



Workshop in Information Security – Homework #2

Description:

In this assignment, we will finish to prepare all we need in order to create a real Firewall.

In the previous assignment, we wrote a kernel module that uses Netfilter API to manipulate packets within the kernel space. Now we will learn how to communicate with this kernel module from user space.

Every computer system needs user-interface, in order to configure it and get an updates from it. The UI needs to be able to communicate with the program, not matters where the program runs; in user space or kernel space. Therefore, in this assignment we will implement a way to communicate with our kernel module from previous assignment by reading data from it and writing data to it.

In addition, we will write a user-space program that will communicate with the kernel module via our interface.

Reading the data in user space:

We want to have a user-space program that will tell the user:

- How many packets our system processed in total so far
- How many packets passed so far
- How many packets dropped so far

Add to your kernel module from previous assignment the Sysfs interface as we learned in the lecture. You can use the template from the Moodle.

Write a user-space program in every programming language that the FW machine supports.

The program may get zero arguments, or one argument, "0". If the program gets more than one argument or the argument is not "0", exit with error message.

The program will print to the standard output (current terminal) in this format, **if no argument passed:**

```
fw@fw:~$ ./a.out
Firewall Packets Summary:
Number of accepted packets: <number>
Number of dropped packets: <number>
Total number of packets: <number>
```

For example:

```
fw@fw:~$ ./a.out
Firewall Packets Summary:
Number of accepted packets: 182
Number of dropped packets: 96
Total number of packets: 278
```

If argument "0" passed, you have to reset all the counters without printing anything.

In this assignment, you have to use the exact module you created in the previous assignment, you have only to add the interface to the user-space and of course, write the user-space program.

Submission

Prepare a ZIP file, contains:

- "module" folder, includes hw2secws.c file (the module) and the Makefile
- "user" folder, includes the user-space program and (if needed) the Makefile
- Dry documentation

General rules for submission, valid for the next assignments as well:

- Document your code
- If you use a code from the internet, document it and add the source
- Individual submission
- If needed, split the files into a modular files

Good luck!