HTTP Traffic Analysis

1. The purpose of this assignment is to sensitize students on the use of web-based tools to analyze the use of HTTP server resources and proxies.
2. The evaluation of two freeware tools is proposed: "Webalizer" and "AWStats". Both should allow statistical analysis and the production of HTTP service usage reports, from the server's log files.
3. For analysis purposes it is suggested to use "Apache" and "Squid" as HTTP server and HTTP proxy, respectively.

Apache

Set up an Apache server to provide two Web sites with dynamic content update. Eventually, if you have this background knowledge, can create two "virtual hosts" (VH) to simulate hits to different sites (see code sample in the appendix), containing one site a list of processes and system load, and the other site the list of TCP/IP sessions established and the ARP table.

The Apache log files must be in /var/log/httpd, and separate logging must be done in different log files, in case we have more than one VH.

Proxy SQUID

Configure the SQUID proxy server on one of the available servers. Using an HTTP client configure periodical access via the proxy to the site(s) who have installed in the Apache. Check that the SQUID logs record the accesses made.

The SQUID log files should be in /var/log/squid.

For simplicity, it is advisable to change the "**`logformat`**" option in the SQUID configuration file, to generate log records in a format similar to the Apache server.

Alternatively, you can maintain the native format of the SQUID log records and use the **`squid2commom.pl`** program to convert the SQUID proxy logs to data in the Apache server format.

When you run the program **`squid2common.pl`** two files are created: **`cache.convert`** and **`proxy.convert`**. The first contains the logs to the cache access and the second logs of proxy access.

Webalizer

Install the tool Webalizer. Webalizer settings are in a single configuration file called /etc/webalizer.conf. You can also put a configuration file in each directory where you run the program and thus webalizer analyze different log files: each site with your log file.

AWStats

Install the tool AWStats. The settings of AWStats are in a configuration file located in the directory /etc/awstats/. In this directory, you can add a configuration file for each server you want to analyze.

Results

Present the detailed comparative analysis of the two tools in functional terms and where possible in performance.

There are other tools for this type of log file analysis, such as "W3Perl", so a comparative analysis of this tool with those previously evaluated should be done.

Crontab

These analysis tools should be run periodically at least once a day in order to gather statistical data and present them in a friendly way.

To run them periodically you can use the crond, which reads the configuration file /etc/crontab where is schedule the time to run.

These should always be executed before the log rotation, which is also scheduled in /etc/crontab or using logrotate tool.

Recommended Reading:

http://httpd.apache.org
http://www.squid-cache.org
http://www.webalizer.org
http://www.awstats.org
http://www.w3perl.com

Prepare a short report with the answers to the questions above and send it by e-mail until the day before the next practical lesson, to <joao.neves at fe.up.pt>.

Appendix

Example code for *Site* 1:

```
<html>
<meta http-equiv="refresh" content="<?php
      $hora = date("H");
      if ($hora > 9 && $hora < 12)
            echo rand(1, 5);
      elseif ($hora >= 12 && $hora < 14)
            echo rand(20, 240);
      elseif ($hora >= 14 && $hora < 18)
            echo rand(1, 4);
      elseif ($hora >= 18 && $hora < 20)
            echo rand(60, 480);
      else
            echo rand(2000, 4000);
?>">

<body>
<?php
      $f = fopen("/proc/net/netstat", 'r');
      while (!feof($f)) {
            echo "<pre>" . fgets($f) . "</pre>";
      }
      fclose($f);
?>
</body>
</html>
```

Example code for *Site* 2:

```
<html>
<meta http-equiv="refresh" content="<?php
      $hora=date("H");
      if($hora>8 && $hora<=18) {
            echo rand(1,60);
      } else if($hora>18 && $hora<=23) {
            echo rand(60,120);
      } else if($hora>23 && $hora<=8) {
            echo rand(500,900);
      }
?>">

<body>
<?php
      $output = shell_exec('netstat -n');
      echo "<pre>".$output."</pre>";
      $output = shell_exec('cat /proc/net/arp');
      echo "<pre>".$output."</pre>";
?>
</body>
</html>
```