Enhancing Network Situational Awareness Using DPI Enhanced IPFIX

Hari Kosaraju

Prepared for Flocon 2013

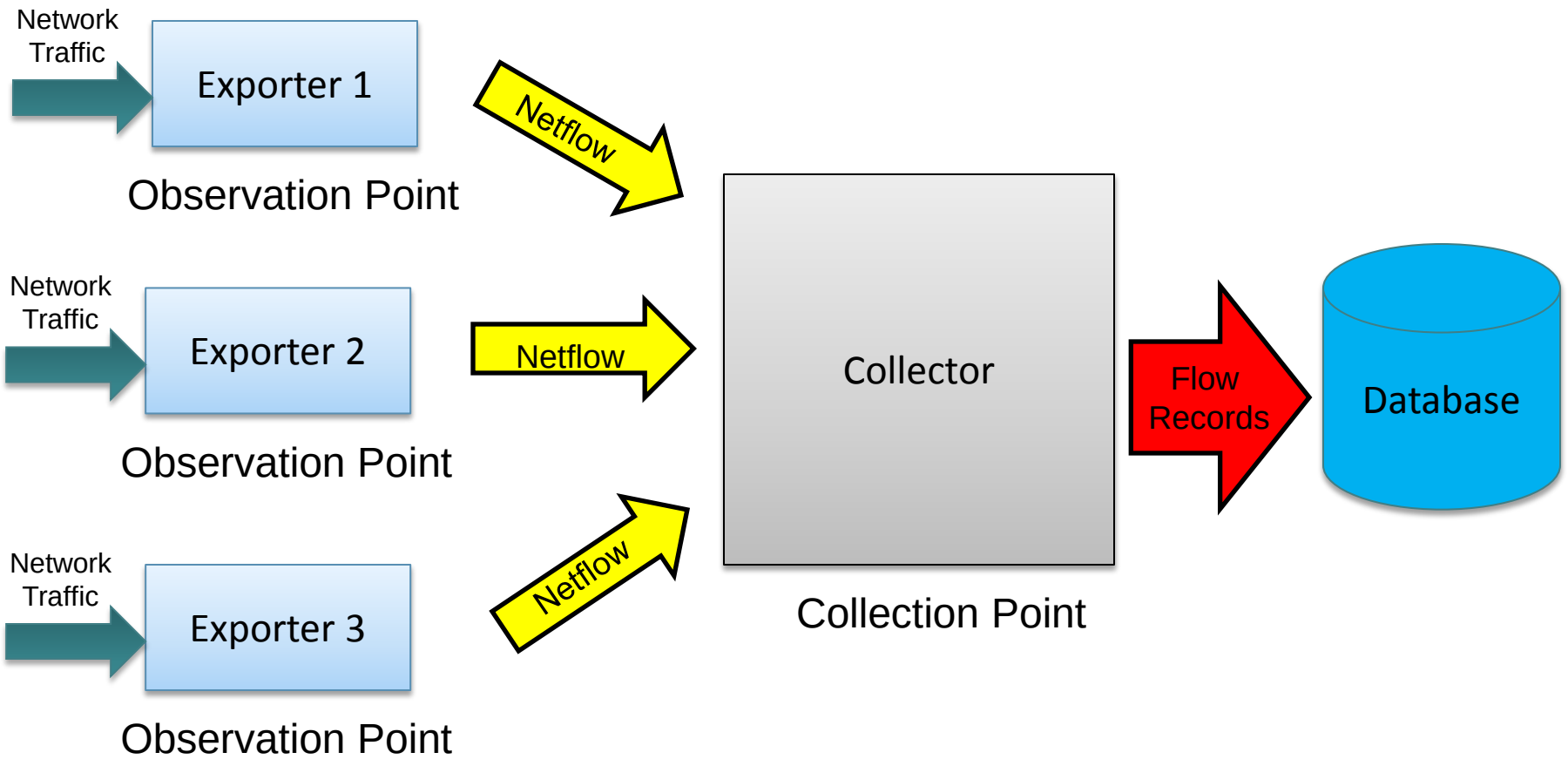
Agenda

- What is the difference between Netflow v5/v9 and IPFIX?
- How can we improve flow based traffic visibility?
- IPFIX Format for SessionVista
- How does this enhance Network Situational Awareness?
- Our implementation

Netflow Introduction

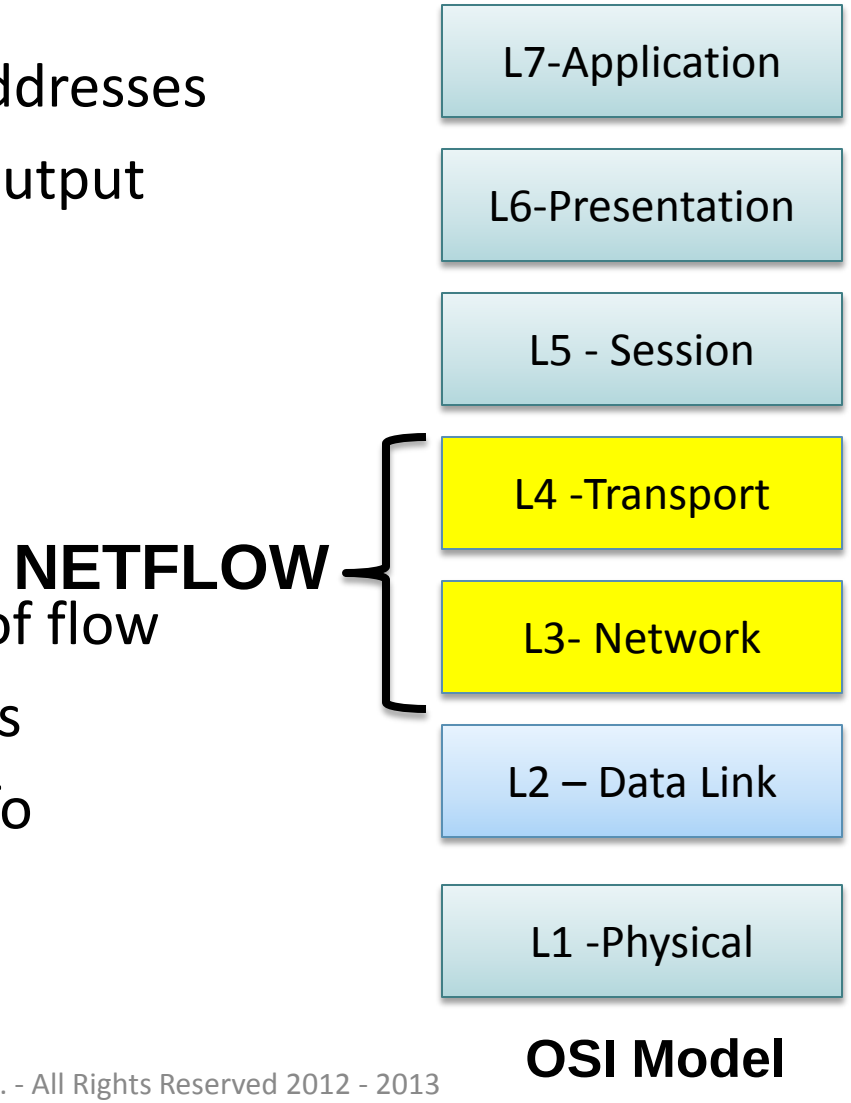
- Netflow is a protocol that was introduced by Cisco and is used for flow reporting on network traffic
- Information is typically reported on a flow basis, rather than on a packet basis
- However it is possible to report on packets via sampling
- The two popular versions are Netflow v5 and Netflow v9
- Other equipment vendors have their own variants but they are similar

Current Monitoring Paradigm



Information Reported in Netflow v5

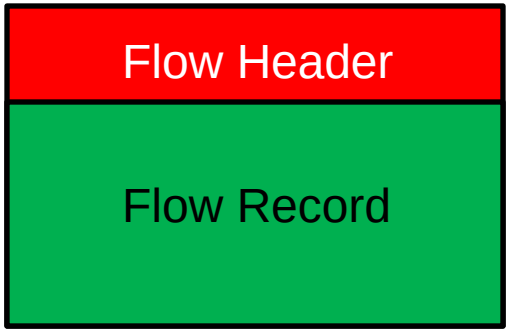
- Source and Destination IP addresses
- SNMP indices of input and output interface
- IP address of next hop
- Packets in the flow
- Total L3 bytes in flow
- Sysuptime of start and end of flow
- Source and Destination ports
- IP protocol, TOS, TCP flag info



The difference between Netflow v5 and v9

- Netflow v9 added support for IPv6 addresses
- **Concept of a template was introduced in Netflow v9**
- A template is a packet that is used to describe the structure of subsequent Netflow packets of the same identifier
- It is like a recipe that tells the Collector the format of the information to follow
- The advantage of this scheme is that the data sets are purely an identifier and associated data. They do not have any other parsing information which makes transport more efficient

Netflow v5

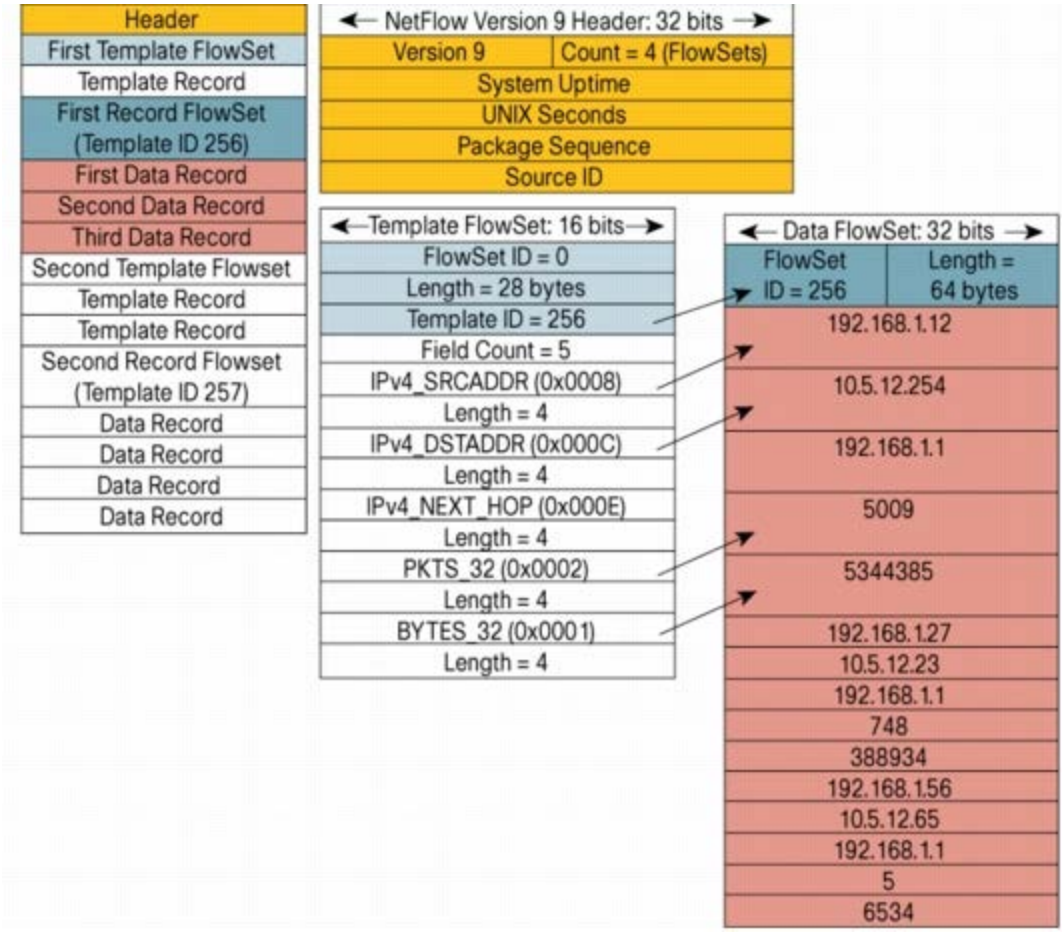


Fixed Format

Netflow v9

Extensible Format

Netflow v9 Structure



Source: Cisco.com

Pros and Cons of Netflow

Pro	Con
Gives flow level traffic visibility which enables numerous applications	Adds processing load to routers and switches
Reports on L3 and L4 information as well as flow timing	Is often run in sampled mode to reduce strain on the router and misses fidelity on small flows
Reports on flow length	Higher layer visibility limited to IP protocol field
Supported on many different networking devices natively	**Most analysis is based on ports**
	Does not handle tunneled traffic
	Only reports L3 and L4 metadata

How Do We Address These Issues?

IPFIX Introduced in 2008

- IPFIX was standardized by the IETF in Jan 2008
- It uses the template based approach started in

Netflow v9

- Completely self-contained in that it adjusts the data format as new elements are added
- Added Two Very Important New Features:
    1. **An Enterprise specific field**
    2. **Variable length fields**

It is space efficient and gives us flexibility to include Enterprise specific data!

Problem:
We need deeper traffic visibility

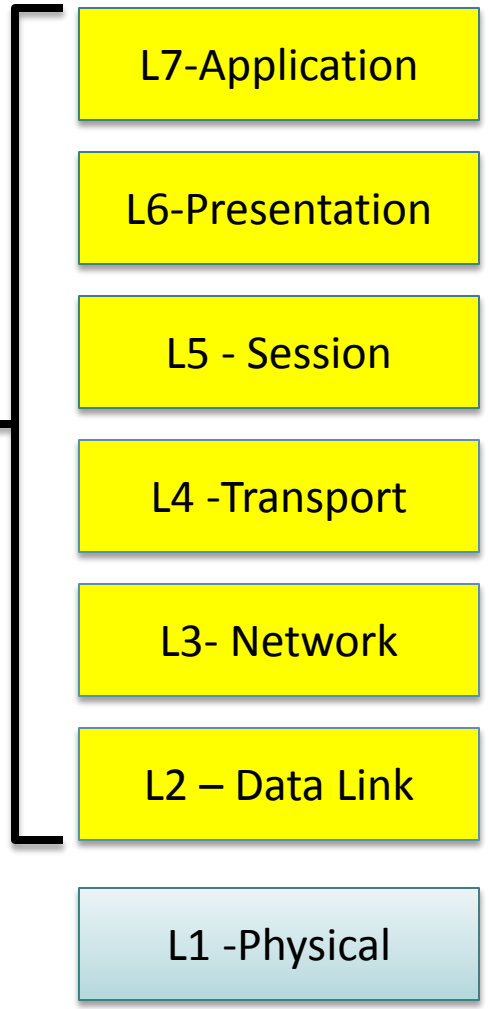
- We'd Like to See More Than L3 and L4 Metadata for better Network Situational Awareness

- Combine Deep Packet Inspection with IPFIX!

Deep Packet Inspection for L2 through L7 Visibility

- IPFIX has enterprise specific fields
- SessionVista has created one to encapsulate metadata extracted through Deep Packet Inspection
- What we do is report session level metadata using an IPFIX enterprise specific field
- The DPI engine can extract application layer metadata from different protocols (700 protocols and over 4000 metadata attributes)

SessionVista IPFIX



SessionVista IPFIX Format Design Challenges

1. Must work with multiple protocols and multiple attribute types
2. Must be able to handle transactional situations within a flow:
    1. Multiple emails within an SMTP download
    2. Multiple attachments on each email
3. Needs to encode data efficiently and perform an information reduction exercise (100: 1)
4. Report on flows with little to lots of metadata
5. Handle attributes that appear multiple times in one flow
6. Handle tunneled protocols and deep protocol stacks
7. Must be easy to add new protocols and attributes without changing the protocol

SessionVista IPFIX Format

- 64 Bit Flow Identifier to identify a bi-directional flow
- Generate an IPFIX report on the end of a session (bi-directional flow)
- Multiple IPFIX packets can be used to report on one Session
- Transactions within a flow are handled using a sub identifiers and transaction identifiers

- Every protocol we report on has a template
- Each template has a number of fixed elements and a variable data field
- All integer encoded data is placed in the fixed portion of the IPFIX packet, unless, it is data that can happen multiple times within a session.

SessionVista
Enterprise Specific Field

- Encodes Protocol Attributes using identifier, length and value semantics.
- This way, any number of variable length data items for a protocol can be stored.

[email-protocol-data]
// fixed fields (packet count and duration)
[2(attachments)**] [43** (seconds)**]**
// variable data field
[<length=65>
<emailSender><len=14>"alice@home.com"
<emailReceiver><len=12>"bob@work.com"
<emailReceiver><len=14>"carol@work.com"
<emailSubject><len=17>" Fwd:Status Report"]

Encoding Example

IPFIX Header
[flowID-data] [12abcd90-12efabcd (flowID)] [0000000013414111 (totalSysPackets)] [0000000000000013(totalFlowPackets)] [0000000049c1901e000000000003c4cb(flowStartTime)] [0000000049c1901f000000000006bd2b (flowEndTime)] [0001 (flowStatus)] [11626173652e69702e7463702e6874747000 (flowPath)]
[ip-protocol-data] [192.168.1.2 (source)] [172.16.17.23 (dest)] [200 (ttl)] [6 (tcp protocol)] [0x0123 (flags)]
[tcp-protocol-data] [3123 (source port)] [80 (dest port)]
[http-protocol-data] *// variable data field* [<length=80> <httpUrl><len=26>"books/list/bestsellers.htm" <httpServer><len=14>"www.amazon.com"]

Note: flowPath translates to base.ip.tcp.http

So what are the advantages?

- 15 layer deep protocol decode of 700 protocols along with per flow statistics
  - i.e. `"base-eth-ip-udp-gtp-tcp-http"`

- 4000 Metadata attributes from 700 protocols that are completely configurable

- Much richer dataset than existing tools without the storage costs associated with packet capture

- Easy to add new protocols and attributes without changing the IPFIX implementation

How does this enhance Network Situational Awareness?

- Survey the network and see things like:
  - Tunneling for obfuscation
  - DDoS (Application layer) detection
  - Network Asset Inventory : What is on your network and what services are they running (port agnostic)
  - Statistics provide the ability to perform application performance monitoring
  - Anomaly Detection (traffic trends, policy violation, data exfiltration)
  - Beaconing detection
- Alert on traffic events as they happen rather than doing a retrospective analysis of packet captures

SessionVista

How have we implemented this?

- IPFIX Exporter implemented in a multi-threaded Linux implementation
  - Scalable to over 10 Gbps in a 1U platform
  - Support for both Napatech and PCAP interfaces
  - Configurable to many multi-core architectures and memory requirements
- IPFIX Collector implemented in C++ on Linux
  - Accepts multiple connections
  - Scalable multi-threaded implementation
  - Backend to log file, MySQL, Hypertable and CEP engines

SessionVista

Thank You!

Hari Kosaraju

hkosaraju@mantaro.com

www.sessionvista.com