Towards Extended NetFlow/IPFIX

Vojtěch Krmíček

CESNET, z.s.p.o.
krmicek@cesnet.cz

Part I

Introduction
Changes over time

- simple counters, SNMP
- packet trace analysis
- IP flow monitoring
- combinations of above mentioned

Current challenges

- high-speed networks
- large amount of transferred data
- increasing requirements on monitoring
- lawful interception etc.
Advantages

- full information about traffic
- complete context
- tools for deep packet inspection (DPI)

Disadvantages

- extreme requirements for storage
  - e.g., 1 day of traffic from 10Gbps line – 30TB of data
- difficult to search
- legal issues (storing full dump)
Flow Based Monitoring

Advantages

- who communicates with whom, how long, how much data, ...
- faster searching inside data
- less amount of data (1 day 10Gbps line – approx. 20GB data)
- enables you to watch your network traffic in real-time.

Disadvantages

- less information (missing information from L5-L7)
- no standards for storing –> various tools for processing
NetFlow Applications in Time

Originally

Accounting
NetFlow Applications in Time

Originally

Accounting

Then

Incident handling
Network forensics
NetFlow Applications in Time

- **Originally**
  - Accounting

- **Then**
  - Incident handling
  - Network forensics

- **Now**
  - Intrusion detection
  - Network protection
Part II

Beyond NetFlow – What We Need?
What We Want to See?

Malicious activities

- accessing blacklisted URLs
- tunneling traffic
- bad DNS requests
- applications running on prohibited port
- botnet traffic

Data context

- IP address x type of communication
- IP address x person
- web server x domains
focus on data value instead of data volume
we need both L4 flow data and L7 knowledge
identify most important L7 fields and add them to the flow
Let’s identify needed L7 knowledge!
  - start with well-known protocols
  - focus on security
  - extract from header only? Or also from payload?
  - tools for protocol identification and DPI analysis needed
Can IPFIX Help Us?

NetFlow

- difficult customization
- bad support for extensions at collector sides
- CISCO proprietary format

IPFIX – IP Flow Information Export Format

- IETF standard providing new standardized possibilities
- templates mechanism, user-defined data types, freely extensible and adaptable to different scenarios
- possibilities to use IPFIX for variety of purposes
- missing proper tools for exporting/collecting IPFIX format
Part III

IPFIX – What To Add?
HTTP/HTTPS

Description

- no information about host name
- possible more webs on one IP address
- TCP ports 80, 443
- other useful information: browser, server, ...

Example HTTP(S) header fields:

Host: en.wikipedia.org
Referer: http://en.wikipedia.org/wiki/Main_Page
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Win64; x64; Trident/5.0)
Server: Apache/2.4.1 (Unix)
X-Client-Cert:

Use Cases

- phishing detection
- access to prohibited web sites
- wrong SSL certificate detection
DNS

Description

- no information about requested domains and answers
- UDP port 53
- other useful information: TTL, number of IP addresses

Example DNS header fields:

QNAME: www.paris.fr
QRTYPE: 1 (Host Address)
RDATA: 195.113.232.98
TTL: 7200

Use Cases

- botnet detection (similar DNS requests)
- fast-flux detection
Description

- only addresses of mail servers in flow.
- TCP ports 25 (SMTP), 110 (POP3), 143 (IMAP)
- useful information: from, to, attachments, etc.

Examples of email header fields:

From: MetadateCD@beamaffection.pl.ua
Date: Thu, 01 Mar 2012 09:56:11 +0100
To: krmicek@cesnet.cz
Subject: I’m young again and full of energy

Use Cases

- SPAM detection
Description

- no information about IRC nicks, channels, etc.
- TCP port 6667 and nearby port numbers
- useful information: Nick, Channel, Connection time

Example IRC commands:

JOIN: [Lsass]
NICK: RBOT|DEU|XP-1234

Use Cases

- IRC Botnet detection
Tunneled IPv4/IPv6 Traffic

Description
- no information about tunneled traffic in flows
- decapsulation methods necessary
- useful information: tunneled protocol, IPv4, IPv6

Use cases
- detection of tunneled applications over port 80
- applications running on wrong ports
- detection of IPv6 in IPv4
- security threats in encapsulated traffic
Other Useful Fields/Protocols/Information

- OS profiling - TCP SYN packet size, TTL, IP ID
- VoIP - control connections, outages
- NAT detection
- lower layers information
  - control networks (BACNET)
  - Ethernet - MAC addresses
  - time characteristics - innerpackets delays, jitter, etc.
- flag "flow exported on active timeout"
- pair flows
Part IV

Current State of Tools
What We Need to Have Extended IPFIX?

Chain of tools

- flow exporter with DPI/L7 apps. identification
- flow protocol supporting additional fields
- flow collector with protocol support
- new tools utilizing added information

What We Can See Around Us

- exporters/routers supporting NetFlow v5/v9
- robust NetFlow collectors
- set of tools for processing standardized NetFlow
- some attempts and playing with IPFIX...
- AppFlow\(^1\) – standard. IPFIX extension with HTTP, DNS fields

\(^1\)www.appflow.org
Existing Open Source Tools Examples

**IPFIX exporters**
- YAF with Application Labeling and DPI
- nProbe with L7 OpenDPI

**IPFIX collectors**
- SiLK
- nTop
- NfSen - experimental IPFIX implementation only
- Vermont

Problems with full IPFIX support, templates/lists in IPFIX, speed limits, compatibility between exporters and collectors.
What We Have at CESNET

**Exporter: FlowMon Probe**
- exporting both NetFlow and IPFIX format
- plugin API for further data processing (DPI, L7, ...)

**Collector: IPFIXcol**
- IPFIX collector fully supporting RFC 5101
- database backend for fast searching in data (FastBit)
- plugins for processing other formats (pcap, NetFlow, ...)
- in development\(^2\), now in pilot deployment

\(^2\)More details about ipfixcol at AIMS 2012: *Flow Information Storage Assessment Using IPFIXcol*
Part V

Conclusion
Conclusion

Need for extended IPFIX

- NetFlow is missing L7 information
- advanced analysis is not possible without such knowledge
- storing full packet traces is not possible
- we need to use IPFIX and add L7 info

Problems with tools

- various implementations, mutual incompatibility
- IPFIX is "too general" - difficult to fully implement
- we need whole chain of tools (exporter, collector, processing)
- missing tools for processing extended NetFlow/IPFIX
  (compare to many tools for processing libpcap format)
your experiences with IPFIX deployment

• tools/exporters/collectors...

• which L7 fields and protocols are interesting?
Thank you for your attention.