Workshop in Information Security

Exercise 1

Eden Koveshi - 316221746

```
git repository: https://github.com/edenkoveshi/infosec-ws
```

Part I
Network Configuration

1 Hosts Configuration

Both hosts are configured similarly.

I configured them by modifying /etc/network/interfaces file, and then ifup -a

The interfaces file (host2 in braces):

```
auto eth0 (eth1)
iface eth0 (eth1) inet static

    address 10.0.1.1 (10.0.2.2)
    network 10.0.1.0 (10.0.2.0)
    netmask 255.255.255.0
    gateway 10.0.1.3 (10.0.2.3)
```

That's it for the hosts

2 Firewall Configuration

Firewall host has 3 interfaces, eth0,eth1 and eth2.

Again, I modified interfaces file, and then ifup -a

The interfaces file:

```
auto eth0
iface eth0 inet static
```

```
    address 10.0.1.3
    network 10.0.1.0
    netmask 255.255.255.0

auto eth1
iface eth1 inet static

    address 10.0.2.3
    network 10.0.2.0
    netmask 255.255.255.0

auto eth2
iface eth2 inet dhcp //this one is for internet connection
```

I followed http://www.ducea.com/2006/08/01/how-to-enable-ip-forwarding-in-linux/ to enable IP forwarding

Part II
Code

As the exercise demands, my kernel module ("packet-sniffer") passes and blocks packets, according to their source and destination IP.

There are two hooks:

HOOK 1:

as stated, this piece of code is partially taken from https://stackoverflow.com/questions/13071054/how-to-echo-a-packet-in-kernel-space-using-netfilter-hooks

The hooknum is NF_INET_PRE_ROUTING to catch *incoming* packets, before making a routing decision.

The function that's called upon catching a packet is *inspect_incoming_packet:*

After passing error checks, the function creates an IP header from the sk_buff struct containing packet information.

It extracts the *destination address (daddr)* field, converts it to Little Endian using be32_to_cpu.

Then it decides whether the packet passes or not.

A packet passes iff is destined to the FW.

HOOK 2:

This hook is of type NF_INET_LOCAL_OUT to catch *outgoing* packets

The function that's called upon catching a packet is *inspect_outgoing_pkt:*

It is defined exactly the same,only that this time it extracts the *source address (saddr)* field and checks whether it belongs to the FW or not, and decides whether to pass the packet or not accordingly.

Init and Exit functions:

As this is a kernel module, it has init and exit functions.

The init function registers the hooks, and the exit function unregisters them.