Due to the redirection, the identity of actual target host is lost.
- Thus the client cannot override private DNS settings for testing purposes.

Summary

- HTTP proxies allow for caching, privacy, auditing and filtering of WWW requests.
- Cache control mechanisms determine the behavior of caches and proxies.
- Transparent proxies allow proxying without client configuration.