Workshop in Information Security – HW 2

As instructed in the assignment guidelines, I used the module I wrote in ex1 and added the option to send and receive data from and to the kernel with a sysfs device.

In addition, I wrote a user-space program that can get and print data regarding the accepted and dropped packets in our lab's network, and also nullify it if the user passes 0 as an argument.

main.c:
The sysfs file is opened according to the expected location.
If "0" is entered as an argument, we write it to the sysfs "file". -> result will be explained in the next section.
We then read the number of accepted packets, followed by the number of dropped packets from the sysfs "file".
We then print the information in the requested format.
Please note that:

- I did not think it important to explain the parsing (reading into a buffer and then using the atoi function).
- The two consecutive reads – number of accepted packets and then number of dropped packets - did not work for me unless I closed and opened the file in between the reads.

hw2secws.c:
In addition to the functionality of ex1:

Initialization:

- The sysfs device is registered to a major number using register_chrdev.
- We create a class under /sys/class using class_create.
- We create a device which links our device to the class using device_create.
- We link the new show/store attributes (explained in the next sections) to the sysfs device using device_create_file.

Show/display:
This function allows the sysfs device to "show" data to the user-space program:
When the user-space program reads data from the sysfs device file, it gets, alternately, the number of accepted/dropped packets according to a parity bit.

Store/modify:
This function allows the sysfs device to "store" data sent by the user-space program:
When the user-space program writes 0 (other values change nothing – and in main we verify that the user does not enter anything but 0) the number of accepted and dropped packets drops to 0.

- The number of accepted/dropped packets is maintained inside the hook functions.

Cleanup:
Everyting is removed in opposite order to the initialization.