# SENG2040 Network Application Development

## Services and Logging

This assignment may be done Individually or in Pairs. Please note that in either case, you will be asked to demonstrate your logger and test client running between two devices (such as across a ZeroTier VPN).

### Background

We have been discussing many aspects of network applications and architectures so far in this course. One thing that may have become apparent to you is that it can be very critical to be able to log detailed access and diagnostic data.

## Submission Guidelines

Please read and understand the submission guidelines below. Failure to correctly submit any required component will result in 20% penalty per missing or incorrect component.

### Paper Submission

There is no paper submission required for this assignment.

### Electronic Submission

Exactly four items will be submitted electronically. Only one of your team needs to make a submission.

* Upload two zips file of your CLEANED source code (one each for client and server).
* Upload a git bundle of each program (client and server), by running (in the relevant project directory)
  + git bundle <<your\_names>><<client | server>>.bundle --all

All files should be named with your user IDs (example jsmith1234\_bjones5678\_client.zip). Complete file naming and upload guidelines are provided near the end of this document.

Please note – you must ensure that all files required to recreate your assignment are included in your submission.

### The Task

Design and develop a logging service and a test client that could be used for various future assignments and projects.

### Requirements

It is up to you to determine the information that you will log and the format you intend to use. You must meet the following requirements:

* You may not use any prepared logging framework (such as log4j)
* The service will be network based. It will listen for logging messages on either TCP or UDP (your choice).
  + The service does NOT need to be configured to run as a windows service
* The service may be developed in any language you wish – suggestions include Java, Python, or Go (other languages may be acceptable – you must ask before using them though.)
* The service will store logging information in plain text files.
* The service does not need a UI (command line is OK)
* You must use a config file or command line arguments – no-hard coded paths or addresses.
* You will have to research what features a good logging service should support.
* You will create a client tool to test your service.
  + This must be developed in a language different than the logging service or a penalty will apply.
  + This tool should allow for manual testing of your service.
  + This tool should be able to run a thorough set of automated tests on your logging service.
* You must adhere to SET standards
* Demonstration is REQUIRED or you will not receive a grade

Please note that it is up to you to develop a GOOD logging solution – appropriate service information and configurability. You must do some additional research to do so.

### Resources

Linux and Unix have a logging service called syslog. You are not duplicating syslog, but you can definitely learn a lot from it. <https://stackify.com/syslog-101/>

Tips on logging microservices: <https://logz.io/blog/logging-microservices/>

## Penalties / Caps

The following penalties / caps will apply:

* Test client written in same language as service – 20% deduction
* Test client must run on same machine as logging service – 40% deduction
* Not possible to run multiple logging clients simultaneously – 20% deduction

## Notes

Logging service tips:

* Your logs must be stored in a single, plain text file.
* Your log message format MUST be configurable.
* Your logging service must be designed in a way that multiple simultaneous clients can be used and identified.
* The content of your stored logs is up to you; you must do some research to determine what should be there.
* Reading, filtering, or displaying your logs is completely out of scope.
  + Writing a log viewing tool is a waste of your time.
  + Use a tool such as lnav (<http://lnav.org/>) to view and filter logs.
  + Your log file format should support using common tools (Notepad++, Vim, lnav, etc.)
* Your logging service must prevent abuse from mis-configured or overly noisy logging clients (this is called rate limiting)
  + Consider the case where a logging client may accidentally be sending debugging messages

Test client tips:

* Your test client must allow you to send manually configured log messages of any type that your service supports
* You should be able to run two or more instances of your test client to generate logs
* Your test client must include the ability to test and demonstrate your service abuse prevention
* Your test client must include the ability to automatically test all log formats and message types that your service supports
* Your test client is not an interface for your logging service, only a test client. Your logging service operates independent of your test client.

## Testing

Note that it is not enough to barely meet the proposed requirements and test only the happy path. It is expected that you test your solution for normal functionality as well as error cases.

## Evaluation

You will be graded on the following basis:

* Requirements Fully and Properly Implemented
* Configurability and Detail
* Service abuse protection
* Usability
* Testing / Bug Free Operation / Adherence to SET standards
* Meaningful git commit history (ensure all of YOUR source files are included) \*

\* As discussed in class and / or labs

A letter grade will be assigned for each of these dimensions. An overall letter grade will be assigned accordingly.

The following is a guideline for the letter grade evaluation that will be used.

|  |  |  |
| --- | --- | --- |
| Letter Grade | Achievement | Description |
| A+ | Exceptional | Thorough demonstration of concepts and/or techniques and exceptional skill or great originality in the use of those concepts/techniques in satisfying requirements. |
| A | Excellent | Thorough demonstration of concepts and/or techniques together with a high degree of skill and/or some elements of originality in satisfying requirements. |
| B+ | Very Good | Thorough demonstration of concepts and/or techniques together with a fairly high degree of skill in the use of those concepts/techniques in satisfying requirements. |
| B | Good | Good level of demonstration of concepts and/or techniques together with considerable skill in using them to satisfy requirements. |
| C+ | Competent | Acceptable demonstration of concepts and/or techniques together with considerable skill in using them to satisfy requirements. |
| C | Adequate | Acceptable demonstration of concepts and/or techniques together with some skill in using them to satisfy requirements. |
| D | Barely Passing | Minimum demonstration of concepts and/or techniques needed to satisfy requirements. |
| F | Unacceptable |  |

# Electronic Submission and File Naming

Often, multiple groups will name their assignment files assignment1.zip (or some other common combination) which results in confusion and complications when downloading and unzipping submitted electronic files.

Please follow the following guidelines for all submissions this term. Failure to follow the electronic submission guidelines may result in a 10% penalty per infraction.

As long as other file naming conventions are followed, you may choose to append an assignment identifier (i.e. “\_a1” or something else as appropriate) as the last element before the file extension.

## Document Naming and Submission

Unless otherwise requested, you should only ever submit a single document containing the complete document as specified. This document must be:

* A professionally formatted document adhering to the SET report standard
* A single PDF file, not included in a zip

Unless indicated otherwise, do not submit the assignment specification as a part of your document.

Documents should be named with the usernames of all team members as follows:

* Individual assignments: username.pdf (i.e. jsmith.pdf)
* When working alone: username\_alone.pdf (i.e. jsmith\_alone.pdf)
* When working with a partner: username1\_username2.pdf (i.e. jsmith\_bjones.pdf)
* When working as a team: username\_team.pdf (i.e. jsmith\_team.pdf – but be certain to include all team members on report title page)

## Zipped Source Code

Unless otherwise requested, you should only ever submit a single zip file containing the source code as specified. Your source code must always be contained within a top-level folder named correctly, so that when it is extracted, it does not lose naming information.

The top-level directory should be named with the usernames of all team members as follows:

* Individual assignments: username\_src.zip (i.e. jsmith\_src.zip)
* When working alone: username\_src\_alone.zip (i.e. jsmith\_src\_alone.zip)
* When working with a partner: username1\_username2\_src.zip (i.e. jsmith\_bjones\_src.zip)
* When working as a team: username\_src\_team.zip (i.e. jsmith\_src\_team.zip – but be certain to include all team members in every code header)

Place your cleaned project file(s) and / or folder(s) inside this correctly named top level directory before zipping your files.