

Proxy Server  
Steven Diviney  
08462267  
CS3031

## 1 Achievements

Blacklisting URLs  
Caching pages to memory  
Request redirection  
Caching DNS lookups

## 2 Proxy

The web-proxy was implemented in python. The client connects to the proxy which then intercepts all of the clients HTTP traffic. Each connection by the client runs in it's own thread. The thread terminates when the client request is completed.

Each request by the client is parsed to identify the destination. If the host is blacklisted, the thread terminates. A DNS lookup is then performed if the host in the URL has never been seen before. The resulting ip address and port are then cached into memory. A hash-map is used to store the DNS query results and is indexed by the host portion of the URL. The proxy then connects to the destination and forwards the original client request as it sees it. The proxy then waits for a response using `select()`. This is a lot more efficient than polling each socket for a response. When the proxy starts to receive data from the host, it immediately sends it on to the client. The incoming data is also stored into a list. When the host stops sending data the list is stored into a hash-map, indexed by the URL. This cache is then written out to disk to preserve it between executions. However, the hash-keys are not preserved, so caching only works for a single execution.

## 3 GUI

The GUI was not implemented because python requires a "while 1" loop to run a GUI. The threads to handle connections are also created in a while one loop and I was unable to do the two at once.