NetFlow Data Visualization Based on Graphs

Pavel Minarik[1], Tomas Dymacek[2]

[1] Institute of Computer Science, Masaryk University
Botanická 68a, 602 00 Brno, Czech Republic
pavel.minarik@mail.muni.cz
[2] Mycroft Mind, Inc.
Lidická 28, 602 00 Brno, Czech Republic
dym@mycroftmind.com

Abstract. We present an innovative approach to NetFlow data processing and visualization developed at Masaryk University in Brno. Our visualization method based on graphs bridges the gap between highly aggregated information visualization represented by charts and too much detailed information represented by the log files. In our visualization method the graph nodes stand for network devices and oriented edges represent communication between these devices. We also present the utilization of external data sources (DNS, port names, etc.), which helps to present NetFlow data in more intuitive way. Hence this approach is very natural one for both network administrators and non-specialists. Based on these methods a proof-of-concept tool called *NetFlow Visualizer* has been developed and is now offered as an plug-in for the NetFlow probes.

Key words:visualization, visual analytics, NetFlow data, graphs

1 Introduction

The usage of NetFlow data [1] in the network monitoring domain is growing. Available tools for NetFlow data processing such as *NFSen* [2] are equipped with large scale visualization represented by charts (see figure 1) on the one hand and very detailed visualization represented by lists of NetFlow data log files on the other (see figure 2).

We believe that these methods are not sufficient for the analysis of network traffic and that there is a gap between these two methods. Charts may give the analyst the whole picture of the situation in the network. Listing log files gives the analyst all the details. However, with thousands of logged connections, it is extremely complicated to process this data, particularly when the analyst does not know exactly what he/she is looking for.

We are therefore introducing a visualization method of NetFlow data based on graphs (see figure 3) which can supply chart-based visualizations. Our method focuses on network devices (graph nodes) and communication between theses devices (oriented edges) aggregated on different levels. This scalable level of detail (level of aggregation) is suitable for the analysis of network traffic where

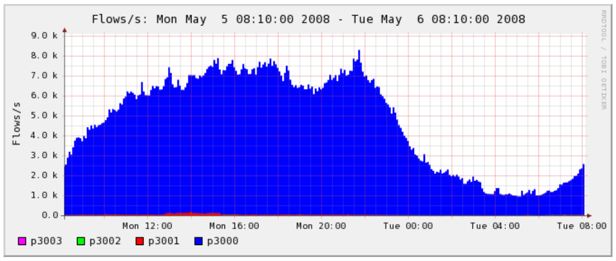
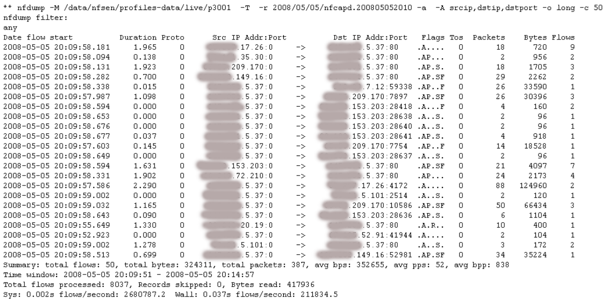


Fig. 1. Chart (based on NetFlow data) visualization example.



```
** nfdump -M /data/nfsen/profiles-data/live/p3001  -T  -r 2008/05/05/nfcapd.200805052010 -a  -A srcip,dstip,dstport -o long -c 50
nfdump filter:
any
Date flow start          Duration Proto      Src IP Addr:Port          Dst IP Addr:Port   Flags Tos  Packets    Bytes Flows
2008-05-05 20:09:58.181     1.965     0     [redacted].17.26:0     ->     [redacted].5.37:80    .A....   0       18      720     9
2008-05-05 20:09:58.094     0.138     0     [redacted].35.30:0     ->     [redacted].5.37:80    .AP...   0        2      956     2
2008-05-05 20:09:58.131     1.923     0    [redacted]209.170:0     ->     [redacted].5.37:80    .AP.S.   0       18     1705     3
2008-05-05 20:09:58.282     0.700     0    [redacted].149.16:0     ->     [redacted].5.37:80    .AP.SF   0       29     2262     2
2008-05-05 20:09:58.338     0.015     0      [redacted].5.37:0     ->    [redacted].7.12:59338 .AP..F   0       26    33590     1
2008-05-05 20:09:57.987     1.098     0      [redacted].5.37:0     ->  [redacted].209.170:7897  .AP.SF   0       26    30396     3
2008-05-05 20:09:58.594     0.000     0      [redacted]5.37:0     ->  [redacted].153.203:28418 .A...F   0        4      160     2
2008-05-05 20:09:58.653     0.000     0      [redacted].5.37:0     ->  [redacted].153.203:28638 .A..S.   0        2       96     1
2008-05-05 20:09:58.676     0.000     0      [redacted].5.37:0     ->  [redacted].153.203:28640 .A..S.   0        2       96     1
2008-05-05 20:09:58.677     0.037     0      [redacted].5.37:0     ->  [redacted]153.203:28641 .AP.S.   0        4      918     1
2008-05-05 20:09:57.603     0.145     0      [redacted].5.37:0     ->  [redacted].209.170:7754  .AP..F   0       14    18528     1
2008-05-05 20:09:58.649     0.000     0      [redacted].5.37:0     ->  [redacted].153.203:28637 .A..S.   0        2       96     1
2008-05-05 20:09:58.594     1.631     0    [redacted].153.203:0     ->     [redacted].5.37:80    .AP.SF   0       21     4097     7
2008-05-05 20:09:58.331     1.902     0    [redacted].72.210:0     ->     [redacted].5.37:80    .AP...   0       24     2173     4
2008-05-05 20:09:57.586     2.290     0      [redacted].5.37:0     ->   [redacted].17.26:4172  .A....   0       88   124960     2
2008-05-05 20:09:59.002     0.000     0      [redacted].5.37:0     ->   [redacted].5.101:2514  .A..S.   0        2      120     1
2008-05-05 20:09:59.032     1.165     0      [redacted].5.37:0     ->  [redacted].209.170:10586 .AP.SF   0       50    66434     3
2008-05-05 20:09:58.643     0.090     0      [redacted].5.37:0     ->  [redacted].153.203:28636 .AP.S.   0        6     1104     1
2008-05-05 20:09:55.649     1.330     0     [redacted]20.19:0     ->     [redacted].5.37:80    .A.R..   0       10      400     1
2008-05-05 20:09:52.923     0.000     0      [redacted].5.37:0     ->   [redacted].52.91:41944 .A....   0        2      104     1
2008-05-05 20:09:59.002     1.278     0     [redacted].5.101:0     ->     [redacted].5.37:80    .A..S.   0        3      172     2
2008-05-05 20:09:58.513     0.699     0      [redacted].5.37:0     ->   [redacted].149.16:52981 .AP.SF   0       34    35224     1
Summary: total flows: 50, total bytes: 324311, total packets: 387, avg bps: 352655, avg pps: 52, avg bpp: 838
Time window: 2008-05-05 20:09:51 - 2008-05-05 20:14:57
Total flows processed: 8037, Records skipped: 0, Bytes read: 417936
Sys: 0.002s flows/second: 2680787.2  Wall: 0.037s flows/second: 211834.5
```

Fig. 2. NetFlow log file listing example.

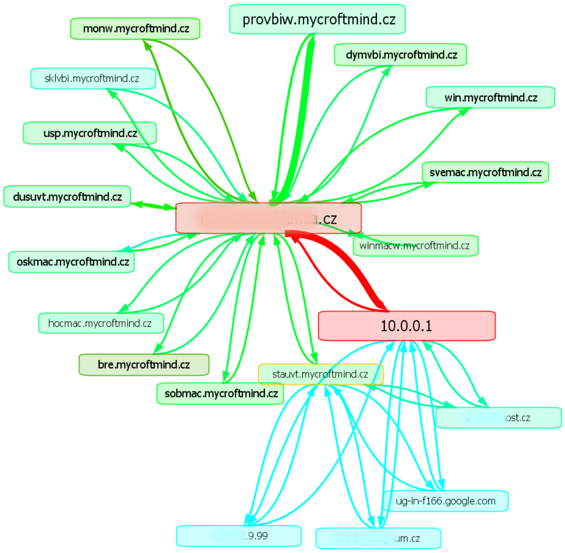


Fig. 3. Visualization of NetFlow data based on graphs example.

the analyst is able to see the whole picture of the situation on the network and is able to focus on every single data transfer (flow) at once.

It is also very important to do as much mechanical work for the analyst as possible. The other concept addresses the utilization of external data sources. The key idea is to provide additional information (domain names, port information, etc.) to help the analyst during the process of data analysis. We should note there that this is the piece of information which the analyst seeks manually when working with the NetFlow log files.

2 Related work

There are several approaches which use graph-based visualization in the network monitoring domain.

Interactive Network Active-traffic Visualization (INAV) is a monitoring solution for use in real-time network environments. It monitors the traffic currently active between nodes [3].

Netview is a graph-based network visualization tool that displays an animated realtime view of the network. Traffic noticed by the *netview-server* is classified and aggregated and in turn animated by the *netview-client* [4].

jpcap – a network packet capture library provides real-time decomposition and graph-based visualization of network traffic [5].

All of these solutions use a similar visualization method but they have a different purpose to our approach. They are used for real-time network traffic monitoring and do not support different time slots comparison and analysis. Therefore only a limited number of traffic attributes is processed. Filtering possibilities are also not sufficient. They do not provide WHOIS information or additional port information.

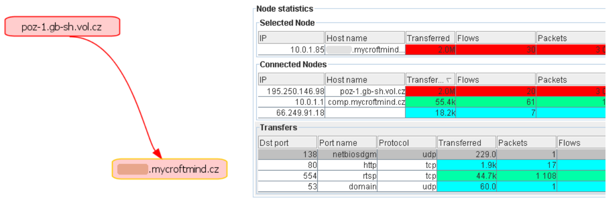
3 Visualization Method Properties

Motivation for our work is the research and development agenda for visual analytics to facilitate advanced analytical insight. This agenda was presented by the *National Visualization and Analytics Center* in the book called *Illuminating the Path* [6].

Our visualization method reflects the recommendations presented in [6]. The main properties of our visualization method may be characterized by the following points (only fundamental functionality related to visualization is presented):

- **Graph-based visualization** (so called dynamic mind map visualization) of the communicating network devices. Visualization method suitable for presenting connections. Edges are oriented according to flow direction.
- **Spreadsheet based visualization** of communication details and statistics.

- **Multiple level of detail** offers communication aggregation between network devices aggregated by protocol, details of the communication aggregated using protocol and destination port or pure NetFlow data visualization.
- **Dynamic visualization adjustment** according to actual data. Coloring and sizing of nodes representing network devices and edges representing communication between these devices. Coloring and sizing changes dynamically according to selected attributes of the NetFlow records and their current values (peaks and lows) present in current data. E.g., size of a node corresponds to number of packets transmitted by the node, its color corresponds to to amount of transferred data, size of an edge corresponds to number of flows transferred and its color corresponds to the number of packets transferred.
- **User defined visualization adjustment** for selected nodes representing network devices. These user defined devices may be visualized using different node shape or size. Visualization settings are tied to IP addresses.



Node statistics

Selected Node

IP	Host name	Transferred	Flows	Packets
10.0.1.85	[redacted].mycroftmind...	2.0M	30	3 0

Connected Nodes

IP	Host name	Transfer... ▽	Flows	Packets
195.250.146.98	poz-1.gb-sh.vol.cz	2.0M	20	3 0
10.0.1.1	comp.mycroftmind.cz	55.4k	61	1
66.249.91.18		18.2k	7	

Transfers

Dst port	Port name	Protocol	Transferred	Packets	Flows
138	netbiosdgm	udp	229.0	1	
80	http	tcp	1.9k	17	
554	rtsp	tcp	44.7k	1 108	
53	domain	udp	60.0	1	

Fig. 4. Example of graph-based visualization complemented with spreadsheet visualization of statistics for selected node.

Presented visualization method is built on *Prefuse* [7] visualization toolkit complemented by standard JAVA components for spreadsheet visualization (see figure 4).

4 NetFlow Visualizer

The visualization method presented was implemented by *Mycroft Mind Inc.* [8]. The resulting product is called *NetFlow Visualizer* [9] and is provided with *INVEA-TECH Inc.* [10] *FlowMon probes* [11] as a freeware plug-in. Innovated version of *NetFlow Visualizer* is being developed concurrently.

NetFlow Visualizer is the client-part of client-server solution. The server is called *NFSel* and its purpose is to provide NetFlow data. *NFSel* is a part of the *FlowMon probe* and is invisible for users. *NFSel* is a *XML-RPC* server which provides NetFlow data from the NetFlow data collector using a standard tool called *NFDump* [12]. *NFSel* converts this data into a *GraphML* format [13]. Figure 5 is a comprehensive illustration of the architecture.

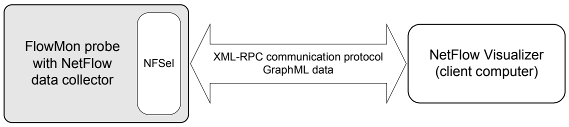
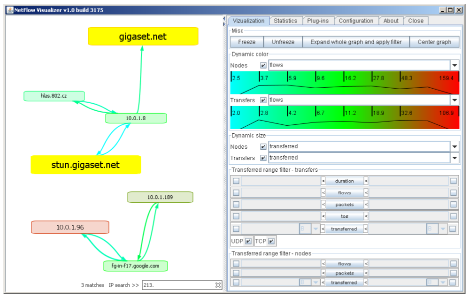


Fig. 5. System architecture overview diagram.

NetFlow Visualizer itself is a client-side Java application which visualizes NetFlow data provided by *NFSel* upon request in *GraphML* format. *NetFlow Visualizer* uses the graph-based visualization method and provides user interface (see figure 6) with additional filtering or search features and external data sources utilization. *NetFlow Visualizer* provides basic filtering possibilities utilizing parameters of pure NetFlow data (protocols, amount of transferred bytes, packets, etc.).


NetFlow Visualizer v1.0 build 3175

Dynamic color
Nodes ☑ flows
Transfers ☑ flows

Dynamic size
Nodes ☑ transferred
Transfers ☑ transferred

UDP ☑ TCP ☑

3 matches IP search >> 213.

Fig. 6. *NetFlow Visualizer* tool. Illustration of the visualization properties tab and IP search feature.

Another extension lies in external data sources utilization. The purpose of data sources utilization is to visualize communication on the network more naturally and clearly for the operators. *NetFlow Visualizer* tries to do as much mechanical work for the analyst as possible and so the analyst can focus on his/her work instead of searching for port names or translating IPs into domain names. The following data sources are utilized by *NetFlow Visualizer*:

- **DNS (Domain Name Service)** – Human beings are used to working with names; computers, however, use numeric identifiers. Translating IP addresses into corresponding domain names is crucial especially in large networks. Information about domain name is not present in the NetFlow data. It should therefore be obtained from a proper DNS server online.
- **WHOIS Service** – It is sometimes necessary to search for additional information about a network device, e.g. its location or administrative contact. Direct integration of the WHOIS service saves analyst's time.

- **Port names** – Even experienced network administrators might not be familiar with uncommon port numbers. The motivation is similar to DNS. Name and description is much more than a number. This data source provides translations from pairs protocol, port into a port name[3] and its description.

5 Use-case

In this section we would like to present a simple use-case and compare *NetFlow Visualizer* with the classic approach using *NFDump* [12] and *NFSen* [2] tool. Our use-case will be the exploration of traffic between the top data producers/consumers in the monitored network.

Use-case procedure using *NFDump*

1. Construct query to obtain top N traffic producers or consumers. Mark the results. Example of corresponding *NFDump* query:
   ```
   nfdump -M /live/p3000 -T -r nfcapd.200805130405 -n 10 -s ip/bytes
   ```
2. Construct query to obtain the communication of the devices from the previous step (aggregation using source IP address, destination IP address). Example of corresponding *NFDump* query:
   ```
   nfdump -M /live/p3000 -T -r nfcapd.200805130405 -a -A srcip,dstip,proto
   IP XXX.YYY.ZZZ.UUU or IP...
   ```
3. Construct query to obtain the communication between selected IP addresses (pure data or aggregated by destination port). Example of corresponding *NFDump* query to get pure NetFlow data:
   ```
   nfdump -M /live/p3000 -T -r nfcapd.200805130405 IP XXX.YYY.ZZZ.UUU or IP...
   ```

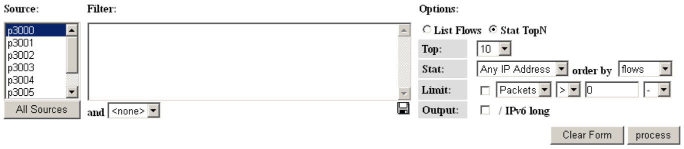
Use-case procedure using *NFSen*

1. Obtain top N traffic producers or consumers using user interface for top N statistics (see figure 7).
2. Set the filter in user interface for flows listing. Copy IP addresses acquired in the previous step into ”Filter” text box. Set aggregation using source IP address, destination IP address and protocol in user interface.
3. Analogous to previous step. Add aggregation using destination port or remove aggregation completely to get pure NetFlow data.

Use-case procedure using *NetFlow Visualizer*

1. Set time interval and press the ”Load data” button, *NFSel* will acquire data from the collector and deliver it to *NetFlow Visualizer*.
2. Set filter ”nodes transferred more than” using user interface (see figure 6).
3. Expand with a single mouse click one or all the nodes to obtain the communication of the devices satisfying the filter.

[3] A ”port name” is just a shortcut for ”typical service running on the given port and available via the given transport protocol”.



Source:

p3000
p3001
p3002
p3003
p3004
p3005

All Sources

Filter:

and <none>

Options:

○ List Flows ● Stat TopN

Top: 10

Stat: Any IP Address order by flows

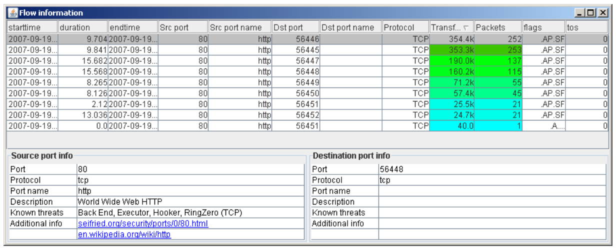
Limit: □ Packets > 0 -

Output: □ / IPv6 long

Clear Form process

Fig. 7. *NFSen* user interface for top N statistics.

4. Open with a single mouse click the statistics tab for selected device to obtain the data aggregated by destination port (see figure 4) or open the edge between two devices to obtain the pure NetFlow data in a table (see figure 8) using its context menu.



Flow information

starttime	duration	endtime	Src port	Src port name	Dst port	Dst port name	Protocol	Transf...	Packets	flags	tos
2007-09-19...	9.704	2007-09-19...	80	http	56446		TCP	354.4k	252	.AP.SF	0
2007-09-19...	9.841	2007-09-19...	80	http	56445		TCP	353.3k	253	.AP.SF	0
2007-09-19...	15.682	2007-09-19...	80	http	56447		TCP	190.0k	137	.AP.SF	0
2007-09-19...	15.568	2007-09-19...	80	http	56448		TCP	160.2k	115	.AP.SF	0
2007-09-19...	8.265	2007-09-19...	80	http	56449		TCP	71.2k	55	.AP.SF	0
2007-09-19...	8.126	2007-09-19...	80	http	56450		TCP	57.4k	45	.AP.SF	0
2007-09-19...	2.12	2007-09-19...	80	http	56451		TCP	25.5k	21	.AP.SF	0
2007-09-19...	13.036	2007-09-19...	80	http	56452		TCP	24.7k	21	.AP.SF	0
2007-09-19...	0.0	2007-09-19...	80	http	56451		TCP	40.0	1	.A....	0

Source port info

Port	80
Protocol	tcp
Port name	http
Description	World Wide Web HTTP
Known threats	Back End, Executor, Hooker, RingZero (TCP)
Additional info	seifried.org/security/ports/0/80.html
	en.wikipedia.org/wiki/http

Destination port info

Port	56448
Protocol	tcp
Port name	
Description	
Known threats	
Additional info	

Fig. 8. Table of pure NetFlow data (replies of a web server to a client).

Let's summarize the use case presented. Using *NFDump* or *NFSen* users have to think out and write their own commands or set up a complicated filter. It is clear that using a tool such as *NetFlow Visualizer* can increase dramatically the productivity of labour of network analysts and make NetFlow data analysis accessible even to non-specialists.

6 Conclusion and Future Work

The NetFlow data visualization method based on graphs was created, presented, evaluated and discussed with network and security experts. The main conclusion is that such a method has big potential for complementing existing methods and is very useful for specific use-cases. The fact that this method has been implemented by a commercial company and is being provided to customers as a visualization tool called *NetFlow Visualizer* with NetFlow data probes confirms its usefulness and potential.

The visualization method presented was adapted in the CAMNEP project [14] to provide visualization of anomalous traffic according to detection layer results. Visualization tool *NetFlow Visualizer* was verified at Masaryk University within the framework of the internal security targeted project.

The development of the visualization tool presented and the visualization method itself is not yet finished. One of the biggest challenges is to provide an analyst with the exact portion of information required. Network traffic data are massive and quick insight will not be possible if the analyst is subjected to information overload. It is necessary to provide the analyst with a powerful and intuitive way of defining and expressing his/her actual focus. Another challenge is to permit any number of centers of focus so that users will be able to view details for more than one node at one time. The solution proposed is to combine graph-based and spreadsheet-based visualization in one workspace. The graph nodes will therefore contain spreadsheets with details.

Acknowledgment: This material is based partially upon work supported by the European Research Office of the US Army under Contract No. N62558-07-C-0001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the European Research Office of the US Army.

References

1. Cisco Systems: Cisco IOS NetFlow. http://www.cisco.com/go/netflow (2007)
2. Haag, P.: NfSen - NetFlow Sensor. http://nfsen.sourceforge.net (2007)
3. Robinson, N. and Scaparra, J.: Interactive Network Active-traffic Visualization (INAV). (http://inav.scaparra.com/docs/whitePapers/INAV.pdf)
4. Cornell University, Department of Computer Science: Netview. (http://netview.gforge.cis.cornell.edu/index.php)
5. jcap project team: jpcap – a network packet capture library. (http://jpcap.sourceforge.net/)
6. Chinchor, N., Hanrahan, P., Robertson, G., Rose, R.: Illuminating the Path: The Research and Development Agenda for Visual Analytics. National Visualization and Analytics Center (2006)
7. Berkeley Institute of Design: The Prefuse Visualization Toolkit. (http://www.prefuse.org)
8. Mycroft Mind Inc.: Mycroft Mind Inc. Company Profile. (http://www.mycroftmind.com)
9. Mycroft Mind Inc.: NetFlow Visualizer. (http://www.mycroftmind.com/products:nfvis)
10. INVEA-TECH Inc.: INVEA-TECH Inc. Company Profile. (http://www.invea.cz/main/home)
11. Čeleda, P., Kováčik, M., Konířr, T., Krmíček, V., Špringl, P., Žádník, M.: FlowMon Probe. Technical Report 31/2006, CESNET, z. s. p. o. (2006) http://www.cesnet.cz/doc/techzpravy/2006/flowmon-probe.
12. Haag, P.: NFDUMP - NetFlow processing tools. http://nfdump.sourceforge.net (2007)
13. Graph Drawing Steering Committee: GraphML format. (http://graphml.graphdrawing.org)
14. Agent Technology Group, Gerstner Laboratory, Czech Technical University in Prague and Institute of Computer Science, Masaryk University in Brno: CAMNEP (Cooperative Adaptive Mechanism for NEtwork Protection) project web page. (http://http://agents.felk.cvut.cz/projects/camnep)