Configuring Unsupported Devices for NetFlow Export (Port Mirroring)

In a situation when your network device does not have NetFlow capability (supports sFlow, NSEL, some other or no export protocol), you can still use a server with a NetFlow probe to analyze traffic from the network device and to generate a NetFlow statistics. We will call this server the NetFlow Daemon Server. Figure below shows an example of this situation:

On this page:

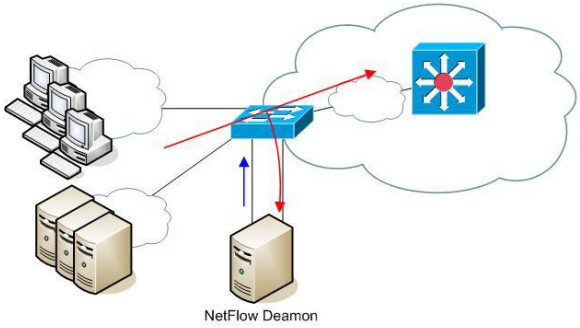
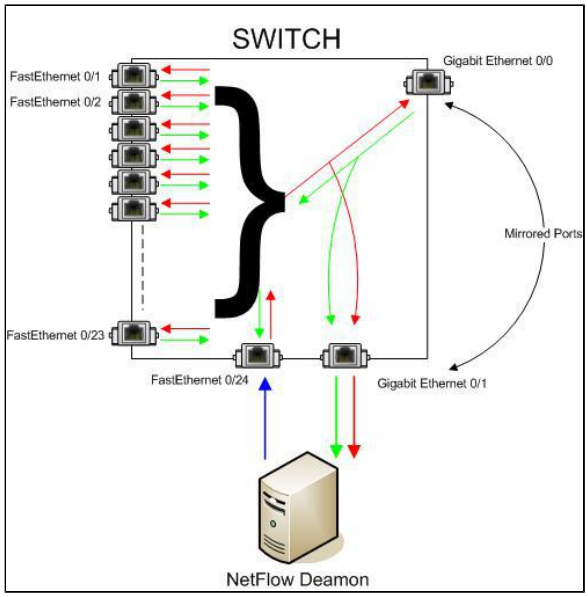


Figure below shows a more detailed illustration. Traffic from uplink interface *Gigabit Ethernet 0/0* is forwarded (mirrored) to interface *Gigabit Ethernet 0/1,* which is connected to the NetFlow Daemon Server. When the port mirroring is started, interface on a switch to whom all traffic is forwarded to becomes useless for normal device communication. It only passes all traffic from a mirroring interface. It will not be possible to collect statistics about the local traffic which doesn't pass uplink interface.



The problem is: How to export NetFlow traffic if the interface on which the NetFlow Daemon Server is connected is unusable for normal communication?

NetFlow daemon server must have two network cards, one for receiving mirrored traffic (eth1) and another one for exporting NetFlow statistics (eth0). This configuration enables NetFlow exporting even from L2 switches. The drawback is the additional port utilization on the switch and the need for an additional server with two network cards. The blue arrow in the figure above shows NetFlow export from the additional network card on the server. Now, it is possible to start the NetFlow probe on the NetFlow Daemon Server.

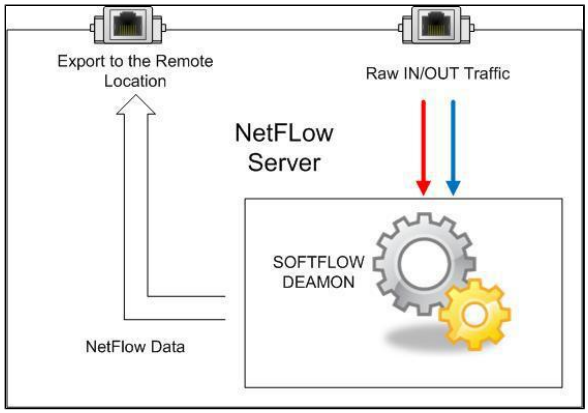
Configuring Cisco Device

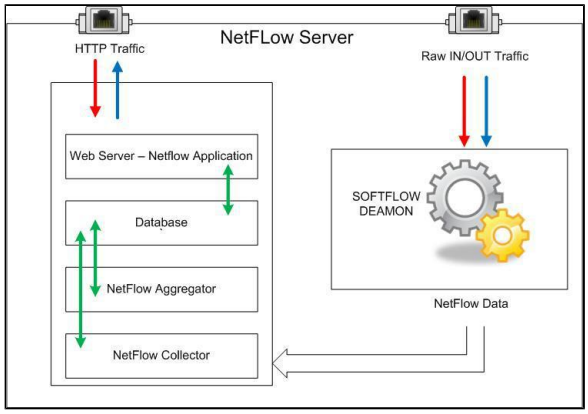
An example of how to configure port mirroring on a Cisco device is shown below.

```
switch(config)#monitor session 1 source interface Gi0/0
switch(config)#monitor session 1 destination interface Gi0/1
```

Using NetFlow Probe

We will show an example of configuration with a free NetFlow probe software called **softflowd**. It has the possibility of exporting NetFlow traffic locally (127.0.0.1) to an UDP port on the same server or to an UDP port on a remote server.





Here you can see how to configure the softflowd on Linux. In our example we are using version softflowd 0.9.9

1. Install softflowd depending on your Debian/Ubuntu distribution:

```
apt install softflowd -y
```

2. To configure softflowd you should edit /etc/softflowd/default.conf and specify for example:

```
INTERFACE="ens18"
OPTIONS="-n 172.16.0.77:2055"
```

3. To be able to run the software in the foreground in Debug mode you should use the following command:

```
softflowd -D -v 5 -i ens18 -n 10.10.10.10:2055 -T full
```

4. Now, you should be able to see that the flows are collecting and that they are being exported in the NetFlow version 5 and set to 10.10.10.10 using destination port 2055, where 10.10.10.10 is destination address of the NetFlow server in this case. This can be done using a utility such as TCPDUMP:

```
tcpdump -n –v dst port 2055
```

5. Now enable softflowd service to start at runtime:

```
systemctl enable softflowd
```

> ⚠ Note that you should change INTERFACE="ens18" and COLLECTOR="10.10.10.10" values with your own.

6. Start the service:

```
systemctl start softflowd
```