

Device Driver Lab-8

How a SNULL works

SNULL stands for **S**imple **N**etwork **U**tility for **L**oading **L**ocalities . It creates two interfaces. It is similar to the loopback interface but not the same. In SNULL, messages transmitted in one interface loops back to another interface. For better demonstration of the task of writing a network driver, it simulates the conversations with real remote hosts. Linux loopback driver is simpler than the network driver. It can see/check the code of the loopback driver. It is present in `drivers/net/loopback.c`

There needs a modification in the source and destination address during data transmission in order to establish a communication through the SNULL interface. It means that when packets are sent through one interface and received by another interface then the receiver of outgoing packets should not be recognised as local host. It is also applicable to the source address of the received packets. It is done by SNULL in a simple manner. It toggles the least significant bit of 3rd octet of the IP Address of the source and the destination. It is important to change the source otherwise when the packet comes back, users will be able to see the sender as themselves, but changing the destination is more crucial because without this we will not receive the packet.

Example:

Let IP[4] represents the 32 bit address which has four parts
part1 - IP[0], part2 - IP[1], part3 - IP[2], part4 - IP[3].

IP address is represented like IP[0] . IP[1] . IP[2] . IP[3]

Each part consists of 8 bit and we need to change IP[2]

```
Snull() {
```

```
// Some initializations
```

```
IP[2] ^= 1;
```

```
// Rebuild checksum
```

```
// Packet ready to be sent
```

```
}
```

When last bit of 3rd octet of IP address is changed using XOR operator, the packet is sent

The packet is now sent and the operation is done by swapping the last bit of 3rd octet of IP address using XOR operator. In this way the user sees that the packet is coming from another network and he can test other codes which take care of incoming packets from other networks.