FlowTag: A Collaborative Attack-Analysis, Reporting, and Sharing Tool for Security Researchers

Christopher P. Lee
Georgia Institute of Technology
School of Electrical and Computer Engineering
Communications Systems Center Lab
chrislee@gatech.edu

John A. Copeland
Georgia Institute of Technology
School of Electrical and Computer Engineering
Communications Systems Center Lab
john.copeland@ece.gatech.edu

ABSTRACT

Current tools for forensic analysis require many hours to understand novel attacks, causing reports to be terse and untimely. We apply visual filtering and tagging of flows in a novel way to address the current limitations of post-attack *analysis*, *reporting*, and *sharing*. We discuss the benefits of visual filtering and tagging of network flows and introduce FlowTag as our prototype tool for Honeynet researchers. We argue that online collaborative analysis benefits security researchers by organizing attacks, collaborating on analysis, forming attack databases for trend analysis, and in promoting new security research areas. Lastly, we show three attacks on the Georgia Tech Honeynet and describe the analysis process using FlowTag.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: GeneralSecurity and Protection;; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and IndexingAbstracting methods;; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces;; I.3.8 [**Computer Graphics**]: Applications

General Terms

Security, Human Factors

Keywords

information visualization, user interfaces, tagging, folksonomy, team collaboration, network attack analysis, parallel coordinates, honeynet

1. INTRODUCTION

After a cyberattack has occurred, it is often necessary to determine the exploit used, the severity of the compromise, and the motives of the attacker. These key pieces of information are commonly used for system repair, strengthening of security policies, malware/exploit collection, and trend analysis. For Honeynet researchers, trend analysis and early detection of novel attacks are often the primary goals. Enterprise administrators focus on system repair and policy. Internet providers generally focus on the verification of a reported attack and identification of the culprit. Regardless of the desired goal, attack analysis can be a difficult and time-consuming task. In this paper, we focus on the needs of Honeynet researchers.

VizSEC'06, November 3, 2006, Alexandria, Virginia, USA.

1.1 Attack Analysis with Logs, IDS, and Ethereal

Analysts have several sources of information to aid them in understanding an attack: system logs, IDS logs, and network capture files (usually PCAP). System logs often do not have adequate information in them to understand an attack and can often be deleted by the attacker. Signature-based IDS logs tend to have excellent descriptions of attacks they recognize, but fail to show what an attacker does once inside. Anomaly-based IDS systems are good at detecting new attacks, but give general alerts that give very little information about the attack itself. Furthermore, IDS systems are notorious at being noisy and difficult to tune to the alarms of interest to the administrator. Network capture files contain all the network traffic. However, with so much detail, it is easy to become overwhelmed and lose context.

Common tools for network capture analysis include tcpdump, Ethereal, and tcpflow. Ethereal is perhaps the most commonly used and the easiest to use without scripting; we describe it here for comparison. Ethereal, shown in Figure 1, is a graphical interface that displays packets in a table, the dissection tree of the selected packet, and the contents of the packet in hex. Ethereal's dissectors support almost all well-known protocols. Filters in Ethereal are specified as query strings of the form `ip.addr == 10.0.43.1` and, when mastered, are extremely powerful at expressing complex queries. Functionality exists to click on a TCP packet and view the reconstructed payload of that TCP flow, but because Ethereal automatically filters out other packets to perform the viewing of the complete payload, context with other flows is lost. This is a key tool for debugging network protocols, but for the specific task of examining attacks, this tool is cumbersome because of its packet-based views, narrow queries, and processing time overhead.

Even for experienced analysts, analyzing a compromise is a long and arduous task that requires lots of copy and pasting, finding key packets in Ethereal, searching, filtering, and keeping context in mind at all times. A proper analysis often takes days for a short attack and weeks for more complex attacks. Because analyzing attacks is such a laborious task, reports are often delayed or never generated. The resulting reports often lack the overview and detail structure for rapid comprehension and have a variety of formats that are hard to index and search. These limitations slow the dissemination of attack captures to the security community and make trend

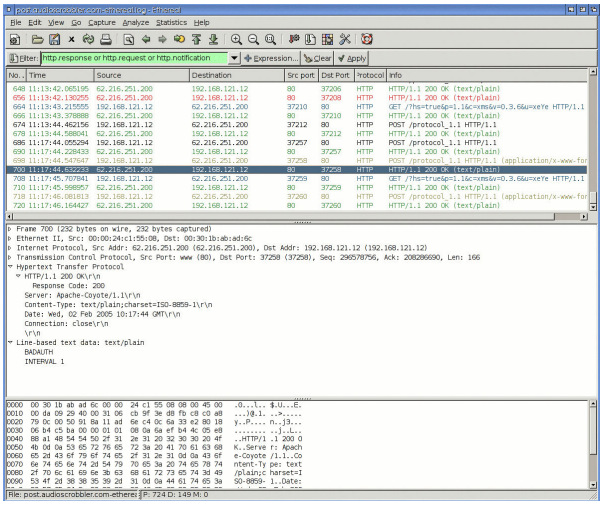


Figure 1: Ethereal screen shot

analysis very difficult due to the lack of internet visibility. This cripples the community's ability to learn new attacks and observe trends.

FlowTag allows the visualization and tagging of flows to enable four tasks: recording of details to keep context, rapid report generation, data sharing, and tag-based searching.

1.2 Social Navigation and Tagging

Tag-based finding and discovery is an internet phenomenon included in the buzzword, "Web 2.0". Among the many popular sites that have embraced tagging are Flickr, Delicious, YouTube, RawSugar, Technorati, and BlinkList. Tagging allows people to create a *folksonomy*, a term coined by Thomas Vander Wal, in which items are not categorized into a strict taxonomy, but loosely associated by keywords. Folksonomies require low cognitive costs of categorization[1, 9, 11], are enjoyable [10], and provide immediate benefits to the tagger [7, 10]. Tagging also allows for serendipitous discovery of related items and people [7]. These sites also allow communities to form where members can discuss items and each other.

FlowTag uses the tags assigned to flows to tag the attack package for use in report generation and sharing with the security community. The security community can then search for the package, apply new tags, and see related attacks easily. This would allow both advanced and beginning analysts to share in a collaborative environment. We apply folksonomies to security analysis due to the weaknesses of taxonomies and the need to share information among the research community. Quintarelli [8] lists the following limitations of hierarchical-enumerative taxonomies: inflexible and over-generalized categorization; terms and concepts for categorization are decided by a small number of people and are not usually universal; can rarely anticipate future items and concepts; miscategorization quickly degrades the efficacy of the system.

Since the security field is constantly changing, new attack taxonomies are constantly created to better categorize the range of attacks and exploits. This has obsoletes most of the useful taxonomies of the past and makes finding attacks difficult because of the changing terminology. In the past, the terms "viruses", "worms", and "trojans" were used to describe various types of malicious software. Most malicious software we see today combines all of these in one download, along with newer components like spyware, ransomware, botnets, key loggers, and rootkits. The software is no longer categorizable in the previous taxonomy, and is hard to find since there is no clear starting point other than the platform on which it runs–and even that assumption is sometimes an invalid way to divide malware samples.

Folksonomies work best on sprawling, heterogeneous sets, made up of an enormous, ever-changing, time-sensitive, ambiguous corpus of items, especially when there is a growing group of participants and no central authority according to Quintarelli [8]. Security research fits this description well.

It can be argued that folksonomies are inexact and degrades the ability to find items. However, there is a trade-off between exactness and expediency, and we argue that there are invariants in an attack which would naturally be used when tagging, keeping it highly findable although it is probable not to find all items of a given category. Folksonomies typically have a long tail distribution of terms where almost everyone agrees on a few terms to describe an item, but there are many terms that only one person would use on that item (typically experts and novices). Voss [12] measured the tag distributions of Wikipedia and Del.icio.us among all the pages and noted that 96% of items are tagged with nine or less tags. Cattuto [2] goes further and measures the correlation between tags and reports that users of collaborative tagging systems share universal common behaviors in tagging. The examples provided in this paper illustrate this phenomenon with security information.

2. FLOWTAG DESCRIPTION

FlowTag addresses three goals and embodies two contributions. The tool enables quick analysis, reporting, and sharing of attack data, and thus address the issues of attack analysis facing Honeynet researchers. Its primary contribution is the application of tagging to network flows to accomplish these goals. Specifically, each TCP flow can have several tags associated with it that are then used to keep context, generate reports, and enable collaboration. Tags enable collaboration by allowing fellow researchers to quickly find attacks with desired tags and allows members with different backgrounds to comment. The secondary contribution is the novel querying interface for selecting flows of interests. This allows the analyst to graphically identify and select flows, represented as lines, using rectangular selections without textual queries.

2.1 Interface Description

There are six screen elements on the FlowTag interface: the flow table, flow tags, payload viewer, connection visualization, selection sliders, and active tags (seen in Figure 2). The flow table lists flows in order of occurrence along with any tags associated with the flow. The flow tags entry box allows the assignment of tags to a flow. The payload viewer displays the reconstructed contents of the flow. Using Ethereal, the analyst would have to select each flow, filter (taking a considerable amount of time), view the payload, and then clear the filter (again, taking a considerable amount of time). With FlowTag, each flow has the indices of each associated packet from the pcap file, allowing quick viewing of the reconstructed payloads of each flow. The connection visualization is discussed in Section 2.2. The selection sliders are fairly standard double-ended sliders except that the numerical sliders for bytes and packets work on a logarithmic scale, allowing fine control on smaller values and course control on higher values (Figure 3). Lastly, the tags view lists all the current tags used for the entire file and allow filtering for flows containing the selected tags.

2.2 Connection Visualization and Querying

The connection visualization displays the flows of the pcap file on a parallel coordinate graph [4] with the left axis as the destina-

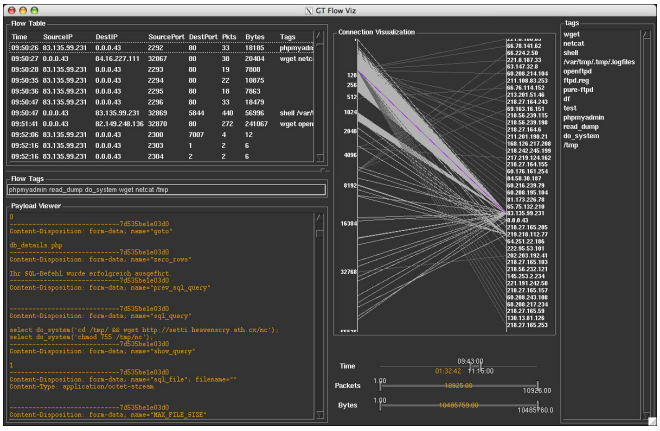


Figure 2: FlowTag main screen

Figure 3: FlowTag double-ended slider filters

tion port and the right axis as the source IP, a technique from [3] and used in [5, 6]. The right axis of Figure 4 is in order of appearance to treat the IP space as nominal rather than quantitative data. This mapping was intentional because time locality is more important to the analysis than the actual IP value. However, FlowTag also supports numerical value mapping of the IP axis. The lines between the axes represent flows between the source IPs and the destination ports. The 0.0.0.0/24 subnet is used to denote flows originating from our honeynet. All other flows originate from outside the honeynet. This method was chosen over using colors or another axis, because color maps to selection state and another axis would use a lot of screen space for a relatively sparse set of values.

The color of line shows its selection state. The currently selected flow (shown in the payload viewer) is purple. Gold represents a flow selected by the mouse and not filtered out by the sliders. Two shades of grey are used to represent whether flows are filtered in or out by the sliders. The darker grey represents flows that are filtered out.

With FlowTag's connection visualization, flows of interest to the analyst are easily found and selected. Simple queries concerning a single IP or single port can be done by selecting the region containing the IP or port as shown with selections 1 and 2 in Figure 4b. Composing queries with a range of IPs or ports is just as simple. Furthermore, selecting a specific IP-port pair has the same effort as in selection 3. You can specify queries that would not be easily done in Ethereal where you can select multiple IP-port pairs as in selection 4. Using the shift key, you can quickly combine regions to form selections of interesting flows with very little effort and no typographical error. Also, since the visual model is used to form the query, less memory strain is needed to store and recall the details of the flows being queried. Since the flows are already indexed by the FlowTag tool, switching between flows is quick. Ethereal is very slow at working with flows, as its strengths lie in packet-based processing.

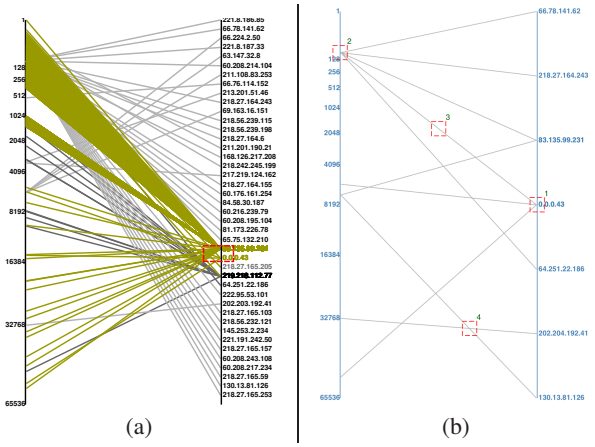


Figure 4: (a) FlowTag connection visualization of phpMyAdmin attack, and (b) FlowTag Connection Visualization Selection Examples

2.3 Tagging (for fun and profit)

2.3.1 Tagging for Understanding

By tagging flows while in the process of analysis, analysts can maintain context without having to recall details or commit to a specific taxonomy. Using Ethereal, they have to follow a TCP flow, note what was in the flow along with the frame number of the packets, then clear the filter for the flow to display all the packets, and find the previous or next flow (using frame numbers) in hopes to find a relationship between the flows. By providing flow-based tagging as part of the interface, analysts can quickly label key elements inside the flows and search for related flows without losing context. Furthermore, tagging has a low cognitive load compared to hierarchical categorization, allowing quick labeling of properties in the flow. The tags assigned to a flow will then be included in the report generated by FlowTag.

2.3.2 Tagging for Reporting

After an attack has been analyzed, a report is needed to summarize the findings and disseminate the information. FlowTag exports a pcap file containing only the packets of tagged flows and automatically generates a report. The report template contains fields for important meta-data along with entries for all the tagged flows using YAML syntax for ease of human editing and machine reading. Each flow entry lists the flow id, source and destination IP and ports, and the list of associated tags. The following example report template, Report 1, was generated from an attack on the Georgia Tech Honeynet, where the attacker used a phpMyAdmin read_dump exploit to download netcat, export a shell, and install an ftp server. We describe this attack in greater detail in Section 3.1.

2.3.3 Tagging for Sharing/Collaboration

After the analyst has filled out a report and exported the packets of interest into a separate pcap file, the report and pcap file can be uploaded to the Honeynet Alliance Attack Library and shared with the research community. This website is very similar to sites like YouTube, where people have profiles with their uploaded attacks and others can download and comment on the attacks. The tags submitted in the report would be associated with the attack package and would instantly be searchable via the tag database. This reduces the barrier of entry for contributing to the community allowing novices to contribute to the larger community. Experts can

Report 1 phpMyAdmin Attack Report Template

```
date  : 2006-04-29
author: chris
os    : linux 2.4
notes : <<sysadmin writes overview here>>
flow:
    - flowid : 1
      srcip  : 83.135.99.231
      dstip  : 0.0.0.43
      srcport: 2292
      dstport: 80
      proto  : tcp
      tags   : phpmyadmin read_dump do_system netcat
      description: <<sysadmin adds comments here>>
    - flowid : 2
      srcip  : 0.0.0.43
      dstip  : 84.16.227.111
      srcport: 32867
      dstport: 80
      proto  : tcp
      tags   : wget netcat
      description: <<sysadmin adds comments here>>
    - flowid : 3
      srcip  : 83.135.99.231
      dstip  : 0.0.0.43
      srcport: 2296
      dstport: 80
      tags   : phpmyadmin read_dump do_system netcat bash
      description: <<sysadmin adds comments here>>
    - flowid : 4
...
      tags   : bash /var/tmp/.tmp/.logfiles openftpd ftpd.reg pure-ftpd
      description: <<sysadmin adds comments here>>
...
    - flowid : 13
...
      tags   : ftp login tcp7007 mkd SvCd Tv XViD XxX SpEeD
      description: <<sysadmin adds comments here>>
```

then evaluate the same attack and collaborate by adding extra tags, analysis, or comments.

The tags attached to an attack are also useful for detecting similarities with other attacks. Attacks that use similar exploits, tools, rootkits, or servers should naturally be tagged similarly. These similarities can be monitored to see what kind of attacks are easily observable, well understood, and novel. The research community can then react by creating new sensors, analysis tools, and trend models. This will be one of the main motivations for sharing.

There are four motivations for sharing attack captures: personal organization, expert collaboration, recognition, and trend monitoring. The Honeynet Alliance requires a biannual report detailing attacks, discoveries, and trends. By storing this information on the community website, this information is already aggregated, organized, and shared. Collaboration allows researchers with various points of views and experiences to help organize attacks by tagging and to help analyze attacks by commenting. This also promotes mentoring and this system allows them to be recognized for their stewardship, which is rewarded in the Honeynet Alliance. Lastly, as stated in the previous paragraph, uploading attacks will allow for a better understanding of what attacks are observable, well understood, and novel.

3. EXAMPLE ATTACKS AND ANALYSIS

To illustrate the efficacy of the FlowTag tool for analysis and report writing, we show the analysis steps for three separate attacks on the Georgia Tech Honeynet. The attacks are titled according to the exploit vector followed by the motive of the attacker. Due to space limitations, we cannot fully describe the analysis process, but we hope to show that the combination of tagging and visualization allows for rapid analysis on a variety of attacks.

3.1 phpMyAdmin - Warez

This was the first attack after the initial version of FlowTag, which motivated a lot of the features that were developed. Refer to Figure 4a and Report 1 for this attack. The first outbound flow from the honeypot, Flow 2, did a wget for the netcat binary. We used the time slider to filter the flows to just before and after outbound flows. The flow immediately prior, Flow 1, used a known phpMyAdmin read_dump.php flaw to execute a do_system SQL command and downloaded the netcat binary with wget. The next outbound flow, Flow 4, contained shell interactions with the attacker. Again, the attacker used the phpMyAdmin exploit in Flow 3 to export a bash shell using netcat.

Now that the compromise is understood, what did the attacker do with the honeypot? In the bash shell, the attacker used wget to download an FTP server and a configuration file that places the ftp server on port 7007. Throughout the next three days, movies, music, and test files were uploaded and downloaded from the FTP server. Although the attacker could have used this honeypot for a bot, spamming, or other nefarious means, it is evident that their interest was in warez sharing.

3.2 Samba - Botnet

The next attack used an LSASS exploit in Windows XP SP1 to make a zombie (bot) out of the honeypot and then immediately started scanning all of the /24, then the /16. The connection visualization for this attack is down in Figure 5a. The samba exploit opened port 7115 on the victim and the attacker downloaded a W32 binary, which was later found to be the Trojan.SdBot-724 via ClamAV. The capture file was over 48 MB due to the amount of scanning (>113,000 IPs) and takes over three minutes to open in Ethereal due to packet parsing (and this delay is repeated each time a filter is allied or removed). FlowTag opened the same file in slightly less than three minutes the first time, but it generates a flow database that loads in 21 seconds and allows instant transitions between flows in the interface.

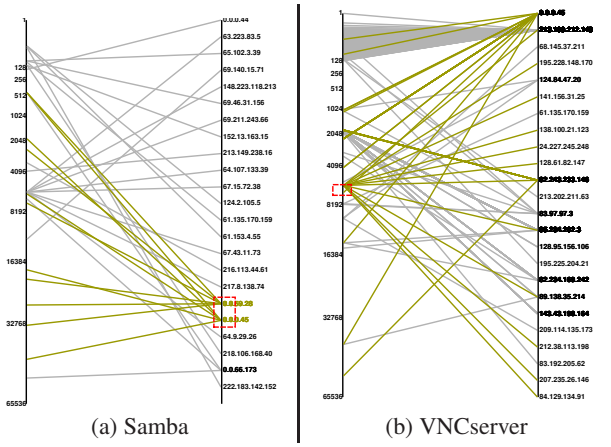
The first outbound flow in Report 2, Flow 3, was a join to an IRC channel, `#dlrowymx0ri`. From there the previous inbound flows, Flow 1 & 2, were the exploit and the malware download. There is more after Flow 3, but we stop here for brevity. After the flows were tagged, a report was generated and sent to Georgia Tech's Office of Information Technology so they could respond to any infected PC found on campus. The turn around time was less than 20 minutes from the alert until the report.

Report 2 Samba Attack Report Template

```
date  : 2006-05-19
author: chris
os    : WinXP SP1
notes : <<sysadmin writes overview here>>
flow:
    - flowid : 1
      srcip  : 0.0.59.28
      dstip  : 0.0.0.45
      srcport: 4829
      dstport: 445
      tags   : SMB samba exploit
    - flowid : 2
      srcip  : 0.0.59.28
      dstip  : 0.0.0.45
      srcport: 4965
      dstport: 7115
      tags   : W32 binary malware
    - flowid : 3
      srcip  : 0.0.0.45
      dstip  : 218.61.146.86
      srcport: 1027
      dstport: 18067
      tags   : irc dlrowymx0ri 83.133.126.75
```

3.3 VNCserver - Warez/Botnet

After the release of the exploit code for the RealVNC server exploit appeared on Metasploit, port 5900 scans spiked. In reaction, we placed a VNC server on a Win2K honeypot. Within 24 hours, the machine was compromised four times. Each attack was different, but again for brevity, we describe only the first one here (also highlighted in Figure 5b). The first outbound flow in Report 3,



(a) Samba (b) VNCserver

Figure 5: Attack connection views

Flow 2, was an ftp connection that downloaded a batch file (Flow 4), a configuration file (Flow 5), and the Serv-U ftp server software (Flow 6). The attacker connected again via VNC and proceeded to rootkit the box and connect it to an IRC channel. The IRC channel would then private message our host, authenticate, and then send filenames for our host to download. Throughout the night, various people connected via FTP to download files. Again, the primary objective of this attack was to setup a warez server.

Report 3 VNCserver Attack Report Template

```
date  : 2006-06-08
author: chris
os    : Win2K
notes : <<sysadmin writes overview here>>
flow:
    - flowid : 1
      srcip  : 82.243.233.146
      dstip  : 0.0.0.45
      srcport: 1075
      dstport: 5900
      tags   : vnc exploit
    - flowid : 2
      srcip  : 0.0.0.45
      dstip  : 82.243.233.146
      srcport: 1835
      dstport: 21
      tags   : ftp session dows.bat KB-91-5-215.ini servu SERVU.exe
...
    - flowid : 6
      srcip  : 82.243.233.146
      dstip  : 0.0.0.45
      srcport: 20
      dstport: 1839
      tags   : SERVU.exe
```

4. CONCLUSION

FlowTag demonstrates the value of adding tagging to a flow-based tool for use in analysis, reporting, and sharing. The connection visualization in FlowTag gives an overview of flows contained within a network capture file and allows quick queries for interesting flows to show in the flow table for detail. The combination of tagging and querying reduces analysis time for attacks by keeping context with tags and switching between related flows rapidly. We show the reports generated from three attacks.The tags can then be used to generate reports and export "attack packages" for sharing on the community site allowing further collaboration and communication among the security research community.

5. FUTURE DIRECTIONS

We would like to see research on automated detection of attack profile changes based on changes in the library's tag database. This has interesting challenges in tagging errors, changes in detection methods (e.g., adding honeyclients), and in associating tags with each other (e.g., `sdbot` and `rdbot` are both `botnet`). This could be combined with port statistics from Dshield. This would need to scale to allow the internet community at large to contribute and benefit. We would also be interested in working with automatic malware characterization researchers in combining attack captures and malware into a unified collaborative environment. This would involve multiple collaborative "views" on the items to show the relationships between the exploits and actions of malware and what was observed in attack captures.

6. REFERENCES

[1] D. Barreau and B. A. Nardi. Finding and reminding: file organization from the desktop. *SIGCHI Bull.*, 27(3):39–43, 1995.

[2] C. Cattuto, V. Loreto, and L. Pietronero. Collaborative tagging and semiotic dynamics, 2006.

[3] G. Conti and K. Abdullah. Passive visual fingerprinting of network attack tools. In *VizSEC/DMSEC '04*, pages 45–54, New York, NY, USA, 2004. ACM Press.

[4] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. pages 361–378, 1990.

[5] S. Krasser, G. Conti, J. Grizzard, J. Gribshaw, and H. Owen. Real-time and forensic network data analysis using animated and coordinated visualization. In *2005 IEEE Workshop on Information Assurance*. IEEE Press, 2005.

[6] C. P. Lee, J. Trost, N. Gibbs, R. Beyah, and J. A. Copeland. Visual firewall: Real-time network security monitoring. In *Visualization for Computer Security, IEEE Workshops on*, pages 16–16, 2005.

[7] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata.

[8] Quintarelli. Folksonomies: power to the people. `http://www-dimat.unipv.it/biblio/isko/doc/folksonomies.htm`.

[9] P. Ravasio, S. G. Schär, and H. Krueger. In pursuit of desktop evolution: User problems and practices with modern desktop systems. *ACM Trans. Comput.-Hum. Interact.*, 11(2):156–180, 2004.

[10] R. Sinha. A cognitive analysis of tagging. `http://www.rashmisinha.com/archives/05_09/tagging-cognitive.html`.

[11] E. Tonkin. Folksonomies: The fall and rise of plain-text tagging. `http://www.ariadne.ac.uk/issue47/tonkin/intro.html`.

[12] J. Voss. Collaborative thesaurus tagging the wikipedia way, 2006.