תומר אמיר
302121645

<u>סדנא באבטחת מידע</u>
<u>תרגיל 4 – Stateful Inspection</u>

בתרגיל זה נדרשנו לבנות מודול ששומר על מצב (State) לאורך החיבור בין ה-client וה-server.

מבנה הקוד:

<u>Module</u>:

Header Files:

- **fw.h**
  The main header file that has all the includes of external libraries, and includes the other header files so the modules can interact with each other.
  It also holds the definitions for globally used types that I didn't think belong to any specific component.
- **log.h, stateful.h, stateless.h**
  These files hold the global variables, function declarations and type definitions for their respective components.
- **fwmod.c** : This is the main C file, where all the devices are created and removed, and where the hook function is defined.
  functions:
  - **get_packet**: Receives the a packet from the hook function, creates a rule_t that I can work with, and then calls the right function that checks if the packet should be dropped or accepted.
  - **hook_func**: A simple function that grabs the packet, and passes it to the get_packet function.
  - **rules_show_func**: Returns the number of rules in the static table to the user space.
  - **rules_store_func**: Dummy function for the device attributes definition.
  - **dynamic_show_func**: Returns the number of rules in the dynamic table to the user space.
  - **dynamic_store_func**: Dummy function for the device attributes definition.
  - **active_show_func**: Dummy function for the device attributes definition.
  - **active_store_func**: Activates or deactivates the firewall according to input from the user.
  - **log_show_func, log_store_func**: Dummy functions for the device attributes definition.
  - **read_log**: This function returns the log entries one by one.
  - **clear_log_func**: As the name states – clears the log.
  - **log_size_show_func**: Return the size of the log to the user space
  - **log_size_store_func**: Dummy function for the device attributes definition.
  - **write_rule**: Receives static rules from the user space, and adds them to the static table.
  - **read_rules**: Returns the rules in the static table to the user space, one by one.
  - **read_dynamic**: Return the rules in the dynamic connection table to the user space one by one. Before returning the rule, the function check if it is still active, or the timeout has passed (if so, the rule is removed and not returned to the user).
  - **module_init_function**: Initializes all the devices and the hook.
  - **module_exit_function**: Removes all the devices and the hook.
- **log.c**
  functions:
  - **clear_log**: The name says it. It iterates over the log (which is made as a linked list) and frees the memory of each link.
  - **log_entry**: Creates a new log entry and adds it to the start of the log linked list according to a given rule.

תומר אמיר
302121645

- **stateless.c**: This file handles the stateless packet filtering.
  functions:
  - **clear_rules**: Clears the rule table.
  - **check_static_action**: after a connection arrives, this function checks if there is a corresponding rule in the static table. If so, I executes the action in the rule (accept/drop packet). If not, the default is accepting the packet.
- **stateful.c**: This file handles the stateful inspection, and manages the connection table.
  functions:
  - **clear_dynamic_table**: Clears the dynamic table (before removing the module).
  - **create_dynamic_rule**: Creates a new dynamic rule according to given connection parameters (as a static rule), and decides whether this is a FTP connection, HTTP connection or an OTHER_TCP connection (This function will only be reached by TCP packets).
  - **update_ftp_rule**: When a connection is already created and recognized as an FTP connection, it arrives to this function which controls the flow of the connection. Also, this function creates a new rule in the table when the data link is specified.
  - **update_http_rule**: This function controls HTTP connections. It also creates a new rule when it recognizes a REDIRECT response (30x response code).
    The redirect will only work with IP addresses, since no DNS was available in the development.
  - **update_connection_state**: Decides which update function to call according to the connection protocol.

Interface:

There is nothing special about the interface. It does everything specified in the HW file.