19 DOS detection*

Denial of Service is a well known attack vector aimed at using up the available resources of servers in order to deny non-malicious users the use of the

service. There are many variations of the basic attack vector depending on the services provided by the server. In part, the attacks have become more sophisticated due to protection layers employed by service providers. In this project, students will dive into the basics of preventing denial of service attacks. While the implementation does not follow the state of the art protection mechanisms, there is an educational benefit to learn about the basics of packet filtering and traffic monitoring.

Java is a virtualized language that seldom lacks low level access to the functionalities provided by the kernel. To overcome this, students will execute an external program such as tcpdump (tested on Linux). The external process can be executed via the ProcessBuilder, and an InputStream can be attached to it to read the output of the process.

An example of a tcpdump monitoring traffic on all host interfaces on port 8080 is given:

```
sudo tcpdump -i any port 8080 and '(tcp[tcpflags]&(tcp-syn|tcp-ack))!=0'
```

The process will output a line for every TCP packet received on the monitored port. For your java application, the external process should be treated as the producer, and other threads should consume the output to perform basic traffic monitoring.

The solution is aimed at detecting anomalies in the traffic based on it's source as well as cumulatively. To achieve this, traffic is monitored depending on the source IP address. For each source, the model builds the expected pattern of traffic (averages, moving averages, and other statistical indicators). In case the currently observed traffic is increased in a statistically significant way, handling those requests/traffic should be given low priority to avoid interrupting the operation of "honest" clients.

The project does not specifically force the use of one indicator over the other. However, to test the solution a basic web server or SocketServer must be implemented, and traffic generated using tools such as ApacheBenchmark. The use of volatility indicators is encouraged (i.e. Bollinger bands), often used in time-series technical analysis and trading.