

Netsniff-ng- Packet Sniffer

Packet sniffing is the process of capturing and analyzing network packets as they travel across a network. It involves intercepting data packets, which contain information such as source and destination addresses, protocols, and payload data. This technique is essential for network management, troubleshooting, and security monitoring, as it allows administrators to identify and resolve issues, monitor network performance, and detect malicious activities. Advanced packet sniffing tools, like Netsniff-ng, leverage high-efficiency capture methods, support for various network interfaces, and real-time analysis capabilities. These tools can handle high-speed network traffic with minimal performance impact, ensuring comprehensive and accurate data capture. Additionally, advanced features such as zero-copy capture, hardware-based timestamping, and packet injection and replay enhance their functionality, making them invaluable for sophisticated network diagnostics and security operations.

Netsniff-ng is a robust and efficient network packet sniffing and analysis tool for Linux, designed for high-speed network environments. As an open-source tool, it offers a suite of utilities for capturing, replaying, and analyzing network traffic. The core component, ``netsniff-ng``, allows for high-performance packet capture, minimizing packet loss and ensuring accurate data collection. The toolkit includes ``trafgen`` for traffic generation, ``mausezahn`` for crafting and sending custom packets, ``ifpps`` for interface performance

statistics, `flowtop` for live traffic flow monitoring, and `bpfc` for compiling Berkeley Packet Filter rules. `Netsniff-ng` operates via a command-line interface, making it ideal for scripting and automation. It supports advanced features such as zero-copy packet capture, hardware-based timestamping, and multi-interface support. This versatility and efficiency make `Netsniff-ng` a valuable tool for network administrators and security professionals.

Example: -

1. Firstly, we use flag -i to define our interface. In our case netsniff will capture packets from eth0 interface.

```
root@kali:~/home/kali# netsniff-ng -i eth0
Running! Hang up with ^C!

> eth0 54 1720372695s.353735016ns #1
[ Eth MAC (00:0c:29:f7:67:f8 => 00:50:56:f4:13:bd), Proto (0x0800, IPv
4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ IPv4 Addr (192.168.172.135 => 34.107.221.82), Proto (6), TTL (64), TOS (
0), Ver (4), IHL (5), Tlen (40), ID (49390), Res (0), NoFrag (1), MoreFr
ag (0), FragOff (0), CSum (0x0cf4) is ok ]
[ TCP Port (47570 => 80 (http)), SN (0xdc108a9c), AN (0x78e92c9c),
DataOff (5), Res (0), Flags (ACK), Window (64024), CSum (0x6d08), UrgPtr
(0) ]

< eth0 60 1720372695s.354293985ns #2
[ Eth MAC (00:50:56:f4:13:bd => 00:0c:29:f7:67:f8), Proto (0x0800, IPv
4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ Eth trailer 000000 ]
[ IPv4 Addr (34.107.221.82 => 192.168.172.135), Proto (6), TTL (120), TOS
(0), Ver (4), IHL (5), Tlen (40), ID (23331), Res (0), NoFrag (0), MoreF
rag (0), FragOff (0), CSum (0x72bf) is ok ]
[ TCP Port (80 (http) => 47570), SN (0x78e92c9c), AN (0xdc108a9d),
DataOff (5), Res (0), Flags (ACK), Window (64240), CSum (0x819f), UrgPtr
(0) ]

> eth0 93 1720372696s.919377279ns #3
[ Eth MAC (00:0c:29:f7:67:f8 => 00:50:56:f4:13:bd), Proto (0x0800, IPv
4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ IPv4 Addr (192.168.172.135 => 34.160.144.191), Proto (6), TTL (64), TOS
(0), Ver (4), IHL (5), Tlen (79), ID (38434), Res (0), NoFrag (1), MoreF
rag (0), FragOff (0), CSum (0x83f7) is ok ]
[ TCP Port (53084 => 443 (https)), SN (0xe67c0860), AN (0x612c7cc0)
```

```
16 packets incoming (0 unread on exit)
16 packets passed filter
0 packets failed filter (out of space)
0.0000% packet droprate
18 sec, 968810 usec in total
```

2. Now we used -f to filter a specific port. Here we capture TCP 80 port.

```
root@kali:~/homer/kali# netuiff-ng -i eth0 -f "tcp port 80"

Running! Hang up with "C"

> eth0 74 1720372758s.393912000ns #1
[ Eth MAC (00:0c:29:f7:67:fb => 00:50:56:f4:13:bd), Proto (0x0000, IPv4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ IPv4 Addr (192.168.172.135 => 44.228.249.3), Proto (6), TTL (64), TOS (0), Ver (4), IHL (5), Tlen (60), ID (28565), Res (0), NoFrag (1), MoreFrag (0),
FragOff (0), Csum (0x3fdf) is ok ]
[ TCP Port (51760 => 80 (http)), SN (0xd311e134), AN (0x0), DataOff (10), Res (0), Flags (SYN), Window (64240), CSum (0x9346), UrgPtr (0) ]
[ Chr .....W.F..... ]
[ Hex 02 04 05 b4 04 02 05 0a ff 77 08 46 00 00 00 01 03 03 0b ]

> eth0 74 1720372758s.394104740ns #2
[ Eth MAC (00:0c:29:f7:67:fb => 00:50:56:f4:13:bd), Proto (0x0000, IPv4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ IPv4 Addr (192.168.172.135 => 44.228.249.3), Proto (6), TTL (64), TOS (0), Ver (4), IHL (5), Tlen (60), ID (18962), Res (0), NoFrag (1), MoreFrag (0),
FragOff (0), Csum (0x5d92) is ok ]
[ TCP Port (51770 => 80 (http)), SN (0x1f3d47a8), AN (0x0), DataOff (10), Res (0), Flags (SYN), Window (64240), CSum (0x9346), UrgPtr (0) ]
[ Chr .....W.F..... ]
[ Hex 02 04 05 b4 04 02 05 0a ff 77 08 46 00 00 00 01 03 03 0b ]

> eth0 74 1720372758s.394158149ns #3
[ Eth MAC (00:0c:29:f7:67:fb => 00:50:56:f4:13:bd), Proto (0x0000, IPv4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ IPv4 Addr (192.168.172.135 => 44.228.249.3), Proto (6), TTL (64), TOS (0), Ver (4), IHL (5), Tlen (60), ID (25716), Res (0), NoFrag (1), MoreFrag (0),
FragOff (0), Csum (0x4330) is ok ]
[ TCP Port (51784 => 80 (http)), SN (0x3a4226ce), AN (0x0), DataOff (10), Res (0), Flags (SYN), Window (64240), CSum (0x9346), UrgPtr (0) ]
[ Chr .....W.F..... ]
[ Hex 02 04 05 b4 04 02 05 0a ff 77 08 46 00 00 00 01 03 03 0b ]

> eth0 74 1720372758s.644081494ns #4
[ Eth MAC (00:0c:29:f7:67:fb => 00:50:56:f4:13:bd), Proto (0x0000, IPv4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ IPv4 Addr (192.168.172.135 => 44.228.249.3), Proto (6), TTL (64), TOS (0), Ver (4), IHL (5), Tlen (60), ID (12161), Res (0), NoFrag (1), MoreFrag (0),
FragOff (0), Csum (0x4330) is ok ]
[ TCP Port (51784 => 80 (http)), SN (0x3a4226ce), AN (0x0), DataOff (10), Res (0), Flags (SYN), Window (64240), CSum (0x9346), UrgPtr (0) ]
[ Chr .....W.F..... ]
[ Hex 02 04 05 b4 04 02 05 0a ff 77 08 46 00 00 00 01 03 03 0b ]

43 packets incoming (0 unread on exit)
43 packets passed filter
0 packets failed filter (out of space)
0.0000% packet droprate
42 sec, 200225 usec in total
```

3. In this example, we use -o flag to store the output in a pcap file

```
(root@kali:~/home/kali)
#4 netstat-ng -i eth0 -o capture.pcap

Running! Hang up with ^C

> eth0 54 1720372821s.49015405ms #1
[ Eth MAC (00:0c:29:f7:67:f8 => 00:50:56:f4:13:bd), Proto (0x0800, IPv4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ IPv4 Addr (192.168.172.135 => 44.228.249.3), Proto (6), TTL (64), TOS (0), Ver (4), IHL (5), Tlen (40), ID (26575), Res (0), NoFrag (1), MoreFrag (0),
  FragOff (0), CSum (0x3fe9) is ok ]
[ TCP Port (51760 => 80 (http)), SN (0x311e27e), AN (0x4696290e), DataOff (5), Res (0), Flags (ACK), Window (62780), CSum (0x9332), UrgPtr (0)
  ]

< eth0 60 1720372821s.50101060ms #2
[ Eth MAC (00:0c:29:f7:67:f8 => 00:0c:29:f7:67:f8), Proto (0x0800, IPv4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ Eth trailer 000000 ]
[ IPv4 Addr (44.228.249.3 => 192.168.172.135), Proto (6), TTL (128), TOS (0), Ver (4), IHL (5), Tlen (40), ID (26137), Res (0), NoFrag (0), MoreFrag (0),
  FragOff (0), CSum (0x419f) is ok ]
[ TCP Port (80 (http) => 51760), SN (0x4696290e), AN (0x311e27f), DataOff (5), Res (0), Flags (ACK), Window (64240), CSum (0x3215), UrgPtr (0)
  ]

> eth0 54 1720372821s.501063352ms #3
[ Eth MAC (00:0c:29:f7:67:f8 => 00:50:56:f4:13:bd), Proto (0x0800, IPv4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ IPv4 Addr (192.168.172.135 => 44.228.249.3), Proto (6), TTL (64), TOS (0), Ver (4), IHL (5), Tlen (40), ID (25726), Res (0), NoFrag (1), MoreFrag (0),
  FragOff (0), CSum (0x433a) is ok ]
[ TCP Port (51784 => 80 (http)), SN (0x3a422843), AN (0x26e3432b), DataOff (5), Res (0), Flags (ACK), Window (64059), CSum (0x9332), UrgPtr (0)
  ]

< eth0 60 1720372821s.562627270ms #4
[ Eth MAC (00:0c:29:f7:67:f8 => 00:0c:29:f7:67:f8), Proto (0x0800, IPv4) ]
[ Vendor (VMware, Inc. => VMware, Inc.) ]
[ Eth trailer 000000 ]

15 packets incoming (0 unread on exit)
15 packets passed filter
0 packets failed filter (out of space)
0.0000% packet droprate
6 sec, 716629 usec in total
```