Linköping university
14 February 2014

# Course TDTS04 Computer Networks and Distributed systems

# Lab assignment 02: NetNinny

# User manual

**Maija Vilkina / maivi782**
**Olle Kvarnström / ollkv086**

This manual assumes that the user is familiar with basic terminal commands in Linux / Mac and that functioning compilation software for C++ is installed in the user's machine. To download and install a compiler for C++, please visit this link and follow the instructions for your operating system.

## How to compile the proxy

1. Extract the NetNinny.tar archive in a folder of your choice.
2. Run command make in the directory.
3. An executable called proxy will appear in the directory after successful compile.

## How to start the proxy

1. Write ./proxy [port number] in the terminal directory, after navigating to the directory where the compiled proxy executable is located.
2. Enable web proxy in your system settings. For example, on Mac you need to go to: System Settings -> Network -> Advanced -> Proxies. There, check Web proxy and enter localhost in the field Web Proxy Server and your chosen port number in the port field.
3. After the proxy is enabled in the system settings, it will filter all web traffic regardless of browser used.

## Features supported

- The proxy supports both HTTP/1.0 and HTTP/1.1
- It searches both urls requested and text content returned for forbidden words according to specification.
- It imposes no limit on received user data.
- Compatible with all major browsers without the requirement to tweak any advanced feature.
- User can select a port by entering it after the command to start up proxy: ./proxy [port].

## Requirements satisfied

1. **The proxy should support both HTTP/1.0 and HTTP/1.1.**

   Our proxy supports both HTTP/1.0 and HTTP/1.1. The function where this is relevant is the function that changes the connection type to closed when proxy receives a request from client.

   Relevant function:

   - HTTPProxy::removeKeepAlive line 248 in HTTPProxy.cc

2. **Handles simple HTTP GET interactions between client and server**

   Default behavior. Proxy can also handle multiple simultaneous connections via fork().

   Relevant functions:

   - HTTPProxy::run line 33 in HTTPProxy.cc

3. **Blocks requests for undesirable URLs, using HTTP redirection to display this error page instead**
4. **Detects inappropriate content bytes within a Web page before it is returned to the user, and redirecting to this error page**

   Our proxy checks for forbidden words in the required urls when the proxy receives a request from the client, as well in the returned content of type text.

   Relevant functions:

   - HTTPProxy::hasBlockedContents line 207 in HTTPProxy.cc
   - HTTPProxy::redirectToError1 line 141 in HTTPProxy.cc
   - HTTPProxy::redirectToError2 line 146 in HTTPProxy.cc

5. **Imposes no limit on the size of the transferred HTTP data**

   We are continuously forwarding the received data to the client after checking for the forbidden words in the content, if the content is of type text.

   Relevant functions:

   - HTTPProxy::handleRequest line 63 and onward in HTTPProxy.cc

6. **Is compatible with all major browsers (e.g. Internet Explorer, Mozilla Firefox, Google Chrome, etc.) without the requirement to tweak any advanced feature**

   Proxy is started by entering ./proxy [port] in the terminal window of the operating system.

7. **Allows the user to select the proxy port (i.e. the port number should not be hard coded)**

   User chooses a port on which the proxy will run by entering desired port number when starting the proxy.

8. **Is smart in selection of what HTTP content should be searched for the forbidden keywords. For example, you probably agree that it is not wise to search inside compressed or other non-text-based HTTP content such as graphic files, etc.**

Our proxy checks for which data type is received from server and whether it is compressed or not. Only content that contains “Content-Type: text/” is searched for forbidden words.

Relevant functions:

- HTTPProxy::handleRequest line 63 and onward in HTTPProxy.cc, especially line 99 and onward for text content and line 117 and onward for other types of content
- HTTPProxy::contentIsText line 229 in HTTPProxy.cc

9. **(Optional) Supporting file upload using the POST method**
   Our proxy does not support this method.

## Testing of the proxy

Proxy was tested with both the test pages provided in the lab assignment as well as a number of external pages like wikipedia.org and aftonbladet.se

**Following results were observed:**

- Example sites provided in the lab assignment were loading fine and according to specification.

The following was output in proxy terminal window while loading the four test pages:
Received connection to www.ida.liu.se
Received connection to www.ida.liu.se
Redirected client due to bad request
Received connection to www.ida.liu.se
Redirected client due to bad content
Received connection to www.ida.liu.se
Redirected client due to bad content
Received connection to www.ida.liu.se
Respective packet trace files are included in the NetNinny.tar.gz file submitted for evaluation.

- The proxy is really slow to load content intensive sites like aftonbladet.se, it took several minutes for the content to be displayed, which initially led us to think that it was not working at all.
- It is fun to see in the proxy status window how many connections are created by a single website visit.
- Forbidden words and urls are not blocked if TLS encryption is enabled in the browser or if the site visited forces TLS encryption.