Operating Systems with C/C++
Semester 1, 2012
Eike Ritter

The University of Birmingham
30 October, 2012

Assessed Exercise 3

Deadline: Fri 16 November, 12noon

The Task

The purpose of this exercise is to write part of firewall in user space. The firewall should implement some rate limitations. More precisely, for a given port, the firewall should drop all incoming connections on this port if more than a given number of connections arrive in a second. After a specified interval the firewall should accept again connections on this port, subject again to the rate limit.

Your program should work as follows:

- You should use the `netfilter-queue`-library on the virtual machine provided. This library is not installed by default–install it with the command

  ```
  sudo aptitude install libnetfilter-queue-dev
  ```

  You need internet access to do this. You need to do this only once.

- The netfilter-queue-library lets you handle each packet which has been re-directed by the kernel firewall code in user space. To set this up, you need to initialise the kernel firewall with

  ```
  sudo iptables-restore ⟨directory⟩/iptables.conf
  ```

  where the file `iptables.conf` is available from the webpages. The number after `--dport` is the port number for which the kernel-level firewall passes all packets arriving on this port to user space.

- An example user space program which demonstrates the use of the netfilter-queue library can be found on the webpage.

- For implementing the specified interval during which the firewall should not accept any connections you should use a timer.

- You should also write a wrapper program which sets the kernel-level firewall and starts the user-space program. Killing this program should kill the user program and reset the firewall. The configuration parameters should be arguments to this program.

- Use of the netfilter-queue library requires root privileges. Hence only root will be able to execute these programs successfully. You should abort immediately with an error if a non-root user tries to execute these programs. Use the system call `geteuid` for this check. The root user has always a user ID of 0.

Marking Scheme

Please use the School submission system for submitting your code. Please submit only the source files you have written yourself. We will compile and run your code on the Linux machines and mark it accordingly. Please in particular note that we will use the compiler option introduced in the lecture and will deduct 6 marks immediately if there is any compiler error or warning.

We will award marks as follows:

- 5 marks for correct usage of the netfilter_queue library.
- 5 marks for correct handling of the rate limitations.
- 5 marks for correctly accepting connections again after timeout.
- 5 marks for the correct functioning of the wrapper program.