Assignment Three

Anonymizing Distributed Proxy

Set: *16th of May 2011*
Due: *3rd of June 2011 @ 23:55 CEST*

Synopsis:
Extend your proxy to form part of a proxy swarm that will enable the browsers IP address to be kept anonymous.

Introduction

This is the third of four assignments in the *Datanet* course. The four assignments are practical in nature, and you will gradually build a distributed anonymising web proxy. Your proxy will be part of a real distributed system running on the Internet, with peers contributed by your fellow students. This third assignment is about participating in a distributed system, and examining the effects of this.

For this assignment, you must extend the proxy from the previous assignment (or write a new proxy from scratch using any HTTP aware libraries that you wish). The extended proxy (the peer) will need to register with a central proxy tracker and use the list of peers to forward requests.

IMPORTANT: Security Note

As you will be transmitting data over an untrusted network, it is important that you do not perform any sensitive browsing while using the anonymous network: *do not log in to ANY website*.

Since others may be making requests for arbitrary websites through your peer, the tracker distributes a whitelist of sites that you should allow through your proxy. It is therefore important that you have implemented this whitelisting correctly before registering with the central server. If you fail to properly implement the whitelist, there is a chance that an evil malicious person will use the proxy for nefarious purposes that might get you in trouble. A nefarious request might well be traced back to *your* IP address, and thus to you. You might find that it is easier to limit the websites that your proxy will contact than deal with any potential fallout of an evil person abusing your proxy.

Implementation

In this assignment you will write a peer for an anonymising proxy network. You may choose to extend your work from Assignment 1 or Assignment 2. Your peer will be interacting with other peers as well as a *tracker*.

Tracker

In order to participate in the network your peer must periodically contact and register with a central server, the *tracker*. The act of registering will include your peer in the list of peers distributed by the *tracker*. This list will be sent to your peer as part of the registration process. The interactions that you will need to perform with the *tracker* are not described in this text. Instead you must refer to the documentation on the *tracker*'s home page: `http://datanet2011tracker.appspot.com/`

The tracker is able to provide your peer with the peer list in a number of formats. You can use the format most convenient for your implementation language. The available formats are: *YAML*, *JSON*, *Bencode*, *XML*, and *plain text*. See the *tracker*'s homepage for concrete examples of the data formats. The tracker also supplies the whitelist, as well as other information, as part of the registration process.

Whitelist

Before you register your peer with the tracker, you should ensure that you test the whitelist functionality of your proxy. The format and function of the whitelist is described on the following page: `http://datanet2011tracker.appspot.com/doc/whitelist`. If a request does not match the whitelist, your peer must send a response with the status code "403" and an appropriate reason phrase, such as "Denied By Proxy". If you need a whitelist for testing, you can extract it from one of the peer lists. The plain text peer list may be a good choice: `http://datanet2011tracker.appspot.com/peers.txt`.

Max-Forwards Header

Once you have the whitelist functionality working, you will need to extend your peer to forward requests. Requests will either need to be forwarded to another proxy **or** the destination of the request. To ensure that requests will eventually exit the proxy network, requests will use the `Max-Forwards` header. `Max-Forwards` is described in in Section 14.31 of RFC2616, which ends by stating that:

> `The Max-Forwards header field MAY be ignored for all other methods defined by this specification and for any extension methods for which it is not explicitly referred to as part of that method definition.`

For the purpose of this assignment we will add the following to the specification of `Max-Forwards`:

> `The Max-Forwards header MUST be taken into account for all request methods which the proxy is willing to forward. The Max-Forwards header MAY be removed`

> when the request is sent to the final destination but MUST NOT be removed otherwise.
>
> If there is no Max-Forwards header in a request, a Max-Forwards header MUST be added. The proxy MUST use a maximum of value of 5 for the Max-Forwards header but SHOULD use a value of 3 under normal circumstances. If a Max-Forwards header is received with a value that is larger than 5, the proxy MUST set the Max-Forwards value to a value no larger than 5. The proxy MAY set Max-Forwards to a value that is less than 5.

The semantics of `Max-Forwards` is otherwise as stated in RFC2616. The use of the `Max-Forwards` header ensures that a request will not be forwarded without bound through the proxy network. The use of the recommended initial value of 3 will ensure that requests will be forwarded through five peers before it exits the network.

Via Header

Your host should be able to update and send correct `Via` headers, as specified in Section 14.45 of RFC2616. However, as this notionally an anonymising proxy network, the requirements have been relaxed from the RFC's specification which states that:

> The Via general-header field MUST be used by gateways and proxies to indicate the intermediate protocols and recipients between the user agent and the server on requests, and between the origin server and the client on responses.

To the following:

> The Via general-header field MUST be used by the proxy if the incoming request already contains a Via header field. The proxy SHOULD NOT add the Via header field if it is not already present.
>
> The proxy SHOULD use the 'host:port' format of the received-by token and SHOULD use the peers IP address for the host portion.
>
> The proxy SHOULD remove the Via header before the request is passed onto destination host. However, the proxy MUST add the Via header to the response if the request contained a Via header.

This allows requests to be traced through the network, if the requester has set a (possibly empty) `Via` header, but keeps the requests route of the request anonymous otherwise. We recommend that the IP address of the peer is used in the received-by token such that the route of the request can be mapped using data from a geolocation database.

Forwarding

When a peer forwards a request to another peer in the proxy network, it should pick a peer from the list of peers supplied by the tracker. It is almost certain that not all peers in the supplied peer list will actually be active at any given moment and your peer must be able to deal with such non-responding peers. You may want to remove non-responding peers from your local peer lists to avoid trying to contact them in the future. Keep in mind that the tracker will probably send the addresses of these non-responding peers each time you request an updated peer list and it is up to you how you deal with this. However, the peers will eventually disappear from the tracker's peer lists, but only when a peer has not contacted the tracker for some time.

In order for your peer to participate in the proxy network it must have an Internet routable IP address and an open port. This is not possible on the KU network and might be difficult (or at least undesirable) to set up even on your home network. The network is also more interesting if peers are active for a reasonably long time, which will not be the case if you just run a peer on your laptop. We recommend that you use Amazon's EC2 service as a convenient location for you to run your peer. Amazon EC2 is a paid for service, but we have USD100 vouchers that you can request so that you can run your peer on EC2 without being out-of-pocket. Please email `cljac@diku.dk` to obtain a voucher.

If you for some reason are unable to have your peer visible on the network and unable to use EC2, then you will still need to test that your peer can forward incoming requests. In this case you **must** start at least two instances of your peer, where the first instance forwards the request to the second instance, which then forwards the request to a peer selected from the tracker's peer list. For information about how to obtain the peer list without registering your tracker, see the tracker's documentation.

Experiment

You must also conduct an experiment that uses the proxy network, where you investigate the latency and bandwidth of the proxy network. You should use your timings from the previous assignment for comparison. You should conduct *at least* the following experiments:

- Start a peer and make your browser use the peer as its proxy. Load some of the same pages as you did in Assignment 2 and measure the latency. Beware that the nature of the proxy network will make your timings vary and you will have to run the test a few times to get an average.

  You *ma*y want to force your proxy to set the `Via` header, for requests that come from your browser, so that you can correlate the route with your timings.

- Start a peer and make your browser use the peer as a its proxy. Download a large file (10-100MB) and measure the bandwidth. You will have to run the test a few times to get an average.

Your Report

Your report *must* be written in ACM format. An ACM template for LaTeXand MS-Word is available for download via Absalon.

Your reports should contain:

- An abstract describing the contents of your report
- A description of security measures that you have implemented
- A description of your strategy for handling non-responsive hosts
- A description of any flaws or problems with the protocol
- A description of your setup, i.e. do you have open ports etc.
- The responsiveness and bandwidth measures that you obtained from your experiment

Deliverables for This Assignment

You are encouraged to work in informal groups for this assignment, for the purpose of discussing implementation details and limitations. We strongly encouraged you to come to the exercise, where we will use time to discuss the design, implementation etc. The implementation and report that you hand in must however be **your own individual work**. You should submit the following items:

- A single PDF file, A4 size, no more than 3 pages, in ACM format, describing each item from report section above
- A single ZIP/tbz2 file with all code relevant to the implementation

Handing In Your Assignment

You will be handing this assignment in using Absalon. Try not to hand in your files at the very last minute, in case SkyNet becomes self-aware and decides to send a Terminator to kill your Internet connection at the exact moment you are trying to submit. **Do not email us your assignments unless we expressly ask you to do so**.

Assessment

You will get written qualitative feedback, but no grade for the assignment. Your assignments will be evaluated together with your exam and produce a final grade for the course.