A stateless and statefull firewall , implemented at the linux kernel. The implementation includes two char devices:

1.Fw_rules – managing the table of rules.

2.Fw_log – managing the list of rules.

How to run the firewall and the proxy :

1.the folder fw contains make file. After writing make at the command shell, writing insmod.ko will load the firewall module to the kernel.

2.the folder interface contains make file. After writing make at the command shell, writing \a.out command ,will run the command specified .

3.the folder proxy contains two files : proxy_part_1.py and proxy_part_2.py.
In order to start the proxy going you should open two command shell windows and write :

In the first window write:

python proxy_part_1.py

in the second window write :

python proxy_part_2.py

From this point the proxy will be up and listen and redirect, ftp and http packets.

Description of the firewall work:

1.The program start at the user space, when the main functions wait for input from the user.

2.The user can do one of the following commands: activate, deactivate, show_rules,clear_rules, load rules, show log , clear log.

Description of the firewall:

Stateless

When the user writes load_rules with a path of of a rules-table file, the main function open the function load _ rules. load_rules checking the content of the file, using the function “check_content_of_file”. if the rule is valid, the line from the file is getting send to the kernel. in the kernel, using sysfs, the function rule_table_modify is getting open. This function send the rules to the function – “build_table_of_rules”.

build_table_of_rules - is in charge of saving the rules to a dynamic array. Is uses auxiliary function called –“addrule”. Add rule in turn uses 8 auxiliary functions to parse the data.

From this point, the load of rules are loaded to the kernel. Every packet in/out-going to the firewall is saved in specific struct called packet_t. every packet first, getting parsed using the function get_packet_details . then, using the function find match the array of rules getting scanned and compares the rules to the packet. The result getting send to a function called policy_enforcer which return the action that the firewall will do according to the result of the find function. then, the function Add_to_log_list is getting called. The rule of this function is to add the information to the log list.

The log list is a dynamic list where every node saves a different log.

From this point a user can press any of the previous actions mentioned above. every action will have a corresponding function at the user space which after checking will be send to the kernel. The kernel will act according to the command sent.

Stateful

The firewall has a connection table which keeps track on the Tcp connection. Every packet first going through the stateless table, and if passed going to the connection table.
In order to see the connection table – you should write – show_connection_table from the user space.
There two files – proxy_part_1.py and Proxy part_2.py which implements proxy via python. Every ftp or http packets are going through this transparent proxy. The Proxy is doing deep packet inspection at the application layer.

For Http : if the content length is more than 2000 the packets is not passed. Also, office files are not approved.

For Ftp: exe files are not approved. Also, the proxy is examining the ftp server return codes for the first steps of the authentication. For every status from the ftp server , the proxy forecasts the next return status.