Datanet

Assignment Three

Anonymizing Distributed Proxy

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

Synopsis:
Extend your proxy to use a proxy swarm to keep the browsing IP anonymous

Introduction

This is the third of four assignments in the *Datanet* course. The four assignments are practical assignments, where you will gradually build a distributed anonymizing proxy where your implementation and that of your fellow students are all parts of the same distributed system.

This third assignment is about participating in a distributed system, and examining the effects of this.

You must extend your proxy from the previous assignment to register with a central proxy list and use this central proxy list 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 using your proxy, it is also important that you make sure that the white list is implemented correctly before registering with the central server. If you fail to do so, there is a chance that a malicious person will use the proxy for evil intent, which will lead back to your IP, and thus to you. Rather than attempt to convince a judge that you are innocent, try to be proactive and limit the available set of websites.

Your implementation

You must extend your proxy server to become part of a distributed anonymizing proxy network.

To participate in the network, your server must periodically register with a central server, aka the tracker. The registration tells the remote server which IP and port your proxy has, and gives your server a list of known proxies as well as a list of approved domain names.

The list of known proxies and allowed domains can be accessed through the tracker, which is located at http://datanet2011tracker.appspot.com/

You can access the data in a number of formats, such as xml, yaml, json, etc. An example data extract looks like this (in yaml format):

```
options: {expire: 86400, min_wait: 600}
peers:
- {ip: 130.225.212.4, last_reqistered:
    !!timestamp ’2011-03-28 14:15:04.846028’,
    port: 0, super_peer: false}
- {ip: 87.72.201.144, last_reqistered:
    !!timestamp ’2011-03-28 10:22:58.757724’,
    port: 0, super_peer: false}
- {ip: 87.72.201.142, last_reqistered:
    !!timestamp ’2011-03-27 14:31:39.003685’,
    port: 9000, super_peer: true}
- {ip: 87.72.201.143, last_reqistered:
    !!timestamp ’2011-03-27 14:30:59.218292’,
    port: 12000, super_peer: true}
- {ip: 87.72.201.145, last_reqistered:
    !!timestamp ’2011-03-27 14:30:08.771984’,
    port: 0, super_peer: false}
whitelist: [www.google.com, www.google.dk, www.wikipedia.org,
    www.wikipedia.dk, www.engadget.com, www.gizmodo.com,
    slashdot.org, newz.dk]
```

Before registering your server, make sure that you test the white list functionality on your proxy to avoid situations where the whitelist does not work. The whitelist simply states a number of domains, and your proxy must send a response with status code ″403 - Denied By Proxy″ if the ″Host″ header contains a value that is not in the list of approved domains.

Once you have the whitelist functionality working, you must extend your proxy to forward the requests. To do this, we have invented a special header called ″Datanet-Forward-Count″. The scheme is that the initial request to a proxy will not have this header, so the proxy sets this header with a value that indicates the desired number number of hops, e.g. ″3″. When a request is received where this header is set, the number decrements. If the value is larger than zero, the request is forwarded to another proxy, obtained by random selection from the list. If the value is zero the proxy must forward the request to the actual host. This ensures that a request cannot run forever, and allows you to also experiment with the number of hops. To ensure some stability in the system, your proxy may also enforce a maximum value of ″5″ for the ″Datanet-Forward-Count″ header.

As it is highly likely that some of the proxies in the list do not respond to requests, your implementation *MUST* be able to deal with non-responding hosts. If you encounter a host that does not respond, simply remove if from the local list of known proxies. The broken host may re-enter the list when you sync with the tracker, but it will eventually disappear from the list when the tracker decides it has not reported for some period.

For this to work you need a network that allows you to open ports externally. The assignment is more fun if many proxies are on the network, but if you cannot change the network configuration to allow external connections, you can limit yourself to only obtaining the list from the tracker, and not register your proxy. In this case, you *MUST* start at least two instances of your proxy and initially forward the request to another instance.

Your Experiment

You must also make an experiment that uses the proxy network.

You should use your timings from the previous assignment for comparison.

While using your proxy, which is a part of the distributed network, you must try to load the same pages as you did in assignment two.

This time the randomization should make the timings much less reliable, so you will have to run the test a few times to get an average.

You must also download a large file (appx. 100mb) and measure the bandwidth. Again, the random nature of the network means that you will have to run the test a few times and calculate the 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 the rest of the Datanet students stage a DDoS attack on Absalon 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.