NetFlow

ein wenig beachtetes Tool zur passiven Netzwerkanalyse

Linuxwochen Eisenstadt
18. April 2026
Goesta Smekal

Überblick

- Definitionen
  (Protokolle, Begriffe, Standards)
- Tools
  (Erfassen, Sammeln, Analyse)
- Erkenntnisse

Wer bin ich?

... und warum tue ich euch das an?

Goesta Smekal
Jahrgang 1972
Computer seit 1986
Linux seit ~1995
LUGA seit 2007
Beruf: IT-Verantwortlicher in einem Verlag

Fragen, die nicht beantwortet werden

- Warum kann Gerät XY keine NetFlow-Daten senden?
- Was hältst Du von der proprietären Management-Lösung des Herstellers ...?
- Wie bekomme ich Zugriff auf meinen Router zu Hause?
- Hast Du kein moderneres Thema gefunden? Etwas mit KI?
- Warum schauen Deine Folien wie Kleinbild-Negative aus?

Worum, bitteschön, geht es da?

NetFlow - IPFIX - sFlow - NetStream - JFlow ..?

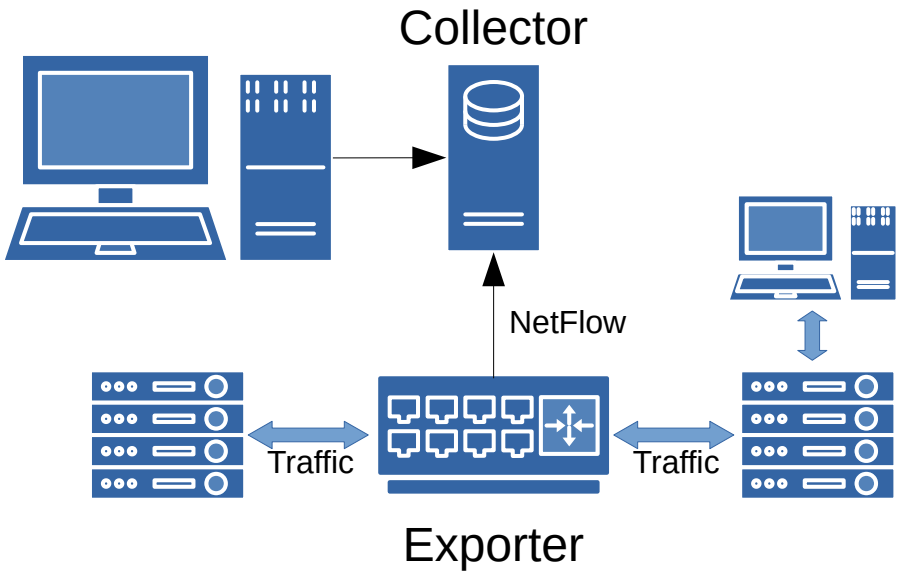
Ursprünglich von Cisco als Technik für effizientes Routing entwickelt (ca. 1996)

andere Hersteller, andere Bezeichnungen → Verwirrung

2004 - erster RFC für IPFIX (RFC 3955)

Terminologie

- **Flow** – ein Datenstrom zwischen zwei Hosts
- **Exporter** – Netzwerkgerät, das Flows erfasst
- **Collector** – Host, der Flows sammelt
- Analyse – das eigentlich Interessante



Worum, bitteschön, geht es da?

OSI / Internet Modell

OSI Layer	TCP/IP
7 - Presentation	Application (HTTP, SMTP, IMAP, SSH, LDAP, ...)
6 - Application	
5 - Session	
4 - Transport	Transport (TCP,UDP)
3 - Network	Internet (ARP, ICMP, IP, ...)
2 - Data Link	Network Access (Ethernet, Token Ring, Frame Relay, ...)
1 - Physical	

Flow vs. Traffic

- in einem Flow sind gleich:
- Ingress interface (SNMP ifIndex)
- Source IP address
- Destination IP address
- IP protocol number
- Source port for UDP or TCP, 0 for other protocols
- Destination port for UDP or TCP, type and code for ICMP, or 0 for other protocols
- IP Type of Service

Von der Session zum NetFlow

Verarbeitung von Flows:

- Datenströme aggregieren
- ein Datensatz je „Session“
- Datenmenge reduziert
- Sampling in modernen Netzen (100 User, Samplerate 1/100, ca. 150MB/d an NetFlow Daten)
- Übertragung als UDP-Paket (“fire and forget”)
- analysieren
- nicht sniffen
- keine Forensik

Ende eines Flows?

- bei TCP einfach:
  am Ende einer Connection immer Fin/Rst
- UDP: ...?Ageing
  jedes neue Paket in einem Flow setzt einen Timer zurück, ist der Flow zu „alt“ wird er als beendet betrachtet

Tools :: Exporter

In Netzwerk-Hardware integriert

- Switches
- Router
- Firewalls

Liefern Daten „aus erster Hand“

Sampling kann Daten reduzieren

Auf dediziertem Host

- ipt-netflow (Flows aus nftables/iptables)
- fprobe (Flows aus libpcap-Daten)

Können auch von Mirrorports versorgt werden

Tools :: Collector

nfdump

- nfcapd - der eigentliche Collector
- nfdump - Daten per CLI auswerten
- nfexpire - Housekeeping

```
apt install nfdump
vi /etc/nfdump/default.conf
options='-D -S 1 -M /var/cache/nfdump -p 2055'
adduser netflow --system --no-create-home --home /var/cache/nfdump
```

Tools :: Analyse

nfdump nativ

Beispiel: top-10 Flows nach Quelle, Ziel und Protokoll

```
goesta@hypercube:~$ nfdump -r /var/cache/nfdump/ -A srcip,dstip,proto,dstport -O flows -n 10

Date first seen   Duration                Src IP Addr      Dst IP Addr      Proto Pt Packets  Bytes  bps  Bpp Flows
2026-03-31 18:44:13.000 16d 05:50:26.000  2a02:ab..f::1305 2a02:ab..7::1001 TCP  443  62.5 M  6.9 G 39121 109   1
2026-03-31 18:33:11.000 16d 06:00:11.000  172.27.7.108     172.27.6.97      UDP  161  625520 60.0 M  342   95   1
2026-03-31 18:35:05.000 16d 05:58:20.000  172.27.7.108     172.27.8.28      UDP  161  440614 42.6 M  242   96   1
2026-03-31 18:33:20.000 16d 05:59:54.000  172.27.7.108     172.27.7.1       UDP  161  423579 37.6 M  214   88   1
2026-03-31 18:45:05.000 16d 05:48:20.000  172.27.7.108     172.27.6.1       UDP   53  162384 10.7 M   61   66   1
2026-03-31 18:45:23.000 16d 05:49:01.000  2a02:ab..7::c0d1 2a02:ab..7::b055 ICMP6  0  102261  7.0 M   39   67   1
2026-03-31 18:46:22.000 16d 05:40:43.000  2a02:ab..82:4494 2a02:ab..6::b055 UDP   53   78783  7.0 M   39   88   1
2026-03-31 18:45:06.000 16d 05:48:53.000  2a02:ab..1f4:8b3 2a02:ab..8::b055 ICMP6 0    36735  2.5 M   14   68   1
2026-03-31 18:45:05.000 14d 01:49:35.000  172.27.8.102     172.27.6.1       UDP   53   47017  2.9 M   19   62   1
2026-03-31 18:43:28.000 16d 05:40:57.000  172.27.10.101    172.27.10.1      UDP   53   25913  1.8 M   10   70   1

Summary: total flows: 3937201, total bytes: 316.9 G, total packets: 371.8 M, avg bps: 1.8 M, avg pps: 264, avg bpp: 852
Time window: 2026-03-31 18:32:05 - 2026-04-17 00:34:51, Duration:16d 06:02:46.000
Total records processed: 3937201, passed: 3937201, Blocks skipped: 0, Bytes read: 656856172
Sys: 0.5828s User: 1.4132s Wall: 1.1175s flows/second: 3523285.3 Runtime: 1.1332s
```

Tools :: Analyse :: SQL

- PostgreSQL
- Unterstützt IP-Objekte nativ
- Komplexe Auswertungen möglich
- Materialized Views
- Partitioning

```
CREATE TABLE public.netflows
(
  date timestamp without time zone,
  dur double precision,
  srcip inet,
  source text COLLATE pg_catalog."default",
  dstip inet,
  dest text COLLATE pg_catalog."default",
  port double precision,
  proto text COLLATE pg_catalog."default",
  service text COLLATE pg_catalog."default",
  inpkt bigint,
  inbyte bigint,
  outpkt bigint,
  outbyte bigint,
  flows bigint
) PARTITION BY RANGE (date);

CREATE TABLE public.subnets
(
  netname text COLLATE pg_catalog."default",
  subnet cidr
);
```

Tools :: Analyse :: Subnets

```
CREATE MATERIALIZED VIEW public.servicesbysubnet
AS
 SELECT f.service,
  f.port,
  f.proto,
  snet.netname AS "source net",
  dnet.netname AS "dest net"
 FROM flowbyservice f
   LEFT JOIN subnets dnet ON f.dstip <<= dnet.subnet::inet
   LEFT JOIN subnets snet ON f.srcip <<= snet.subnet::inet
  WHERE dnet.netname IS DISTINCT FROM snet.netname
  GROUP BY f.proto, f.port, f.service, dnet.netname, snet.netname
  ORDER BY f.service, dnet.netname
WITH DATA;
```

```
netflow=> select * from servicesbysubnet where port=53 and proto='udp';
 service | port | proto |   source net    |  dest net
---------+------+-------+-----------------+-------------
 domain  |   53 | udp   | client.demo.at  | srv.demo.at
 domain  |   53 | udp   | dmz.demo.at     | srv.demo.at
 domain  |   53 | udp   | mgmt.demo.at    | srv.demo.at
 domain  |   53 | udp   | openvpn.demo.at | srv.demo.at
 domain  |   53 | udp   | virt.demo.at    | srv.demo.at
 domain  |   53 | udp   | vpn.demo.at     | srv.demo.at
 domain  |   53 | udp   | wlan.demo.at    | srv.demo.at
 domain  |   53 | udp   |                 | srv.demo.at
```

Tools :: Analyse :: Services

```
CREATE MATERIALIZED VIEW public.flowbyservice
AS
 SELECT count(n.flows) AS count,
  n.service,
  n.port,
  n.proto,
  n.srcip,
  n.dstip
 FROM netflows n
 WHERE n.service <> 'UNDEF'::text
 GROUP BY n.proto, n.port, n.service, n.srcip, n.dstip
WITH DATA;
```

```
netflow=> select * from flowbyservice order by count desc limit 10;
  count  |    service     | port  | proto |     srcip     |     dstip
---------+----------------+-------+-------+---------------+---------------
 1624972 | https          |   443 | tcp   | 192.168.24.4  | 192.168.20.45
 1426344 | mysql          |  3306 | tcp   | 192.168.24.16 | 192.168.20.15
 1336735 | zabbix-trapper | 10051 | tcp   | 192.168.24.55 | aa.bb.cc.dd
 1235314 | domain         |    53 | udp   | 192.168.24.55 | 192.168.20.11
 1104969 | https          |   443 | tcp   | 192.168.24.16 | ee.ff.gg.hh
  873579 | https          |   443 | tcp   | 192.168.24.4  | 192.168.20.21
  728732 | https          |   443 | tcp   | 192.168.24.4  | 192.168.20.28
  503770 | ucs-ldap       |  7389 | tcp   | 172.31.0.11   | 192.168.20.11
  492373 | ucs-ldap       |  7389 | tcp   | 192.168.23.15 | 192.168.20.11
  429036 | domain         |    53 | udp   | 192.168.20.11 | 9.9.9.9
```

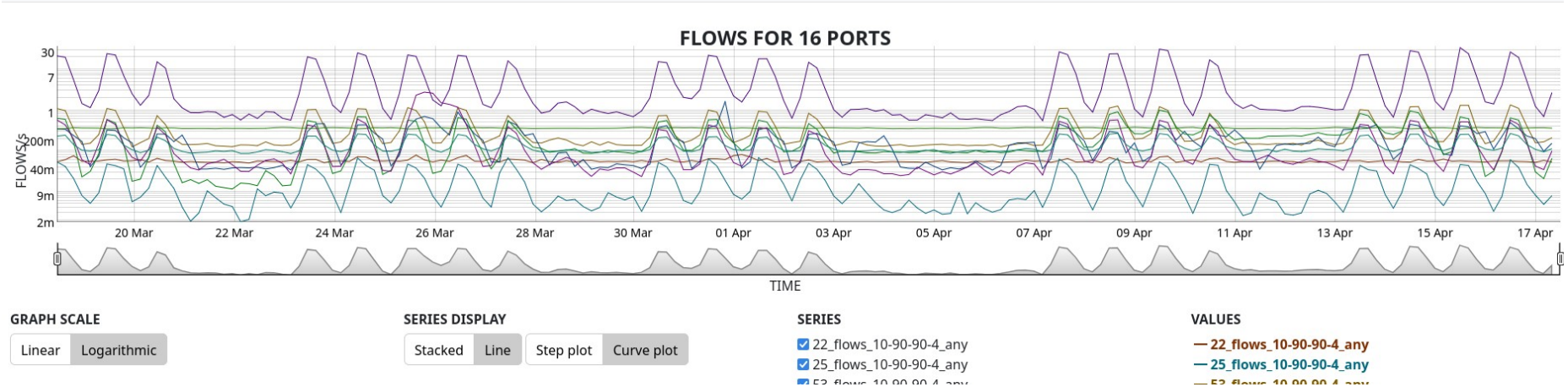
Tools :: Analyse :: WTF?

```
netflow=> SELECT count(flows), port, proto, dstip, dest
FROM netflows WHERE srcip='192.168.24.55'
GROUP BY port, proto, dstip, dest
ORDER BY count(flows) DESC LIMIT 25;
```

count	port	proto	dstip	dest
1336735	10051	tcp	18.196.204.143	ec2-18-196-204-143.eu-central-1.compute.amazonaws.com
1235314	53	udp	192.168.20.11	dc01.demo.at
10795	443	tcp	62.212.166.102	unnamed
1265	443	tcp	84.242.---.---	ip------------.rev.nessus.at
1177	443	tcp	178.128.6.101	repo.zabbix.com
486	80	tcp	151.101.2.132	unnamed
483	80	tcp	151.101.130.132	unnamed
477	80	tcp	151.101.194.132	unnamed
455	80	tcp	151.101.66.132	unnamed
135	443	tcp	185.125.188.54	api.snapcraft.io
130	443	tcp	185.125.188.59	api.snapcraft.io
119	0	icmp	192.168.20.9	watchbox.demo.at
113	443	tcp	185.125.188.58	api.snapcraft.io
108	443	tcp	172.65.32.248	unnamed
102	123	udp	78.41.116.149	time1.funkfeuer.at
100	443	tcp	185.125.188.57	api.snapcraft.io
41	123	udp	91.206.8.34	svn.mediainvent.at
34	123	udp	217.175.196.134	extern3.nemox.net
22	123	udp	152.53.51.182	mn9.tjdev.de
22	53	udp	192.168.20.20	dc02.demo.at
18	123	udp	91.206.8.36	time2.mediainvent.at
16	11000	tcp	192.168.20.37	sesamrdssrv.demo.at
4	443	tcp	91.189.91.101	snapstore-content-cache-3.ps6.canonical.com
3	134	tcp	177.8.133.35	177.8.133-35.jetnetwork.net.br
3	145	tcp	177.8.132.181	177.8.132-181.jetnetwork.net.br

Tools :: Analyse :: GUI

- nfsen-ng (WebUI)
- Observium (paid only)
- ntop (uiuiui ...)
- darkstat (nur lokale Interfaces - Mirror-Port)



Quellen

https://github.com/phaag/nfdump/
https://en.wikipedia.org/wiki/NetFlow
https://www.rfc-editor.org/rfc/rfc7012.html
https://datatracker.ietf.org/wg/ipfix/documents/