Global Network Observability with `goProbe` and `goQuery`

Bärner Go Meetup, 27.03.2025

Lennart Elsen

Fabian Kohn

Observability Team @ Open Systems AG

Lennart Elsen

Systems/Software Engineer at Open Systems

> Observability, Fleet Management, Traffic Analysis, `golang`

Born and raised in Hamburg, Germany

> Zurich, ZH, CH

Surfing, Coffee and Open Source Software

> South Shore Beach, RI, US, Double Espresso (no cream, no sugar), els0r/goProbe

Fabian Kohn

Systems/Software Engineer at Open Systems

> Performance Optimization, High-Energy Physics, Traffic Analysis, `golang`

Born and raised in Göttingen, Germany

> Hamburg, HH, DE

Running, Coffee and Open Source Software

> Everywhere, Flat White, fako1024/slimcap

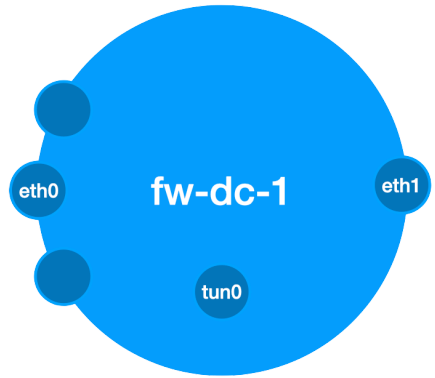
Internet Traffic

Internet Traffic

Customer

Internet Traffic

Open Systems



Customer

What's the traffic composition?

An IP packet

For `t == now`

Live capture

```
tcpdump -ni eth0
```

For `t == now`

Live capture

```
tcpdump -ni eth0
```

Output

```
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
11:33:16.002178 IP 211.154.236.12.35178 > 10.236.2.18.22: Flags [.], ack 188, win 83, options [nop,nop,TS val 515841640
11:33:16.021053 IP 211.154.236.12.35178 > 10.236.2.18.22: Flags [P.], seq 1:37, ack 188, win 83, options [nop,nop,TS va
11:33:16.021268 IP 10.236.2.18.22 > 211.154.236.12.35178: Flags [P.], seq 188:224, ack 37, win 83, options [nop,nop,TS v
```

For `t == now`

Live capture

```
tcpdump -ni eth0
```

What a network engineer looks at

```
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
11:33:16.002178 IP 211.154.236.12.35178 > 10.236.2.18.22: Flags [.], ack 188, win 83, options [nop,nop,TS val 515841640
11:33:16.021053 IP 211.154.236.12.35178 > 10.236.2.18.22: Flags [P.], seq 1:37, ack 188, win 83, options [nop,nop,TS va
11:33:16.021268 IP 10.236.2.18.22 > 211.154.236.12.35178: Flags [P.], seq 188:224, ack 37, win 83, options [nop,nop,TS
```

For `t == now`

Live capture

```
tcpdump -ni eth0
```

What a network engineer looks at

```
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
11:33:16.002178 IP 211.154.236.12.35178 > 10.236.2.18.22: Flags [.], ack 188, win 83, options [nop,nop,TS val 515841640
11:33:16.021053 IP 211.154.236.12.35178 > 10.236.2.18.22: Flags [P.], seq 1:37, ack 188, win 83, options [nop,nop,TS va
11:33:16.021268 IP 10.236.2.18.22 > 211.154.236.12.35178: Flags [P.], seq 188:224, ack 37, win 83, options [nop,nop,TS
```

Bi-directional traffic (SSH session) from 211.154.236.12 to 10.236.2.18

Bi-directional traffic

(SSH session, TCP port 22)

from 211.154.236.12

to 10.236.2.18

For `t == now - 24h`?

Did we run

```
tcpdump -ni eth0 -w eth0.pcap
```

?

For `t == now - 24h`?

Did we run

```
out of disk space
```

?

For `t == now - 24h`?

What if we had captured the metadata?

For `t == now - 24h`?

What if we queried the metadata?

```
query -i eth0 -f -24h -c "dip=10.236.2.18 and sip=211.154.236.12 and dport=22 and proto=tcp" sip,dip,dport,proto
```

For `t == now - 24h`?

What if we queried the metadata?

```
goquery -i eth0 -f -24h -c "dip=10.236.2.18 and sip=211.154.236.12 and dport=22 and proto=tcp" sip,dip,dport,proto

                                          packets  packets             bytes      bytes
             sip          dip  dport  proto      in      out       %        in        out       %
  211.154.236.12  10.236.2.18     22    TCP    481      475    100.00  59.27 kB   65.91 kB  100.00

                                            481      475            59.27 kB   65.91 kB

         Totals:                                    956                     125.18 kB

Timespan    : [2025-03-10 11:47:03, 2025-03-11 11:50:00] (1d3m0s)
Interface   : eth0
Sorted by   : accumulated data volume (sent and received)
Conditions  : (dip = 10.236.2.18 & (sip = 211.154.236.12 & (dport = 22 & proto = tcp)))
Query stats : displayed top 1 hits out of 1 in 9ms
```

Write Path

Read Path

Write Path

- continuous capture of network metadata

Read Path

- a means to query it

Write Path

- continuous capture of network metadata
- low footprint, non-invasive

Read Path

- a means to query it
- low read-latency

Isn’t this a solved problem?

Yes.

It is.

Yes,

BUT

low footprint, non-invasive

low read-latency

`goProbe`

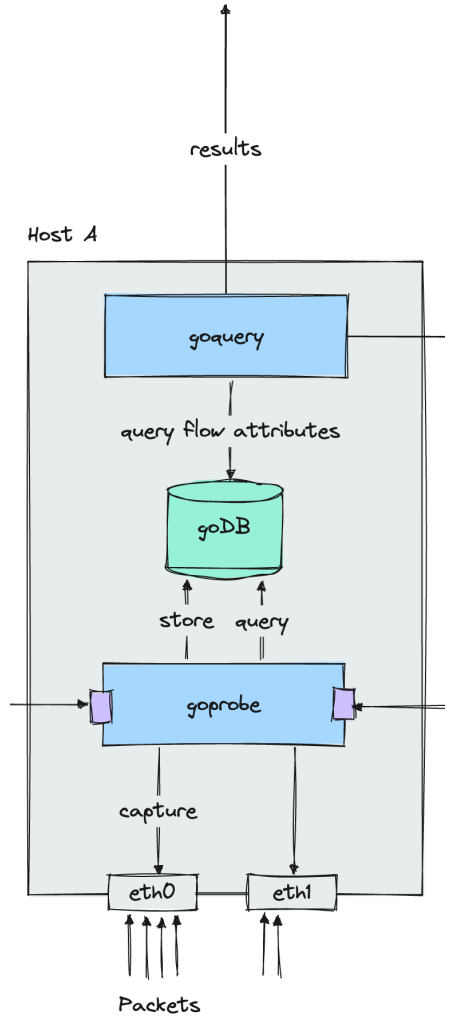
Write Path

- continuous capture of network metadata
- low footprint, non-invasive

`goQuery`

Read Path

- a means to query it
- low read-latency



Capture

`goProbe`

Next-Gen Packet Capture

Previous capture solution (`goProbe` v3):

- Does a lot *[more than we need]* under the hood
- Complex / intricate to use (stateful `pcap` capture handle vs. lots of interfaces)
- `C(GO)` / system library dependency (`libpcap`)
- Customizations / fork required
- Abysmal testing capabilities


FINE - I'LL DO IT MYSELF

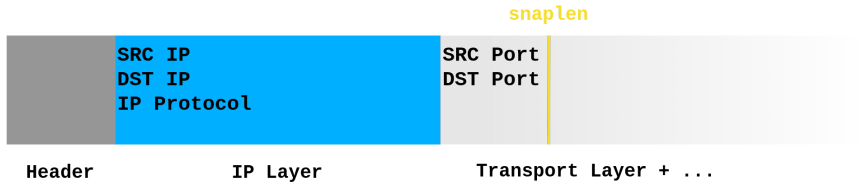
low footprint, non-invasive

low read-latency

Next-Gen Packet Capture

Minimize Overhead:

- IP Layer extraction (if exists)
- Limit to start of transport layer



Linux only (but keep extensible)

Native Go, no external [read: `C(GO)`] dependencies

~~Easy~~ Trivial to use (semi-stateless)

Zero-copy / zero-allocation support

Out-of-the-box tests / benchmarks

Capture Setup

AF_PACKET & MMAP()

Capture Setup

AF_PACKET & MMAP()

Capture Setup

AF_PACKET & MMAP()

Capture Setup

Filtering

BPF[1]: Let kernel do the heavy lifting (filter valid IPv4 / IPv6 packets)

```
// LinkTypeLoopback
// not outbound && (ether proto 0x0800 || ether proto 0x86DD)
var bpfInstructionsLinkTypeLoopback = func(snapLen int) ([]bpf.RawInstruction) {
    return []bpf.RawInstruction{
        {Op: opLDH, Jt: 0x0, Jf: 0x0, K: regPktType},        // Load pktType
        {Op: opJEQ, Jt: 0x4, Jf: 0x0, K: pktTypeOutbound},   // Skip duplicate "OUTBOUND" packets
        {Op: opLDH, Jt: 0x0, Jf: 0x0, K: regEtherType},      // Load byte 12 from the packet (ethernet type)
        {Op: opJEQ, Jt: 0x1, Jf: 0x0, K: etherTypeIPv4},     // Check for IPv4 header
        {Op: opJEQ, Jt: 0x0, Jf: 0x1, K: etherTypeIPv6},     // Check for IPv6 header
        {Op: opRET, Jt: 0x0, Jf: 0x0, K: uint32(snapLen)},   // Return up to snapLen bytes of the packet
        {Op: opRET, Jt: 0x0, Jf: 0x0, K: 0x0},               // Return (no data)
    }
}
```

---

https://en.wikipedia.org/wiki/Berkeley_Packet_Filter ↩

Interfaces

```
// Source denotes a generic packet capture source
type Source interface {
    NextPacket(pBuf Packet) (Packet, error)
    NextPayload(pBuf []byte) ([]byte, byte, uint32, error)
    NextIPPacket(pBuf IPLayer) (IPLayer, PacketType, uint32, error)
    NextPacketFn(func(payload []byte, totalLen uint32, pktType PacketType, ipLayerOffset byte) error) error

    Stats() (Stats, error)
    Close() error
    // ...
}
```

Interfaces

```
// SourceZeroCopy denotes a capture source that supports zero-copy operations
type SourceZeroCopy interface {
    NextPayloadZeroCopy() ([]byte, PacketType, uint32, error)
    NextIPPacketZeroCopy() (IPLayer, PacketType, uint32, error)

    // Wrap generic Source
    Source
}
```

Minimal Example

```
src, err := afring.NewSource(
    “enp1s0”,
    afring.CaptureLength(link.CaptureLengthMinimalIPv4Transport),
    afring.BufferSize(
        1024*1024,       // Block Size
        4,               // Number of Blocks
    ),
)
if err != nil {
    // Error handling
}
```

Minimal Example

```
src, err := afring.NewSource(
    “enp1s0”,
    afring.CaptureLength(link.CaptureLengthMinimalIPv4Transport),
    afring.BufferSize(
        1024*1024,       // Block Size
        4,               // Number of Blocks
    ),
)
if err != nil {
    // Error handling
}
```

Minimal Example

```
src, err := afring.NewSource(
    “enp1s0”,
    afring.CaptureLength(link.CaptureLengthMinimalIPv4Transport),
    afring.BufferSize(
        1024*1024,       // Block Size
        4,               // Number of Blocks
    ),
)
if err != nil {
    // Error handling
}
```

Minimal Example (cont'd)

```
for {
    ipLayer, pktType, pktLen, err := src.NextIPPacketZeroCopy()
    if err != nil {
        if errors.Is(err, capture.ErrCaptureStopped) {
            // Graceful stop
            break
        }
        // Error handling
    }
}
```

Minimal Example (cont'd)

```
for {
    ipLayer, pktType, pktLen, err := src.NextIPPacketZeroCopy()
    if err != nil {
        if errors.Is(err, capture.ErrCaptureStopped) {
            // Graceful stop
            break
        }
        // Error handling
    }

    // Do stuff ...
    _ = ipLayer        // Raw IP layer data (up to snaplen)
    _ = pktType        // Packet Type (direction flag)
    _ = pktLen         // Total packet length
}
```

Integration

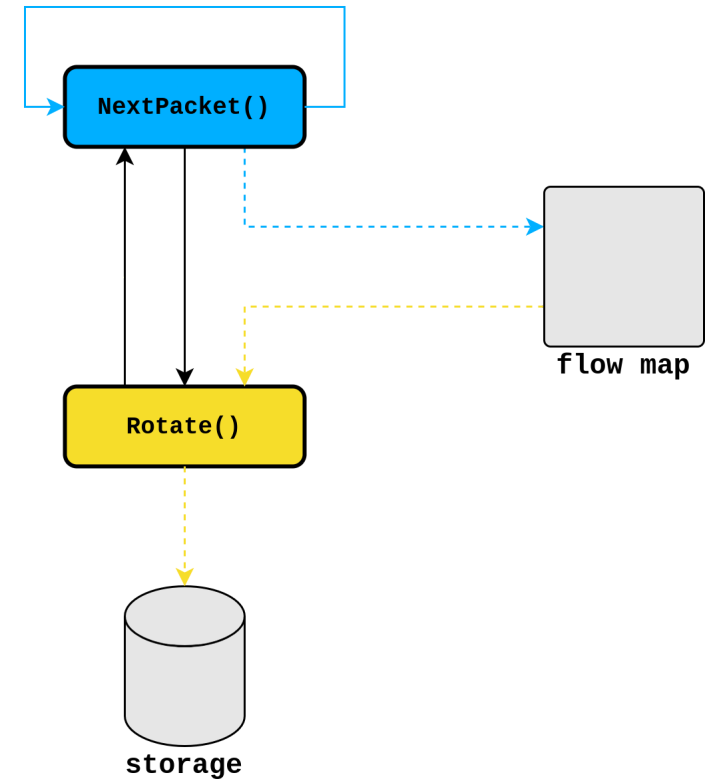
Capture Rotation

During data writeout (flow map “rotation”) in `goProbe`:

- Fundamentally concurrency-safe read / write
  Permanent overhead

  OR

- Interrupt capture during rotation
  Potential ring buffer overflow



Integration

Capture Rotation

During data writeout (flow map “rotation”) in `goProbe`:

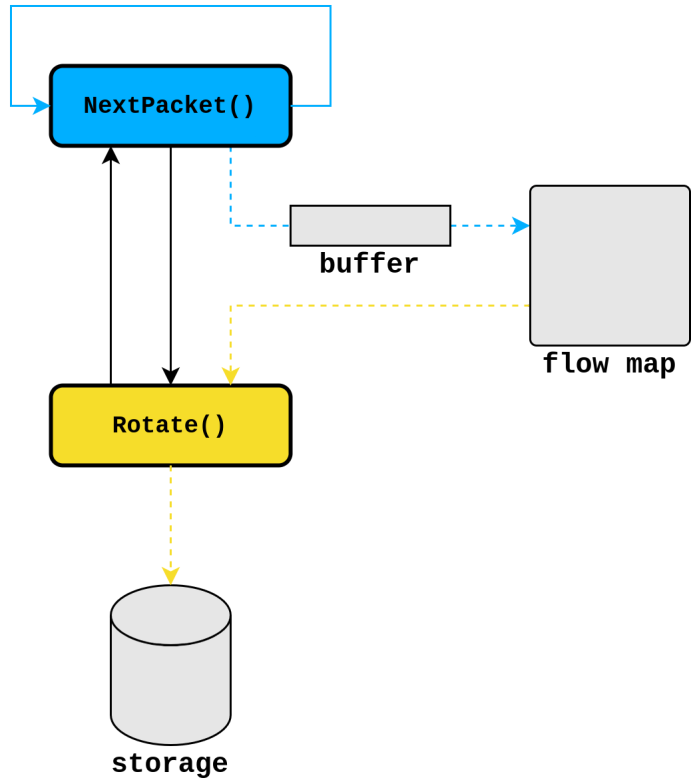
- Fundamentally concurrency-safe read / write
  Permanent overhead

  OR

- Interrupt capture during rotation
  Potential ring buffer overflow

Mitigation:

- Sequential rotation of interfaces
- Additional (shared) local buffer
- Packet processing & parsing while buffering

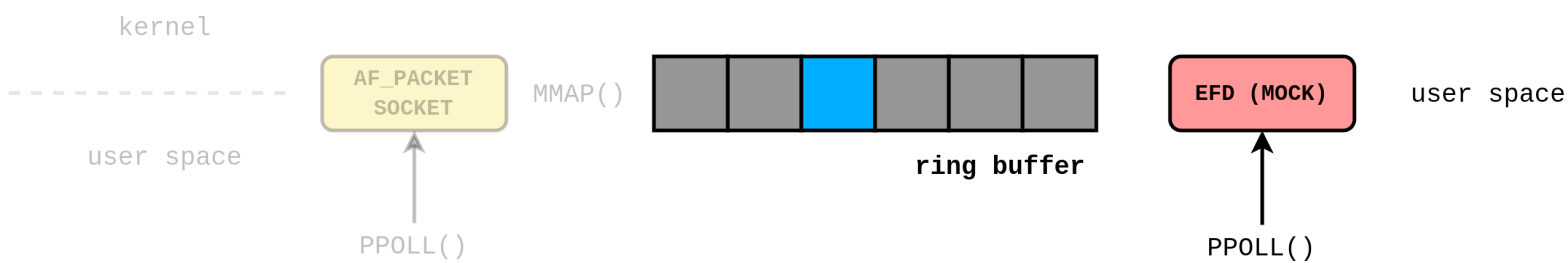


Testing

Mock Capture Sources

Stand-in wrappers (down to socket interaction) around actual sources:

- `AF_PACKET` socket vs. simple FD / EFD semaphore
- `MMAP` 'ed area vs. user space slice
- Memory barrier vs. atomic status flag / field



Testing

Mock Capture Sources

```
src, err := afring.NewSource(
    “enp1s0”,
    afring.CaptureLength(link.CaptureLengthMinimalIPv4Transport),
    afring.BufferSize(
        1024*1024,       // Block Size
        4,               // Number of Blocks
    ),
)
```

Testing

Mock Capture Sources

```
src, err := afring.NewMockSource(
    “enp1s0”,
    afring.CaptureLength(link.CaptureLengthMinimalIPv4Transport),
    afring.BufferSize(
        1024*1024,       // Block Size
        4,               // Number of Blocks
    ),
)
```

Testing

Mock Capture Sources

Stand-in wrappers (down to socket interaction) around actual sources:

- `AF_PACKET` socket vs. simple FD / EFD semaphore
- `MMAP`'ed area vs. user space slice
- Memory barrier vs. atomic status flag / field

Features:

- Reading & replay of pcap dumps (no timing)
- Synthetic packet / payload generation
- No privileges (or *actual* interfaces)
- Piping from other mock sources
- High-throughput mode (benchmarks)

Testing

Benchmarks

Testbed: Production Host (DC Firewall)

Scenario: 1h Real-life capture `goProbe` v3 (`gopacket`) / v4 (`slimcap`), 676.9 M packets

CPU Time:	43:17.7 min	vs.	1:57.7 min	~ x22
Peak Mem Usage:	150 MiB	vs.	153 MiB	~ x1
Dropped Packets:	98 k	vs.	0	

Read Path

`goQuery`

Local Queries

A `Row` of metadata

Load only what you need

```
goquery -i eth0 -f -24h -c "sip=211.154.236.12 and dport=22 and proto=tcp" dip
```

Local Queries

A `Row` of metadata

Load only what you need

```
goquery -i eth0

eth0/
```

Local Queries

A `Row` of metadata

Load only what you need

```
goquery -i eth0 -f -24h

eth0/
  2025/03/1742860800
    sip.gpf
    dip.gpf
    dport.gpf
    proto.gpf

    bytes_rcvd.gpf
    bytes_sent.gpf
    pkts_rcvd.gpf
    pkts_sent.gpf
```

Local Queries

A `Row` of metadata

Load only what you need

```
goquery -i eth0 -f -24h dip

eth0/
  2025/03/1742860800
    sip.gpf
    dip.gpf
    dport.gpf
    proto.gpf

    bytes_rcvd.gpf
    bytes_sent.gpf
    pkts_rcvd.gpf
    pkts_sent.gpf
```

Local Queries

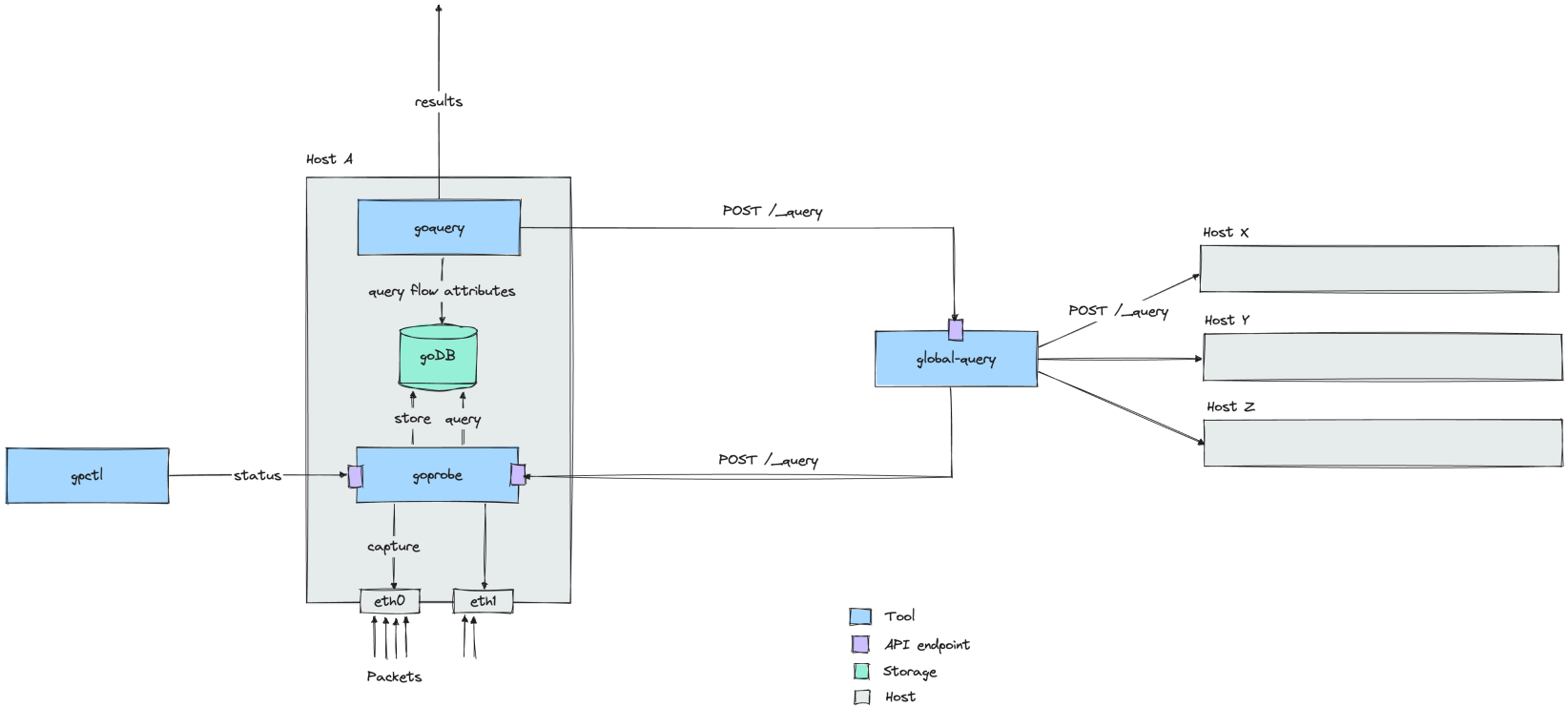
A `Row` of metadata

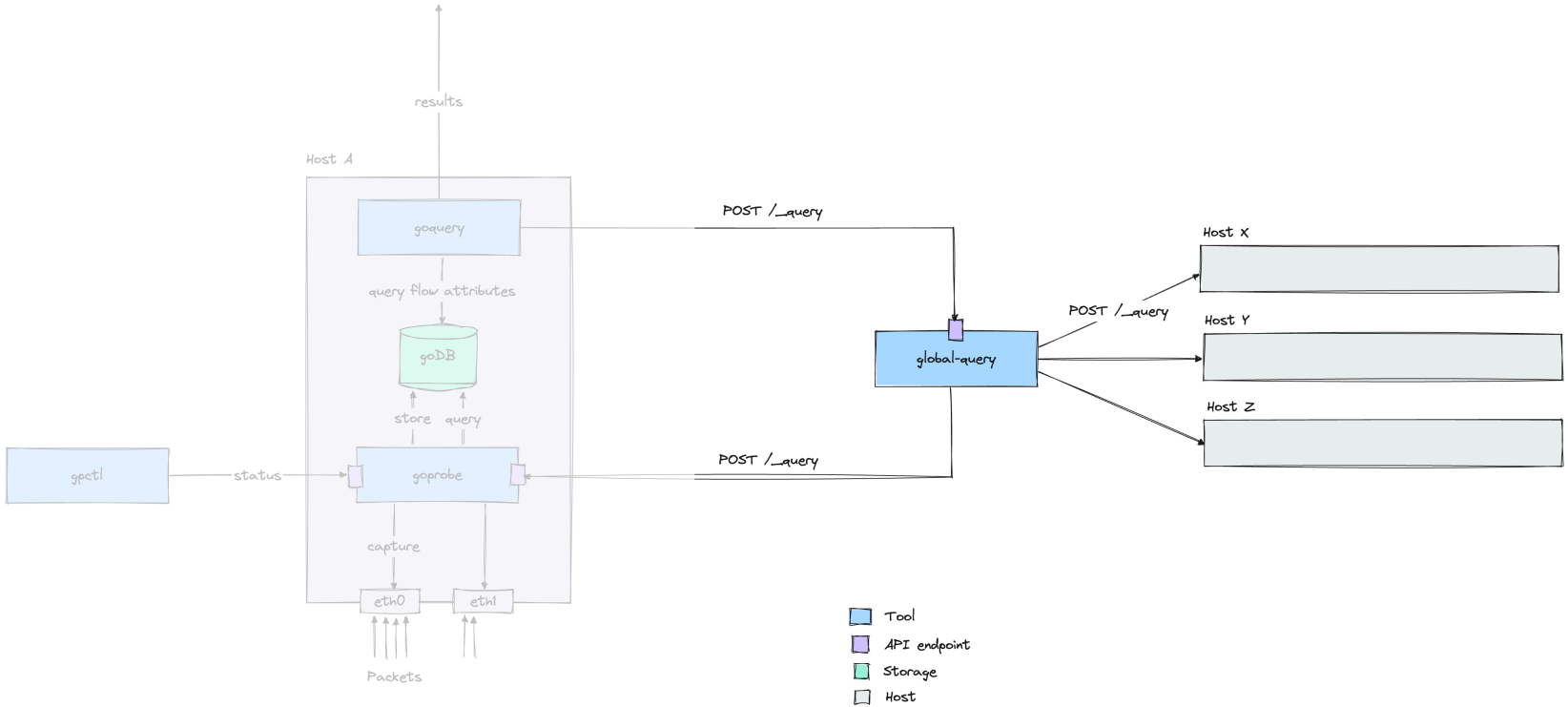
Load only what you need

```
goquery -i eth0 -f -24h -c "sip=211.154.236.12 and dport=22 and proto=tcp" dip

eth0/
  2025/03/1742860800
    sip.gpf
    dip.gpf
    dport.gpf
    proto.gpf

    bytes_rcvd.gpf
    bytes_sent.gpf
    pkts_rcvd.gpf
    pkts_sent.gpf
```

```
(SSH session, TCP port 22)

   from 211.154.236.12

        on hostA
```

```
(SSH session, TCP port 22)

    from 211.154.236.12

        on all hosts
```


results
Host A
goquery
query flow attributes
goDB
store
query
gpctl
status
goprobe
capture
eth0
eth1
Packets
POST /_query
POST /_query
POST /_query
global-query
Host X
Host Y
Host Z
Tool
API endpoint
Storage
Host


results
Host A
goquery
query flow attributes
goDB
store
query
gpctl
status
goprobe
capture
eth0
eth1
Packets
POST /_query
POST /_query
POST /_query
global-query
Host X
Host Y
Host Z
Tool
API endpoint
Storage
Host

Global Queries

A `Row` of metadata

```
type Row struct {
    // Labels
    Timestamp   time.Time
    Iface       string
    Hostname    string
    HostID      string
}
```

Global Queries

A `Row` of metadata

```
type Row struct {
    // Labels
    Timestamp   time.Time
    Iface       string
    Hostname    string
    HostID      string

    // Attributes
    SrcIP       netip.Addr
    DstIP       netip.Addr
    IPProto     uint8
    DstPort     uint16
}
```

Global Queries

A `Row` of metadata

```
type Row struct {
    // Labels
    Timestamp   time.Time
    Iface       string
    Hostname    string
    HostID      string

    // Attributes
    SrcIP       netip.Addr
    DstIP       netip.Addr
    IPProto     uint8
    DstPort     uint16

    // Counters
    BytesRcvd   uint64
    BytesSent   uint64
    PacketsRcvd uint64
    PacketsSent uint64
}
```

Global Queries

A `Row` of metadata

```
type Row struct {
    // Labels are the partition Attributes
    Labels Labels

    // Attributes which can be grouped by
    Attributes Attributes

    // Counters for bytes/packets
    Counters types.Counters
}
```

Global Queries

A `Row` of metadata

```
type Row struct {
    MergeableAttributes

    // Counters for bytes/packets
    Counters types.Counters
}
```

Global Queries

A first class data structure for flow aggregation

```
// RowsMap is an aggregated representation of a Rows list
type RowsMap map[MergeableAttributes]types.Counters
```

Which systems did `211.154.236.12` access via SSH?

Which systems did `211.154.236.12` access via SSH?

```
goquery -i eth0 -f -24h -c "sip=211.154.236.12 and dport=22 and proto=tcp" dip
```

Which systems did `211.154.236.12` access via SSH?

```
goquery -i eth0 -f -24h -c "sip=211.154.236.12 and dport=22 and proto=tcp" dip -q hostA,hostB,...,hostK
```

Which systems did `211.154.236.12` access via SSH?

```
goquery -i eth0 -f -24h -c "sip=211.154.236.12 and dport=22 and proto=tcp" dip -q hostA,hostB,...,hostK

                                  packets   packets             bytes      bytes
                host           dip       in       out      %         in        out      %
           acme-sg-1   10.236.2.19   55.11 k   56.80 k  32.01    7.30 MB   17.92 MB  18.16
           acme-sg-2   10.236.2.20   69.14 k   62.46 k  37.65    6.46 MB    7.57 MB  10.10
    acme-node4-test-1  10.236.50.32    7.25 k    4.54 k   3.37   10.92 MB  731.57 kB   8.37
    acme-node1-test-1  10.236.50.30    5.82 k    1.75 k   2.16   11.38 MB  159.01 kB   8.31
    acme-node2-test-1   10.236.50.2    5.08 k    1.36 k   1.84   11.34 MB  133.30 kB   8.25
    acme-node3-test-1  10.236.50.31    3.70 k    1.08 k   1.37   10.60 MB  115.16 kB   7.71
           acme-sg-6  10.236.50.17    5.04 k    1.82 k   1.96    9.70 MB  189.60 kB   7.12
   acme-node4-ch-zh-1  10.236.48.26    4.85 k    1.09 k   1.70    9.71 MB  145.36 kB   7.09
   acme-node3-ch-zh-1   10.236.48.7    5.38 k    1.07 k   1.84    9.70 MB  111.05 kB   7.06
   acme-node1-ch-zh-1   10.236.48.5    3.96 k    1.10 k   1.45    9.64 MB  141.61 kB   7.04
           acme-sg-3   10.236.50.7    3.61 k    1.37 k   1.42    9.58 MB  131.51 kB   6.99
           acme-sg-4   10.236.2.59   19.89 k   17.63 k  10.73    1.97 MB    2.38 MB   3.13
           acme-sg-5   10.236.2.18    4.53 k    4.17 k   2.49  459.46 kB  495.22 kB   0.67

                                  193.32 k  156.25 k         108.75 MB   30.17 MB

             Totals:                        349.57 k                   138.92 MB
```

Which systems did `211.154.236.12` access via SSH?

```
goquery -i eth0 -f -24h -c "sip=211.154.236.12 and dport=22 and proto=tcp" dip -q hostA,hostB,...,hostK

Timespan          : [2025-03-24 10:33:35, 2025-03-25 10:35:00] (1d1m0s)
Interface / Hosts : eth0 on 34 hosts: 25 ok / 0 empty / 9 error
Query stats       : displayed top 13 hits out of 13 in 10.196s
Trace ID          : c7c51c6e5c463716cedcb69bd40a36e4
```

Which systems did `211.154.236.12` access via SSH?

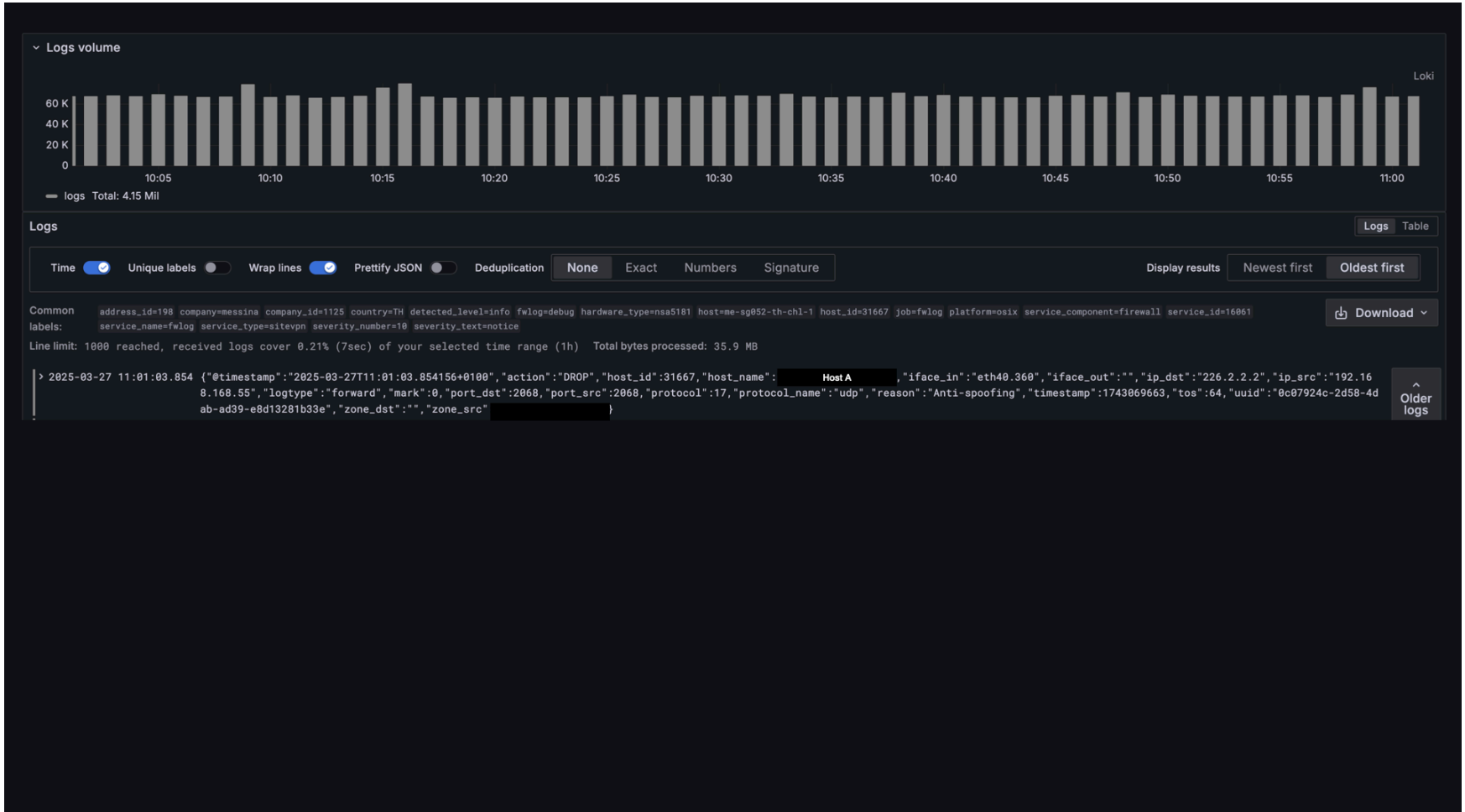
```
goquery -i eth0 -f -24h -c "sip=211.154.236.12 and dport=22 and proto=tcp" dip -q hostA,hostB,...,hostK

Timespan          : [2025-03-24 10:33:35, 2025-03-25 10:35:00] (1d1m0s)
Interface / Hosts : eth0 on 34 hosts: 25 ok / 0 empty / 9 error
Query stats       : displayed top 13 hits out of 13 in 10.196s
Trace ID          : c7c51c6e5c463716cedcb69bd40a36e4
```

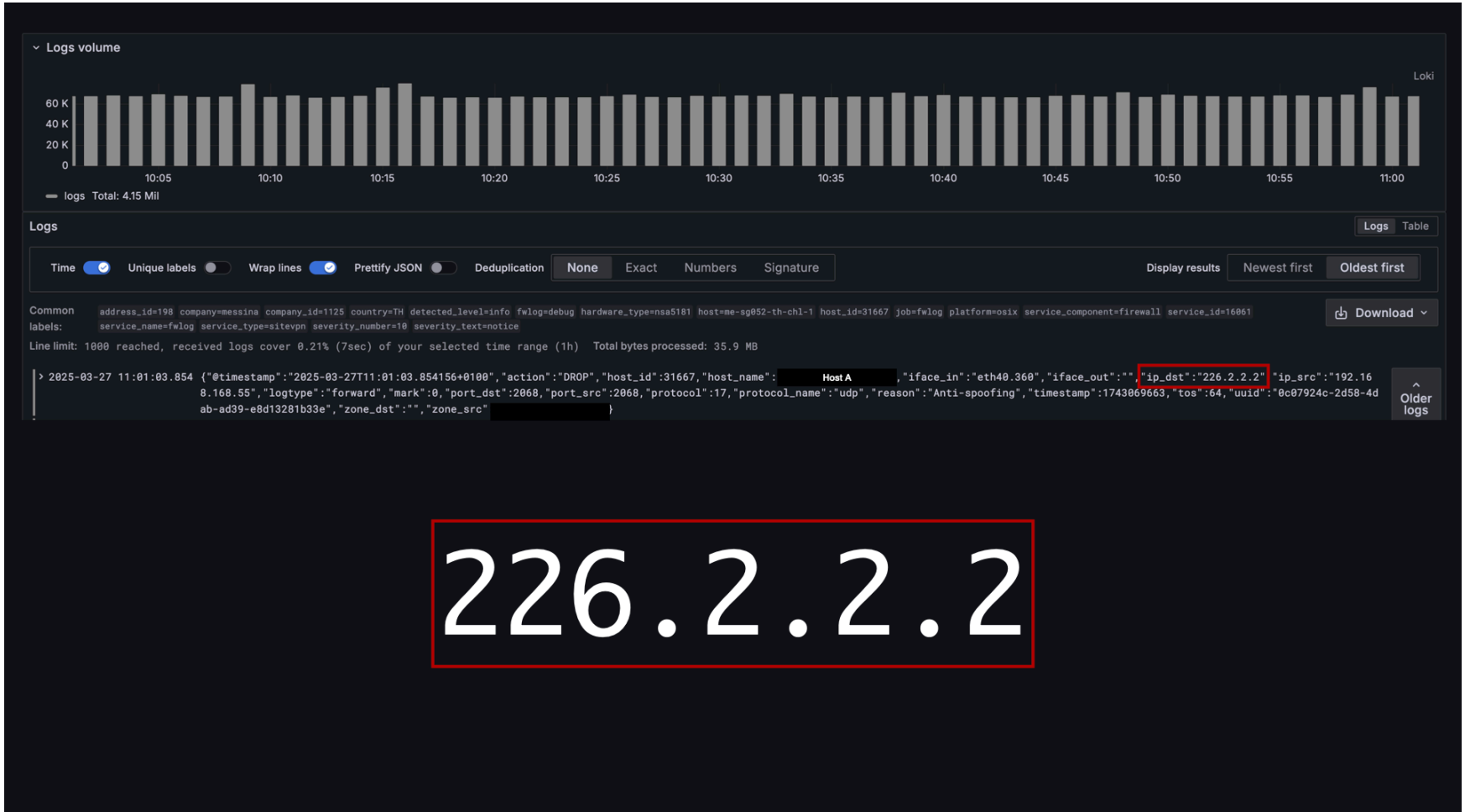
Global Network Observability

Global Network Observability

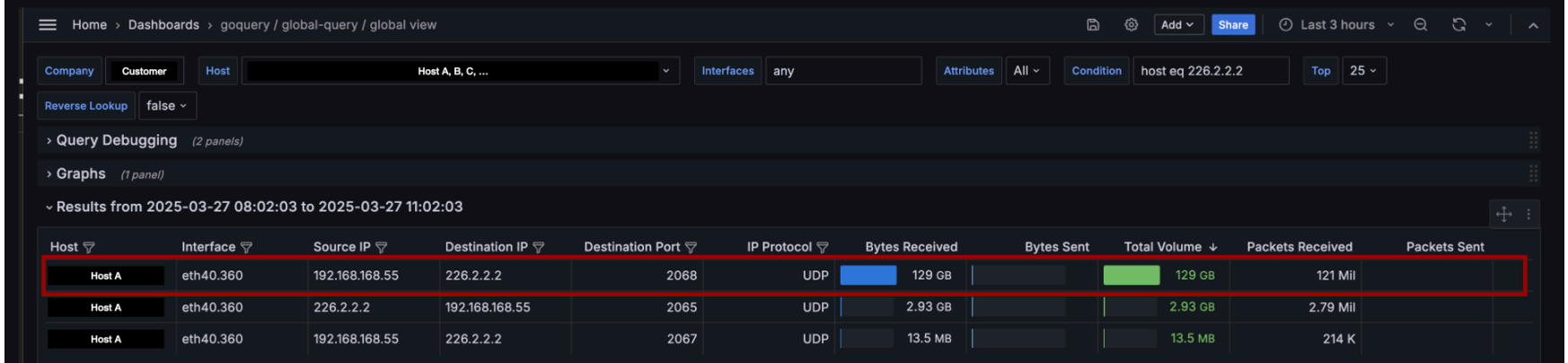
Troubleshooting

- historic traffic patterns across the WAN path
  - did `A` access `B`?
  - did `A` get blocked?
  - is there asymmetric routing?
- biggest bandwidth hogs in WAN fleet


Logs volume
Loki
60 K
40 K
20 K
0
10:05
10:10
10:15
10:20
10:25
10:30
10:35
10:40
10:45
10:50
10:55
11:00
logs Total: 4.15 Mil
Logs
Logs
Table
Time
Unique labels
Wrap lines
Prettify JSON
Deduplication
None
Exact
Numbers
Signature
Display results
Newest first
Oldest first
Common labels:
address_id=198 company=messina company_id=1125 country=TH detected_level=info fwlog=debug hardware_type=nsa5181 host=me-sg052-th-chl-1 host_id=31667 job=fwlog platform=osix service_component=firewall service_id=16061 service_name=fwlog service_type=sitevpn severity_number=10 severity_text=notice
Download
Line limit: 1000 reached, received logs cover 0.21% (7sec) of your selected time range (1h) Total bytes processed: 35.9 MB
> 2025-03-27 11:01:03.854 {"@timestamp":"2025-03-27T11:01:03.854156+0100","action":"DROP","host_id":31667,"host_name":
Host A
,"iface_in":"eth40.360","iface_out":"","ip_dst":"226.2.2.2","ip_src":"192.168.168.55","logtype":"forward","mark":0,"port_dst":2068,"port_src":2068,"protocol":17,"protocol_name":"udp","reason":"Anti-spoofing","timestamp":1743069663,"tos":64,"uuid":"0c07924c-2d58-4dab-ad39-e8d13281b33e","zone_dst":"","zone_src"
Older logs


Logs volume
Loki
60 K
40 K
20 K
0
10:05
10:10
10:15
10:20
10:25
10:30
10:35
10:40
10:45
10:50
10:55
11:00
logs Total: 4.15 Mil
Logs
Logs
Table
Time
Unique labels
Wrap lines
Prettify JSON
Deduplication
None
Exact
Numbers
Signature
Display results
Newest first
Oldest first
Common labels:
address_id=198 company=messina company_id=1125 country=TH detected_level=info fwlog=debug hardware_type=nsa5181 host=me-sg052-th-chl-1 host_id=31667 job=fwlog platform=osix service_component=firewall service_id=16061 service_name=fwlog service_type=sitevpn severity_number=10 severity_text=notice
Download
Line limit: 1000 reached, received logs cover 0.21% (7sec) of your selected time range (1h) Total bytes processed: 35.9 MB
2025-03-27 11:01:03.854 {"@timestamp":"2025-03-27T11:01:03.854156+0100","action":"DROP","host_id":31667,"host_name":
Host A
,"iface_in":"eth40.360","iface_out":"","ip_dst":"226.2.2.2","ip_src":"192.168.168.55","logtype":"forward","mark":0,"port_dst":2068,"port_src":2068,"protocol":17,"protocol_name":"udp","reason":"Anti-spoofing","timestamp":1743069663,"tos":64,"uuid":"0c07924c-2d58-4dab-ad39-e8d13281b33e","zone_dst":"","zone_src"
Older logs

226.2.2.2


Home › Dashboards › goquery / global-query / global view
Add
Share
Last 3 hours
Company
Customer
Host
Host A, B, C, ...
Interfaces
any
Attributes
All
Condition
host eq 226.2.2.2
Top
25
Reverse Lookup
false
› Query Debugging (2 panels)
› Graphs (1 panel)
› Results from 2025-03-27 08:02:03 to 2025-03-27 11:02:03
Host
Interface
Source IP
Destination IP
Destination Port
IP Protocol
Bytes Received
Bytes Sent
Total Volume
Packets Received
Packets Sent
Host A
eth40.360
192.168.168.55
226.2.2.2
2068
UDP
129 GB
129 GB
121 Mil
Host A
eth40.360
226.2.2.2
192.168.168.55
2065
UDP
2.93 GB
2.93 GB
2.79 Mil
Host A
eth40.360
192.168.168.55
226.2.2.2
2067
UDP
13.5 MB
13.5 MB
214 K

Global Network Observability

Troubleshooting

- historic traffic patterns across the WAN path
  - did `A` access `B`?
  - did `A` get blocked?
  - is there asymmetric routing?
- biggest bandwidth hogs in WAN fleet

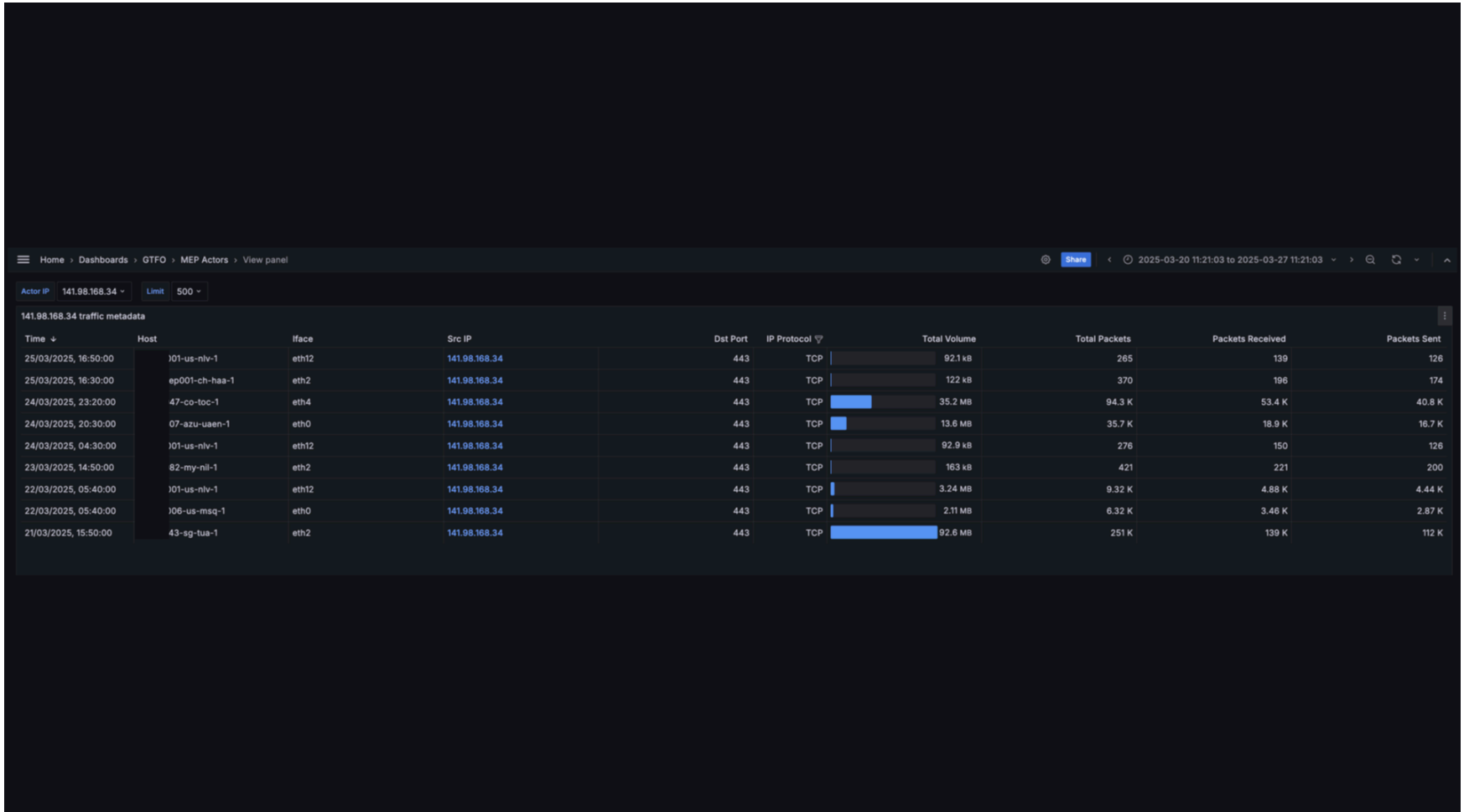
Security

- am I affected?
- has malicious actor with IP `x.y.z.a` accessed any systems?


MEP Blocked or MEP Rejected
1200
1000
800
600
400
200
0
03/20 12:00
03/20 18:00
03/21 00:00
03/21 06:00
03/21 12:00
03/21 18:00
03/22 00:00
03/22 06:00
03/22 12:00
03/22 18:00
03/23 00:00
03/23 06:00
03/23 12:00
03/23 18:00
03/24 00:00
03/24 06:00
03/24 12:00
03/24 18:00
03/25 00:00
03/25 06:00
03/25 12:00
03/25 18:00
03/26 00:00
03/26 06:00
03/26 12:00
03/26 18:00
03/27 00:00
03/27 06:00
blocked
rejected
accepted

Actors

ip	accepted	rejected	blocked	hosts	companies
141.98.168.34	0	31	31	21	5


Home › Dashboards › GTFO › MEP Actors › View panel
Share
2025-03-20 11:21:03 to 2025-03-27 11:21:03
Actor IP
141.98.168.34
Limit
500
141.98.168.34 traffic metadata
Time
Host
Iface
Src IP
Dst Port
IP Protocol
Total Volume
Total Packets
Packets Received
Packets Sent
25/03/2025, 16:50:00
)01-us-nlv-1
eth12
141.98.168.34
443
TCP
92.1 kB
265
139
126
25/03/2025, 16:30:00
ep001-ch-haa-1
eth2
141.98.168.34
443
TCP
122 kB
370
196
174
24/03/2025, 23:20:00
47-co-toc-1
eth4
141.98.168.34
443
TCP
35.2 MB
94.3 K
53.4 K
40.8 K
24/03/2025, 20:30:00
07-azu-uaen-1
eth0
141.98.168.34
443
TCP
13.6 MB
35.7 K
18.9 K
16.7 K
24/03/2025, 04:30:00
)01-us-nlv-1
eth12
141.98.168.34
443
TCP
92.9 kB
276
150
126
23/03/2025, 14:50:00
82-my-nil-1
eth2
141.98.168.34
443
TCP
163 kB
421
221
200
22/03/2025, 05:40:00
)01-us-nlv-1
eth12
141.98.168.34
443
TCP
3.24 MB
9.32 K
4.88 K
4.44 K
22/03/2025, 05:40:00
)06-us-msq-1
eth0
141.98.168.34
443
TCP
2.11 MB
6.32 K
3.46 K
2.87 K
21/03/2025, 15:50:00
43-sg-tua-1
eth2
141.98.168.34
443
TCP
92.6 MB
251 K
139 K
112 K

Who is `141.98.168.34`?

```
dig -x 141.98.168.34 +short
```

Who is `141.98.168.34`?

```
dig -x 141.98.168.34 +short
vm3783913.stark-industries.solutions.
```

Who is `141.98.168.34`?

```
dig -x 141.98.168.34 +short
vm3783913.stark-industries.solutions.
```

> “Two weeks before Russia invaded Ukraine in February 2022, a large, mysterious new Internet hosting firm called Stark Industries materialized and quickly became the epicenter of massive distributed denial-of-service (DDoS) attacks on government and commercial targets in Ukraine and Europe. An investigation into Stark Industries reveals it is being used as a global proxy network that conceals the true source of cyberattacks and disinformation campaigns against enemies of Russia.” [1]

Thank You

Want to Contribute?

- open an issue on github.com/els0r/goProbe
- PRs welcome. See `good first issue`

Backup

Testing

Benchmarks

Testbed: Quad-core Odroid H3, 32 GiB RAM

Scenario: Synthetic mock benchmark (zero-copy packet retrieval) on `slimcap`

Time / op:	16.2 ns	± 1%
Throughput:	61.7 Mpps	± 1%
Allocations / op:	0	