Packet Sniffer

Document revision 1.6 (February 5, 2008, 15:52 GMT)
This document applies to V3.0

Table of Contents

General Information

Summary

Packet sniffer is a feature that catches all the data travelling over the network, that it is able to get (when using switched network, a computer may catch only the data addressed to it or is forwarded through it).

Specifications

Packages required: ***system***
License required: ***level1***

Home menu level: ***/tool sniffer***
Standards and Technologies: ***none***
Hardware usage: ***Not significant***

Description

It allows you to "sniff" (listen and record) packets going through the router (and any other traffic that gets to the router, when there is no switching in the network) and view them using specific software.

Packet Sniffer Configuration

Home menu level: ***/tool sniffer***

Property Description

file-limit (*integer*; default: **10**) - the limit of the file in KB. Sniffer will stop after this limit is reached

file-name (*text*; default: **""**) - the name of the file where the sniffed packets will be saved to

filter-address1 (*IP addressnetmaskport*; default: **0.0.0.0/0:0-65535**) - criterion of choosing the packets to process

filter-address2 (*IP addressnetmaskport*; default: **0.0.0.0/0:0-65535**) - criterion of choosing the packets to process

filter-protocol (*all-frames* | *ip-only* | *mac-only-no-ip*; default: **ip-only**) - specific protocol group to filter

- **all-frames** - sniff all packets
- **ip-only** - sniff IP packets only
- **mac-only-no-ip** - sniff non-IP packets only

filter-stream (*yes* | *no*; default: **yes**) - whether to ignore sniffed packets that are destined to the stream server

interface (*name* | *all*; default: **all**) - the name of the interface that receives the packets

memory-limit (*integer*; default: **10**) - maximum amount of memory to use. Sniffer will stop after this limit is reached

only-headers (*yes* | *no*; default: **no**) - whether to save in the memory packets' headers only (not the whole packet)

running (*read-only: yes* | *no*; default: **no**) - if the sniffer is started then the value is yes otherwise no

streaming-enabled (*yes* | *no*; default: **no**) - whether to send sniffed packets to a remote server

streaming-server (*IP address*; default: **0.0.0.0**) - Tazmen Sniffer Protocol (TZSP) stream receiver

Notes

filter-address1 and **filter-address2** are used to specify the two participients in communication (i.e. they will match only in the case if one of them matches the source address and the other one matches the destination address of a packet). These properties are taken in account only if **filter-protocol** is **ip-only**.

Not only **Wireshark** (ex-Ethereal, http://www.wireshark.org) and **Packetyzer** (http://www.packetyzer.com) can receive the sniffer's stream but also MikroTik's program **trafr**

http://www.mikrotik.com/download.html) that runs on any IA32 Linux computer and saves received packets **libpcap** file format.

Example

In the following example **streaming-server** will be added, streaming will be enabled, **file-name** will be set to *test* and packet sniffer will be started and stopped after some time:

```
[admin@MikroTik] tool sniffer>set streaming-server=10.0.0.241 \
\... streaming-enabled=yes file-name=test
[admin@MikroTik] tool sniffer> prin
            interface: all
         only-headers: no
         memory-limit: 10
            file-name: "test"
           file-limit: 10
    streaming-enabled: yes
     streaming-server: 10.0.0.241
        filter-stream: yes
      filter-protocol: ip-only
      filter-address1: 0.0.0.0/0:0-65535
      filter-address2: 0.0.0.0/0:0-65535
              running: no
[admin@MikroTik] tool sniffer>start
[admin@MikroTik] tool sniffer>stop
```

Running Packet Sniffer

Command name: ***/tool sniffer start, /tool sniffer stop, /tool sniffer save***

Description

The commands are used to control runtime operation of the packet sniffer. The **start** command is used to start/reset sniffering, **stop** - stops sniffering. To save currently sniffed packets in a specific file **save** command is used.

Example

In the following example the packet sniffer will be started and after some time - stopped:

```
[admin@MikroTik] tool sniffer> start
[admin@MikroTik] tool sniffer> stop
```

Below the sniffed packets will be saved in the file named *test*:

```
[admin@MikroTik] tool sniffer> save file-name=test
[admin@MikroTik] tool sniffer> /file print
  # NAME                          TYPE         SIZE       CREATION-TIME
  0 test                          unknown      1350       apr/07/2003 16:01:52

[admin@MikroTik] tool sniffer>
```

Sniffed Packets

Home menu level: ***/tool sniffer packet***

Description

The submenu allows to see the list of sniffed packets.

Property Description

data (*read-only: text*) - specified data inclusion in packets

dst-address (*read-only: IP address*) - destination IP address

dst-mac-address (*MAC address*) - destination MAC address

fragment-offset (*read-only: integer*) - IP fragment offset

identification (*read-only: integer*) - IP identification

interface (*read-only: name*) - name of the interface the packet has been captured on

ip-header-size (*read-only: integer*) - the size of IP header

ip-packet-size (*read-only: integer*) - the size of IP packet

ip-protocol (*ip* | *icmp* | *igmp* | *ggp* | *ipencap* | *st* | *tcp* | *egp* | *pup* | *udp* | *hmp* | *xns-idp* | *rdp* | *iso-tp4* | *xtp* | *ddp* | *idrp-cmtp* | *gre* | *esp* | *ah* | *rspf* | *vmtp* | *ospf* | *ipip* | *encap*) - the name/number of IP protocol

- **ip** - Internet Protocol
- **icmp** - Internet Control Message Protocol
- **igmp** - Internet Group Management Protocol
- **ggp** - Gateway-Gateway Protocol
- **ipencap** - IP Encapsulated in IP
- **st** - st datagram mode
- **tcp** - Transmission Control Protocol
- **egp** - Exterior Gateway Protocol
- **pup** - Parc Universal packet Protocol
- **udp** - User Datagram Protocol
- **hmp** - Host Monitoring Protocol
- **xns-idp** - Xerox ns idp
- **rdp** - Reliable Datagram Protocol
- **iso-tp4** - ISO Transport Protocol class 4
- **xtp** - Xpress Transfer Protocol
- **ddp** - Datagram Delivery Protocol
- **idpr-cmtp** - idpr Control Message Transport
- **gre** - General Routing Encapsulation
- **esp** - IPsec ESP protocol
- **ah** - IPsec AH protocol
- **rspf** - Radio Shortest Path First
- **vmtp** - Versatile Message Transport Protocol
- **ospf** - Open Shortest Path First
- **ipip** - IP encapsulation (protocol 4)
- **encap** - IP encapsulation (protocol 98)

protocol (*read-only: ip | arp | rarp | ipx | ipv6*) - the name/number of ethernet protocol

- **ip** - Internet Protocol
- **arp** - Address Resolution Protocol
- **rarp** - Reverse Address Resolution Protocol
- **ipx** - Internet Packet exchange protocol
- **ipv6** - Internet Protocol next generation

size (*read-only: integer*) - size of packet

src-address (*IP address*) - source address

src-mac-address (*MAC address*) - source MAC address

time (*read-only: time*) - time when packet arrived

tos (*read-only: integer*) - IP Type Of Service

ttl (*read-only: integer*) - IP Time To Live

Example

In the example below it's seen, how to get the list of sniffed packets:

```
[admin@MikroTik] tool sniffer packet> print
  # TIME    INTERFACE SRC-ADDRESS             DST-ADDRESS            IP-.. SIZE
  0 0.12    ether1    10.0.0.241:1839         10.0.0.181:23 (telnet) tcp   46
  1 0.12    ether1    10.0.0.241:1839         10.0.0.181:23 (telnet) tcp   40
  2 0.12    ether1    10.0.0.181:23 (telnet)  10.0.0.241:1839        tcp   78
  3 0.292   ether1    10.0.0.181              10.0.0.4               gre   88
  4 0.32    ether1    10.0.0.241:1839         10.0.0.181:23 (telnet) tcp   40
  5 0.744   ether1    10.0.0.144:2265         10.0.0.181:22 (ssh)    tcp   76
  6 0.744   ether1    10.0.0.144:2265         10.0.0.181:22 (ssh)    tcp   76
  7 0.744   ether1    10.0.0.181:22 (ssh)     10.0.0.144:2265        tcp   40
  8 0.744   ether1    10.0.0.181:22 (ssh)     10.0.0.144:2265        tcp   76
[admin@MikroTik] tool sniffer packet>
```

Packet Sniffer Protocols

Home menu level: ***/tool sniffer protocol***

Description

In this submenu you can see all kind of protocols that have been sniffed.

Property Description

bytes (*integer*) - total number of data bytes

ip-protocol (*ip | icmp | igmp | ggp | ipencap | st | tcp | egp | pup | udp | hmp | xns-idp | rdp | iso-tp4 | xtp | ddp | idrp-cmtp | gre | esp | ah | rspf | vmtp | ospf | ipip | encap*) - the name/number of IP protocol

- **ip** - Internet Protocol
- **icmp** - Internet Control Message Protocol
- **igmp** - Internet Group Management Protocol
- **ggp** - Gateway-Gateway Protocol

- **ipencap** - IP Encapsulated in IP
- **st** - st datagram mode
- **tcp** - Transmission Control Protocol
- **egp** - Exterior Gateway Protocol
- **pup** - Parc Universal packet Protocol
- **udp** - User Datagram Protocol
- **hmp** - Host Monitoring Protocol
- **xns-idp** - Xerox ns idp
- **rdp** - Reliable Datagram Protocol
- **iso-tp4** - ISO Transport Protocol class 4
- **xtp** - Xpress Transfer Protocol
- **ddp** - Datagram Delivery Protocol
- **idpr-cmtp** - idpr Control Message Transport
- **gre** - General Routing Encapsulation
- **esp** - IPsec ESP protocol
- **ah** - IPsec AH protocol
- **rspf** - Radio Shortest Path First
- **vmtp** - Versatile Message Transport Protocol
- **ospf** - Open Shortest Path First
- **ipip** - IP encapsulation
- **encap** - IP encapsulation

packets (*integer*) - the number of packets

port (*name*) - the port of TCP/UDP protocol

protocol (*read-only: ip | arp | rarp | ipx | ipv6*) - the name/number of ethernet protocol

- **ip** - Internet Protocol
- **arp** - Address Resolution Protocol
- **rarp** - Reverse Address Resolution Protocol
- **ipx** - Internet Packet exchange protocol
- **ipv6** - Internet Protocol next generation

share (*integer*) - specific type of traffic share compared to all traffic in bytes

Example

```
[admin@MikroTik] tool sniffer protocol> print
  # PROTOCOL IP-PR... PORT          PACKETS   BYTES   SHARE
  0 ip                              77        4592    100 %
  1 ip       tcp                    74        4328    94.25 %
  2 ip       gre                    3         264     5.74 %
  3 ip       tcp      22 (ssh)      49        3220    70.12 %
  4 ip       tcp      23 (telnet)   25        1108    24.12 %

[admin@MikroTik] tool sniffer protocol>
```

Packet Sniffer Host

Home menu level: ***/tool sniffer host***

Description

The submenu shows the list of hosts that were participating in data excange you've sniffed.

Property Description

address (*read-only: IP address*) - IP address of the host
peek-rate (*read-only: integerinteger*) - the maximum data-rate received/transmitted
rate (*read-only: integerinteger*) - current data-rate received/transmitted
total (*read-only: integerinteger*) - total packets received/transmitted

Example

In the following example we'll see the list of hosts:

```
[admin@MikroTik] tool sniffer host> print
  # ADDRESS       RATE          PEEK-RATE           TOTAL
  0 10.0.0.4      0bps/0bps     704bps/0bps         264/0
  1 10.0.0.144    0bps/0bps     6.24kbps/12.2kbps   1092/2128
  2 10.0.0.181    0bps/0bps     12.2kbps/6.24kbps   2994/1598
  3 10.0.0.241    0bps/0bps     1.31kbps/4.85kbps   242/866

[admin@MikroTik] tool sniffer host>
```

Packet Sniffer Connections

Home menu level: ***/tool sniffer connection***

Description

Here you can get a list of the connections that have been watched during the sniffing time.

Property Description

active (*read-only: yes | no*) - if yes the find active connections
bytes (*read-only: integerinteger*) - bytes in the current connection
dst-address (*read-only: IP address*) - destination address
mss (*read-only: integerinteger*) - Maximum Segment Size
resends (*read-only: integerinteger*) - the number of packets resends in the current connection
src-address (*read-only: IP address*) - source address

Example

The example shows how to get the list of connections:

```
[admin@MikroTik] tool sniffer connection> print
Flags: A - active
  #   SRC-ADDRESS       DST-ADDRESS             BYTES     RESENDS   MSS
  0 A 10.0.0.241:1839   10.0.0.181:23 (telnet)  6/42      60/0      0/0
  1 A 10.0.0.144:2265   10.0.0.181:22 (ssh)     504/252   504/0     0/0

[admin@MikroTik] tool sniffer connection>
```

Sniff MAC Address

You can also see the source and destination MAC Addresses. To do so, at first stop the sniffer if it is running, and select a specific interface:

```
[admin@MikroTik] tool sniffer> stop
[admin@MikroTik] tool sniffer> set interface=bridge1
[admin@MikroTik] tool sniffer> start
[admin@MikroTik] tool sniffer> print
           interface: bridge1
        only-headers: no
        memory-limit: 10
           file-name:
          file-limit: 10
    streaming-enabled: no
     streaming-server: 0.0.0.0
        filter-stream: yes
      filter-protocol: ip-only
      filter-address1: 0.0.0.0/0:0-65535
      filter-address2: 0.0.0.0/0:0-65535
              running: yes
[admin@MikroTik] tool sniffer>
```

Now you have the source and destination MAC Addresses:

```
[admin@MikroTik] tool sniffer packet> print detail
 0 time=0 src-mac-address=00:0C:42:03:02:C7 dst-mac-address=00:30:4F:08:3A:E7
   interface=bridge1 src-address=10.5.8.104:1125
   dst-address=10.1.0.172:3987 (winbox-tls) protocol=ip ip-protocol=tcp
   size=146 ip-packet-size=146 ip-header-size=20 tos=0 identification=5088
   fragment-offset=0 ttl=126

 1 time=0 src-mac-address=00:30:4F:08:3A:E7 dst-mac-address=00:0C:42:03:02:C7
   interface=bridge1 src-address=10.1.0.172:3987 (winbox-tls)
   dst-address=10.5.8.104:1125 protocol=ip ip-protocol=tcp size=253
   ip-packet-size=253 ip-header-size=20 tos=0 identification=41744
   fragment-offset=0 ttl=64

 2 time=0.071 src-mac-address=00:0C:42:03:02:C7
   dst-mac-address=00:30:4F:08:3A:E7 interface=bridge1
   src-address=10.5.8.104:1125 dst-address=10.1.0.172:3987 (winbox-tls)
   protocol=ip ip-protocol=tcp size=40 ip-packet-size=40 ip-header-size=20
   tos=0 identification=5089 fragment-offset=0 ttl=126

 3 time=0.071 src-mac-address=00:30:4F:08:3A:E7
   dst-mac-address=00:0C:42:03:02:C7 interface=bridge1
   src-address=10.1.0.172:3987 (winbox-tls) dst-address=10.5.8.104:1125
   protocol=ip ip-protocol=tcp size=213 ip-packet-size=213 ip-header-size=20
   tos=0 identification=41745 fragment-offset=0 ttl=64

-- [Q quit|D dump|down]
```