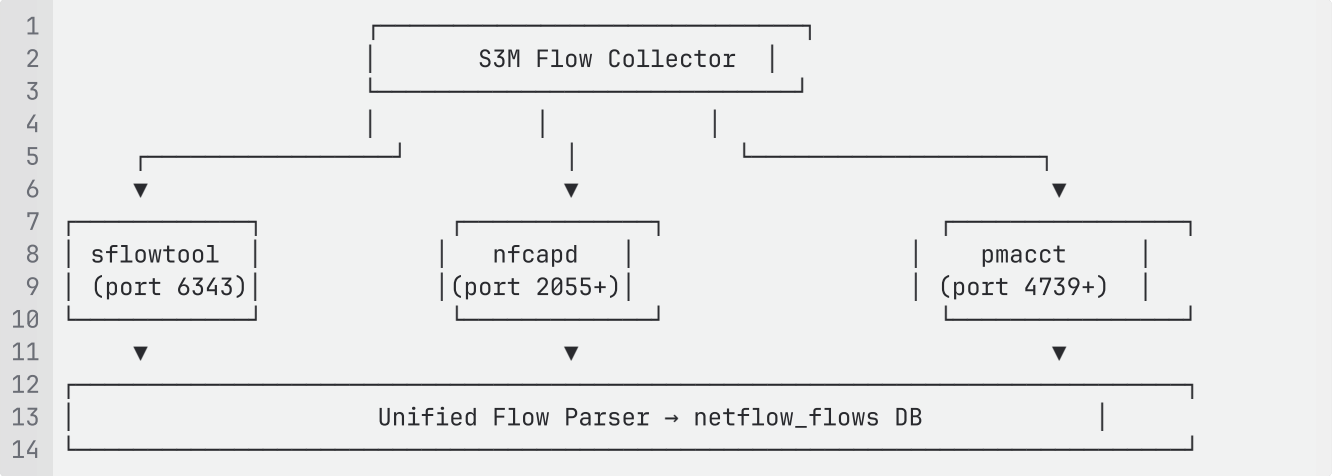
S3M Unified Flow Listener Architecture

Supported Protocols

Protocol	Vendors	Collector Tool
sFlow	Ruijie, HP, Alcatel	`sflowtool`
NetFlow v5/v9	Cisco, Fortinet, Juniper	`nfcapd` from `nfdump`
IPFIX	Palo Alto, Huawei, newer routers	`pmacct` or `logstash`

Architecture Diagram



Unified Flow Table Schema

```
CREATE TABLE netflow_flows (
    id SERIAL PRIMARY KEY,
    timestamp TIMESTAMPTZ DEFAULT now(),
    src_ip TEXT,
    dst_ip TEXT,
    src_mac TEXT,
    dst_mac TEXT,
    protocol TEXT,
    src_port INTEGER,
    dst_port INTEGER,
    byte_count INTEGER,
    ingress_interface INTEGER,
    egress_interface INTEGER,
    source_type TEXT, -- 'sflow', 'netflow', 'ipfix'
    device_ip TEXT,
    raw_sample JSONB
);
```

To-do List

[ ] Build a listener for each protocol

- `sflowtool` for Sflow
- `nfcapd` setup for NetFlow v5/v9
- `pmacct` or `logstash` for IPFIX

[ ] Normalize output into unified format (per protocol)

- Write adapters/parsers that output the shared structure above

[ ] Add correlation/enrichment layer

- Map `src_mac` to known endpoints via ARP, RADIUS, or SNMP
- Map `dst_ip` to threat feeds / known services

[ ] Enable filtering in UI

- By device, port, protocol, volume, time, source_type

S3M Passive Flow Telemetry via sFlow

Purpose

This document outlines the architecture, configuration, integration, and value proposition of using sFlow for passive traffic telemetry within the S3M Network Access Control (NAC) platform.

sFlow provides device-agnostic, agentless visibility into traffic behavior by capturing statistical samples of network flows and interface counters. This capability enables S3M to passively detect and analyze device behavior, profile endpoints, and identify anomalies—without requiring port mirroring, inline deployment, or endpoint agents.

Architecture Overview

```
[Ruijie Switch (sFlow Exporter)]
        → UDP 6343
        → [S3M Listener Node]
             → sflowtool (listener)
             → Log Parser
             → PostgreSQL (netflow_flows)
             → Web Interface / Analytics
```

Switch Configuration (Ruijie – Example)

Enable sFlow Globally

```
sflow agent address <ip-address-s3m-knows>
sflow collector 1 destination <S3M-ip-address> 6343
```

> Alternatively:
>
> ```
> sflow agent interface vlan1 # if not sflow agent ip <ip-address-s3m-knows>
> ```

Per-Interface sFlow Activation

```
interface GigabitEthernet 0/17
 sflow counter collector 1
 sflow flow collector 1
 sflow enable
```

Installing sFlow Listener on S3M

```
# Clone the source
git clone https://github.com/sflow/sflowtool.git
cd sflowtool

# Build it
./boot.sh
./configure
make
sudo make install
```

Sample Data Examples (sflowtool -p 6343)

COUNTERSSAMPLE (Interface Statistics)

```
startDatagram =================================
datagramSourceIP 10.34.10.15
datagramSize 204
unixSecondsUTC 1750319487
unixSecondsUTC_uS 421950
localtime 2025-06-19T10:51:27+0300
datagramVersion 5
agentSubId 0
agent 10.34.10.15
packetSequenceNo 35
sysUpTime 2919388856
samplesInPacket 1
startSample ----------------------
sampleType_tag 0:2
sampleType COUNTERSSAMPLE
sampleSequenceNo 79
sourceId 0:17
counterBlock_tag 0:1
ifIndex 17
networkType 6
ifSpeed 100000000
ifDirection 1
ifStatus 3
ifInOctets 204079871
ifInUcastPkts 1973749
ifInMulticastPkts 17937
```

```
ifInBroadcastPkts 6033
ifInDiscards 0
ifInErrors 15
ifInUnknownProtos 0
ifOutOctets 5558259298
ifOutUcastPkts 3858031
ifOutMulticastPkts 79230
ifOutBroadcastPkts 170566
ifOutDiscards 0
ifOutErrors 0
ifPromiscuousMode 2
counterBlock_tag 0:2
dot3StatsAlignmentErrors 6
dot3StatsFCSErrors 13
dot3StatsSingleCollisionFrames 0
dot3StatsMultipleCollisionFrames 0
dot3StatsSQETestErrors 0
dot3StatsDeferredTransmissions 0
dot3StatsLateCollisions 0
dot3StatsExcessiveCollisions 0
dot3StatsInternalMacTransmitErrors 0
dot3StatsCarrierSenseErrors 23
dot3StatsFrameTooLongs 0
dot3StatsInternalMacReceiveErrors 0
dot3StatsSymbolErrors 4294967295
endSample   ----------------------
endDatagram   =================================
```

FLOWSAMPLE (Traffic Snapshot)

```
startDatagram =================================
datagramSourceIP 10.34.10.15
datagramSize 156
unixSecondsUTC 1750319619
unixSecondsUTC_uS 739315
localtime 2025-06-19T10:53:39+0300
datagramVersion 5
agentSubId 0
agent 10.34.10.15
packetSequenceNo 40
sysUpTime 2919521176
samplesInPacket 1
startSample ----------------------
sampleType_tag 0:1
sampleType FLOWSAMPLE
sampleSequenceNo 6
sourceId 0:17
meanSkipCount 8192
samplePool 40960
dropEvents 0
inputPort 17
outputPort 0
flowBlock_tag 0:1
flowSampleType HEADER
headerProtocol 1
sampledPacketSize 256
strippedBytes 4
headerLen 64
```

```
headerBytes 00-0C-29-57-BE-2C-00-E0-4C-36-AD-17-08-00-45-00-00-EE-00-00-40-00-40-11-15-69-0A-22-0A-65-08-08-08-08-DD-7B-01-BB-00-DA-DD-04-54-E3-0F-38-8F-2B-F8-1A-CD-6E-30-98-13-3E-59-9B-F0-83-73-B4-7F-29
dstMAC 000c2957be2c
srcMAC 00e04c36ad17
ethernet_type 2048
IPSize 238
ip.tot_len 238
srcIP 10.34.10.101
dstIP 8.8.8.8
IPProtocol 17
IPTOS 0
IPTTL 64
IPID 0
UDPSrcPort 56699
UDPDstPort 443
UDPBytes 218
endSample   ----------------------
endDatagram   ================================
```

Sample Data Examples (sflowtool -l)

```
CNTR,10.34.10.15,17,6,100000000,1,1,217166129,2037293,20705,7155,0,15,0,6422251767,3952294,1440053,2032181,850219,0,2
FLOW,10.34.10.15,16,0,00e04c36ad17,000c2957be2c,0x0800,0,0,10.34.10.101,52.123.134.206,6,0x00,64,61561,443,0x10,70,52,8192
```

Breakdown of important fields

Field	Value	Meaning
`srcIP`	`10.34.10.101`	Local device — your endpoint
`dstIP`	`8.8.8.8`	Google DNS
`srcMAC`	`00:e0:4c:36:ad:17`	Device MAC (very important for correlation)
`dstMAC`	`00:0c:29:57:be:2c`	Likely the router/gateway
`ethernet_type`	`2048`	IPv4
`IPProtocol`	`17`	UDP
`UDPSrcPort`	`56699`	Ephemeral source port
`UDPDstPort`	`443`	TLS/HTTPS
`UDPBytes`	`218`	Payload length (excluding headers)

`inputPort`	`17`	Switch interface index (maps to GigabitEthernet 0/17)
`outputPort`	`0`	Could be unknown or broadcast in this case

Summary:

A device with IP `10.34.10.101` and MAC `00:e0:4c:36:ad:17` sent 218 bytes via UDP port 443 to Google DNS over interface 17.

Analysis & Value in S3M

Counter Samples

- Interface health monitoring (errors, discards)
- Link activity trends
- Port misbehavior or idle port detection

Flow Samples

- Per-device communication mapping (MAC/IP)
- Passive OS and role inference
- Service fingerprinting (e.g., DNS, SMB, TLS)
- Detection of unauthorized communication patterns

Correlations in S3M

- Match srcMAC with RADIUS-authenticated sessions
- Cross-reference dstIP/dstPort against threat intelligence
- Use flow metadata for policy violation detection (e.g., printer doing DNS?)
- Tag devices with behavior-based labels ("beaconing", "heavy TLS", etc.)

Suggested DB Schema

```
CREATE TABLE netflow_flows (
    id SERIAL PRIMARY KEY,
    timestamp TIMESTAMPTZ DEFAULT now(),
    src_ip TEXT,
    dst_ip TEXT,
    src_mac TEXT,
    dst_mac TEXT,
    protocol TEXT,
    src_port INTEGER,
    dst_port INTEGER,
    byte_count INTEGER,
    ingress_interface INTEGER,
    egress_interface INTEGER,
```

```
    device_ip TEXT,
    raw_sample JSONB
);
```

Next Steps

- Build lightweight parser for `sflowtool -l` output
- Insert parsed flow data into `netflow_flows`
- Enable frontend display for behavioral flows
- Correlate flow identity with RADIUS, SNMP, OUI
- Add Ansible automation for switch configurations
- Correlate destination ip addresses with existing dns records.

Notes

- sFlow provides statistical samples, not full traffic logs—ideal for scalable NAC.
- Requires no mirror ports or inline deployment.
- Works across vendor devices that support sFlow or NetFlow/IPFIX (future extension).

References

- sflowtool GitHub
- sFlow.org Specification
- Ruijie RGOS CLI Manual (Model-specific)

S3M Passive Flow Telemetry via NetFlow

S3M Passive Flow Telemetry via IPFIX