Workshop in Information Security – HW 4+5

This documentation file explains the design and logics behind my implementation of exercise 4+5.

Please note that I did not go into detail with each function as I did in previous documentation files (of exercises 1,2,3) as most functions seemed self-explanatory and I thought it not to be the main goal of the documentation of the final project.

User Space Management Interface

Possible commands

./main activate will activate the firewall.

./main deactivate will deactivate it.

./main show_rules will show the rules by loaded. If none are loaded, by default the firewall has the localhost rule.

./main clear_rules will clear all rules but the localhost rule.

./main load_rules <path> will load the rules from the file in the specified path, line by line, by the format specified in ex3.

sudo ./main show_log will show the logs by the format specified in ex3.

./main clear_log will clear the log.

./main show_connection_table will show the connection table by the format specified in ex4.

- An action is done based on the argument passed by the user.
  These actions allow us to send and receive data to and from the FW.
  In most cases (apart from the show_log) the function opens and reads/writes from/to a sysfs device file.
  In some of them, some parsing is done – for example: in load_rules – we parse the data we got from the file in the path we got as an argument and in show_rules – we parse the data we got from the sysfs device file.

main.c contains parsing the user input and parsing + printing functions in accordance with what was requested by the user.

aux.c contatins some auxiliary functions for the parsing.

Proxies

All proxies are implemented by the select method.

In all proxies we listen to a "fixed port" in a "fixed" ip address, and connect to an ip and port we get from the kernel. We also send the proxy port to the kernel and it is stored in the relevant connection table line, in order to redirect the packets accordingly.

In general, we transfer data between the sockets, meaning – if we get data in one of the sockets and we think it is "OK" (we are not inclined to block it) – we send it to through the other socket.

In case of blocked data we shut down the sockets and close the connection in the FW by informing the kernel about it – we have a sysfs device we write into.

http proxy (10.0.1.3,8007 and 10.0.2.3,random_port):

Blocking terms:

- conn_in – c_code
- conn_out: no Content Length label OR word doc with content length of more than 2000 – tested by comparing the magic number byte by byte to the asci table values expected of those of a word doc.

the ftp21 and ftp20 communicate between them using a txt file called signal.txt.

ftp21 proxy(10.0.1.3,8008 and 10.0.2.3,random_port):

conn_in: looking for the PORT command.

ftp20 proxy(10.0.2.3,8009 and 10.0.1.3,random_port):

Blocking terms:

- conn_in – exe files

smtp proxy(10.0.1.3,8006 and 10.0.2.3,random_port):

- conn_in – c_code

Kernel Space Management Interface

The 'firewall' module has the following components:

1. **netfilter component**: it has a pre-routing and output hook registered to it.
   In each hook, depending on the packet details, the routing decision is being made and there might be a redirection of the packet to the proxy, which will be explained later.
2. **rules component**: implemented by a static table of size 50 (which is the max by ex3's definition). These rules should be loaded by the network administrator.
3. **log component**: implemented by a dynamic array. These logs are recoded by in the hook functions.
4. **conn component:** implemented by a dynamic array. Connections are added and maintained only for the tcp protocol.

Char devices, sysfs devices and attributes:

1. There is one char device called "fw_log".
   When the user initiates the show_log command the open and then read fops of this char device are being called.
2. There is one class under which all of the sysfs devices are created, which is called "fw".
3. There is one sysfs device by the name "fw_rules" under which there are the following attributes:
   - rules size (R) with display_rules_size.
   - clear rules (W) with modify_clear_rules.
   - show_rules (R) with display_show_rules.
   - load_rules (W) with modify_load_rules.
   - active (RW) with display_active and modify_active.
4. There is one sysfs device by the name "fw_log" under which there are the following attributes:
   - log_size (R) with display_log_size.
   - log_clear(W) with modify_log_clear.
5. There is one sysfs device by the name "conn_tab" under which there are the following attributes:
   - conn_table_size (R) with display_conn_table_size.
   - conns (W) with modify_connection.
   - proxy (RW) with display_to_proxy and modify_from_proxy.
   - show_conn_table (R) with display_show_connections.

Packet Filtering: When a packet is filtered in one of the hooks the data is extracted from it (direction, ip, port, protocol, etc..), and then a decision is made and there may be a redirection to the http/ftp/smtp proxy server:

- If the protocol is not TCP then the packet is checked against the static rule table.
- If the protocol is TCP:
  - If it is a SYN packet with ACK=0 (meaning this is the connection initiation) then the packet is checked against the static rule table and a connection row is added to the connection table.
    - In the case of http/ftp: in the pre-routing hook a "normal" row will be added and in the output hook a reversed row will be added.
      Eventually in order for a connection to be established/closed/etc both lines should have that state.
  - If it is a packet with ACK=1 (meaning this connection was already initiated and the three-way handshake had already begun), then the packet is checked against the connection table – we check whether the current packet's flags are in correspondence with the TCP state machine. If so, we change the state of the suitable line in the connection table.
  - HTTP packets will be redirected to the http proxy – in the prerouting hook they will simply be redirected to 10.0.1.3 and 8007 and in the output hook they will be redirected to the data we got from the proxy.
  - FTP and SMTP packets will be redirected similarly

- When the module is loaded, the firewall is initially inactive.