Filtering, Fusion and Dynamic Information Presentation: Towards a General Information Firewall

Gregory Conti[1], Mustaque Ahamad[1] and Robert Norback[1]

[1] Georgia Tech Information Security Center, Georgia Institute of Technology,
801 Atlantic Avenue, Atlanta, Georgia, 30332-0280 USA
http://www.gtisc.gatech.edu/

Abstract. Intelligence analysts are flooded with massive amounts of data from a multitude of sources and in many formats. From this raw data they attempt to gain insight that will provide decision makers with the right information at the right time. Data quality varies from very high quality data generated by reputable sources to misleading and very low quality data generated by malicious entities. Disparate organizations and databases, global collection networks and international language differences further hamper the analyst's job. We present a web based information firewall to help counter these problems. It allows analysts to collaboratively customize web content by the creation and sharing of dynamic knowledge-based user interfaces that greatly improve data quality, and hence analyst effectiveness, through filtering, fusion and dynamic transformation techniques. Our results indicate that this approach is not only effective, but will scale to support large entities within the Intelligence Community.

1 Introduction

Intelligence analysts are besieged with data from legitimate sources and the problem is compounded by active malicious entities attempting to subvert their work by injecting misleading or incorrect data into their information space. From this sea of data, analysts attempt to glean useful information that meets the intelligence needs of their customers. While exact statistics on the problem are classified, it is useful to consider the amount of open source data being generated. The recent University of California, Berkeley study on how much information is created each year clearly illustrates the problem [1]:

- In 2002, about 5 exabytes of new information was created in print, film, magnetic and optical formats. Five exabytes is equivalent to 37,000 times the size of the United States Library of Congress book collection or 800 megabytes per person based on the world population.
- From 1999 to 2002 information in these formats grew at a rate of 30% per year.
- Ninety-two percent of this information was stored on magnetic media.

The rate at which data is being produced, combined with the immense amount of existing data, sets the stage for denial of information attacks against both analysts and their customers. Denial of Information (DoI) attacks are similar to Denial of Service (DoS) attacks against machines [2]. While DoS attacks attempt to deny users access to system resources by consuming machine resources, DoI attacks target the human by exceeding their perceptual, cognitive and motor capabilities. In most cases, a small amount of malicious information is all that is required to overwhelm or deceive the human. A successful DoI attack occurs when the human does or does not take action they otherwise would have [3]. Denial of Information attacks are of critical importance to intelligence analysts. Every bit of time, albeit small, wasted on a false lead or due to information overload reduces the probability of timely and accurate intelligence. To counter Denial of Information attacks against analysts we employed collaborative, knowledge-based user interfaces that improve data quality. These interfaces, based upon filtering, fusion and dynamic transformation techniques, reduce the amount of irrelevant data (noise) and increase the useful information (signal) presented to the analyst. The system will support the following situations:

- Filtering unneeded information based on other analysts' and customers' experiences.
- Fusing multiple information sources into a single consolidated page.
- Transforming poorly designed information architectures and interfaces into far more usable ones.
- Sharing of transforms via simple techniques such as browsing an index or emailing a link to a colleague.

The system we present contains the following actors:

Information producers: Information producers may be either internal or external to the organization and provide data via web pages to information consumers. These information producers may be untrusted sources on the commercial Internet or trusted sources such as other analysts, offices of primary interest or operational intelligence systems within an organizational intranet.

Meta-information producers: Meta-information producers develop interface and data transforms which embed knowledge and filter, fuse and transform the data generated by information producers. These transforms are then shared via easily accessible search engines and indices.

Information consumers: Information consumers are intelligence analysts as well as intelligence customers who search for and utilize the information available.

The following example, shown in Figure 1, demonstrates the operation of the system. In step one, an information producer generates a large web page of moderate quality data with a poorly designed interface. Note that the page would require the analyst to scroll through about eight screens of information. The page is rife with links to

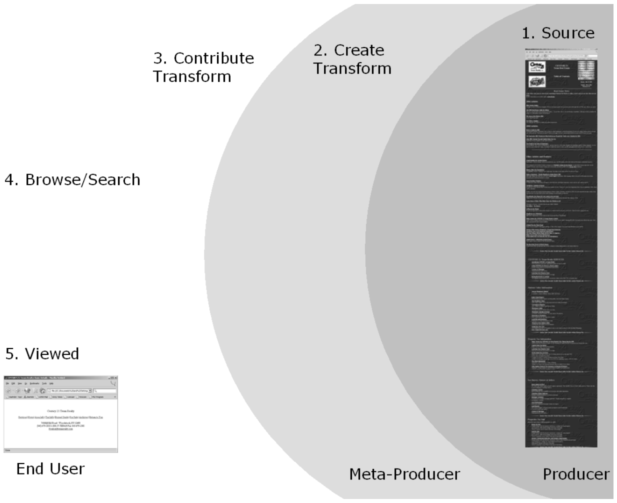


Fig. 1. System Operation: Information producer generates large, low quality web page. Through a transformation process, the end user receives much higher quality information and a far more usable interface

irrelevant information (such as advertisements) and is constructed with poorly chosen foreground and background colors. In step two, an analyst creates a transform for the page which dramatically improves the quality of information and interface. This transform is then shared via a centralized transform server in step three. In step four, other users, both intelligence customers and other analysts, can then browse/search the server for relevant transforms. These results contain ratings based on previous users' experiences as well as descriptions of the transforms. After a consumer selects a transform, step five, they see a dramatically altered version of the page. The result is a significantly more usable page with much higher information gain. The interface problems have been corrected and the page shows just the relevant information, but there is a link to the original source information, if required. The end user may then create a new transform or modify the existing transform and submit it to the server. In addition, the user may help collaboratively filter the transform itself by voting on its quality and usefulness. The cycle continues with new consumers utilizing and improving upon the transforms.

The system was implemented using open-source software (Apache[4], Mozilla[5] and Perl[6]) on commercial off the shelf (COTS) hardware. The ubiquitous Hypertext Transfer Protocol (HTTP) mediated client-server interactions. The combination of open-source software and COTS hardware facilitated low-cost implementation.

For this work, we assumed that analysts will operate primarily within private intranets. This assumption greatly reduces the legal implications of filtering content without the permission of the information producer. In particular, the act of removing paid advertisements from content is of questionable legality, even by intelligence analysts employed by government organizations. To avoid classification concerns, we tested the system using unmodified open source web pages. As we explore the broader applicability of the system, it is important to consider the threat model. Malicious meta-information producers could generate transforms designed to mislead or filter information from legitimate customers. In our work, we assumed that transforms would only be generated by trusted members of the organization. While not all transforms meet the needs of all customers, we believe, that given this assumption, the combination of filter descriptions, direct links to original unmodified source material and collaborative rankings of transforms will allow end users to find and utilize the transforms that they need.

Section two of this paper discusses related research and places our work in the current field. Section three presents our system model and architecture. Section four describes and analyzes our results. Section five presents our conclusions and directions for future work.

2 Related Work

The uniqueness of this work springs from the distributed and collaborative approach to increase the quality of information accessed from data sources in order to provide better products to analysts and intelligence customers. We facilitate this human assisted collaborative analysis by incorporating the insights of both analysts *and* customers in the analytic loop. While the underlying technologies will change, the collaborative fusion, filtering and interface transforming approaches we present will be far more enduring. Communities of analysts and their customers can iteratively improve the quality of intelligence by creating and sharing dynamic user interfaces and information transforms based upon their tasks and needs.

The Galaxy of News system designed by Rennison is most directly applicable to our work [7]. The system employs visualization techniques and a relationship construction engine to build implicit links between related, but independently authored news articles. While we were influenced by this work, our work differs significantly. The primary differences are in the following areas. Galaxy of news relies upon the relationship construction engine to build links between news articles. Our work focuses instead upon using human moderated, collaborative transforms to increase the information quality of individual articles and to fuse together disparate information sources to meet customized user needs. In addition, we incorporate the notion of trust through the use of user rankings and links to original source content. Also, there exists a large body of work in the domain of web usability. See the books by Nielsen for excellent examples [8,9]. These works describe best practices and design techniques to create more usable and information rich web pages, but were designed for individual web designers to apply to their *own* content. Our system provides the mechanism for users to apply, share and improve upon the *work of others* to meet their own specific information needs. Given our assumption that transforms are only created by trusted members of the intelligence organization, we believe that our collaborative ranking system and link to the source document is sufficient to protect users from unintentional, badly designed transforms. In some instances this assumption may not be valid as these measures can be attacked by malicious entities. For those interested in more robust defenses, Levin's dissertation on attack resistant trust metrics discusses many relevant issues and is an excellent starting point [10].

There have been several approaches, both centralized and decentralized, to filtering web content. The primary difference with our work is in the ability to collaboratively share, rate and improve upon filters created by others. The Google Mirror is representative of the centralized approach [11]. It uses a proxy to rewrite content returned from the Google search engine. It does not attempt to add value or increase the information quality of the search results. BugMeNot [12] is another interesting centralized approach. The website seeks to increase information quality and access by bypassing login procedures through communally shared user ID's and passwords. While the overlap is minimal with our work, the sharing of access credentials to increase information access, avoid advertisements and protect privacy is worthy of examination. DOM Inspector [13] and Adblock [14] illustrate the current state of the art in the decentralized, browser, plug-in approach. DOM Inspector allows a user to view the document object module [15] of HTML pages and to selectively filter based upon object type. The Adblock plug-in gives the user the ability to block advertisements with far greater precision than Mozilla's default image blocker. Our work further extends this notion by providing the additional ability to fuse multiple datasources, filter with higher resolution and dynamically alter the interface and information architecture of websites. Real Simple Syndication (RSS) is a push technology, incorporating eXtensible Markup Language (XML), that distributes news and other new content from websites to virtually any client platform. Clients can subscribe to various streams of interest and tools exist to fuse together multiple streams to form a single consolidated picture. It is important to consider the domain of intelligent information retrieval such as distributed retrieval, search strategies, semantic filtering, content indexing and information discovery. We believe that current and projected techniques in this area complement our work and can be incorporated, as applicable, into the information firewall architecture that we propose. Finally, two other classes of technology merit discussion: HTTP proxy content rewriting and inline content transforming devices. Representive of proxy content rewriting is Privoxy [16]. Its primary focus differs from ours in that it transforms content to remove advertisements and protect privacy of individual users. WebWasher [17] offers a suite of tools and inline appliance that transforms information streams between organizational users and external data sources. WebWasher is designed to protect enterprise networks from frivolous use and malware. It is typically used in an adversarial manner to enforce organizational policy on users.

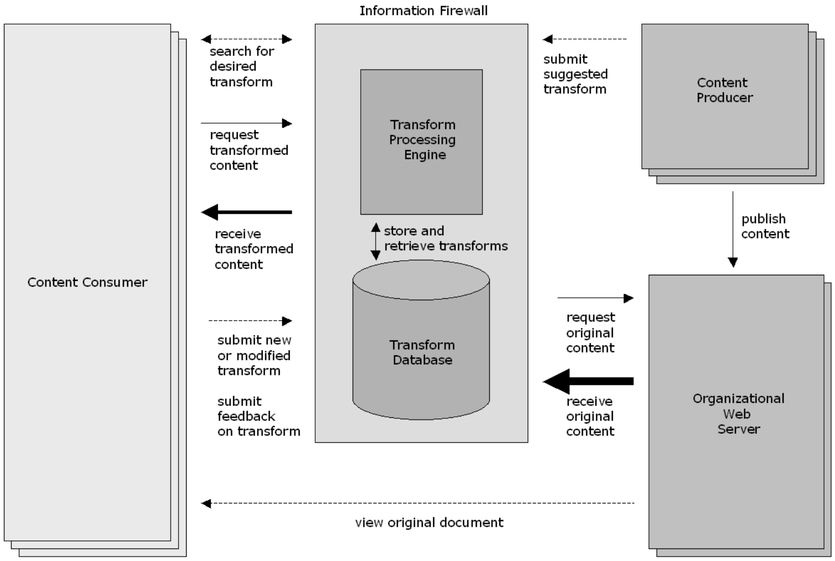

Information Firewall
Content Consumer
search for desired transform
request transformed content
receive transformed content
submit new or modified transform
submit feedback on transform
Transform Processing Engine
store and retrieve transforms
Transform Database
submit suggested transform
Content Producer
publish content
request original content
receive original content
Organizational Web Server
view original document

Fig. 2. System Architecture: Information producers generate content and optionally submit suggested transforms to transform database. Information consumers request and receive transformed content via the transform processing engine. Information consumers optionally submit new or modified transforms, provide feedback on transform quality or request the original source document

3. Information Firewall Design and Implementation

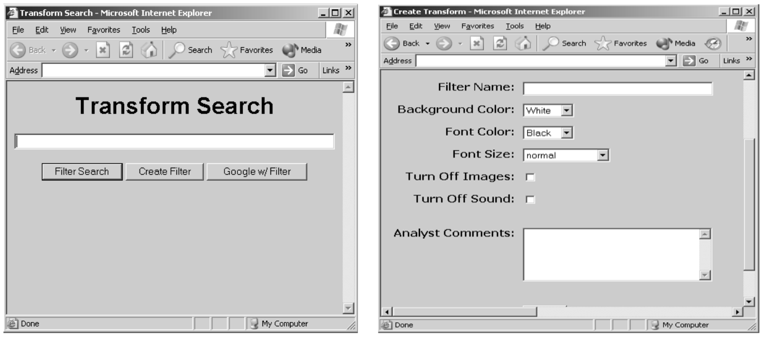
3.1 Design Goals

The design of the system is based upon the analyst's need to acquire the highest quality information with a minimum of effort. Our core users are intelligence analysts, but they may also be any consumers of information. Hence, the primary design goal was to maximize the valuable information contained in web content while reducing unwanted information. We include in this definition the ability to modify the information interface and navigation structure to assist in improving the task-specific interaction and presentation of information. In other words, we wish to increase signal while decreasing noise such that the information is presented in a format and information architecture desired by users. It is important to note that the definition of valuable information will vary from analyst to analyst. Because of this need, our second design goal was to allow information consumers and producers to create, share and collaboratively rank information transforms. This capability affords information producers the opportunity to create initial transforms based on their best interpretation of customer requirements. Their end users could then use these transforms as is, or modify and layer them to meet their specific needs as well as share the resulting transform with other users. To allow efficient and effective sharing, our third design goal was to create centralized communities of trust where users could easily seek out and contribute transforms. To support decentralized sharing we wished to reduce the routine usage complexity of information producer-to-analyst and analyst-to-analyst sharing to the order of emailing a hyperlink or bookmarking the transform in a common browser. Transforming the content of information used by analysts runs the risk of masking important information. To counter this effect, our fourth design goal was to provide the analyst with easy access to the original source content.

3.2 System Architecture

The following sections describe the information flow and usage of the system including content production and transform search, creation, execution, interaction and sharing. Figure 2. provides a graphical representation of this architecture.

Content Production: Use of the system begins with analytic organizations generating and publishing content to network accessible web servers. This content can be in any web accessible format, but for purposes of our experimentation we focused exclusively on HTML documents. Optionally information producers will publish a variety of suggested transforms based on their perception of analyst and other customer requirements.


Transform Search - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address Go Links
Transform Search
Filter Search
Create Filter
Google w/ Filter
Done
My Computer
Create Transform - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address Go Links
Done
My Computer

Filter Name:
Background Color: White
Font Color: Black
Font Size: normal
Turn Off Images: □
Turn Off Sound: □
Analyst Comments:

Fig. 3. Transform Search and Creation: Analysts are presented with a clean interface to search for, create and use transforms (left). Creating a simple general-purpose static filter (right)

Transform Search and Creation: In order to gain the benefits provided by the information firewall, analysts typically do not access content directly. Instead, they perform a search of available information transforms on a web server hosted by the information firewall, Figure 3(left). This search is conducted based on the website name, URL and transform name. (using the leftmost button in the figure) The search index returns transforms that match any of these values and are ordered alphabetically and by user rating. Alternatively, the analyst may depress the rightmost button and perform a general search of the Internet using the Google search engine. The transform engine, acting as a proxy, sends the search to Google and receives the unmodified results. These results are modified by the transform engine to include applicable transform links associated with each search result before presentation to the user. For example, if the search was for newswebsite.com, the transform engine would add transform links associated with that URL to the Google search results returned to the user. These might include transforms such as "top stories," "international news" or "ad free news." If an analyst is unable to find a transform providing the desired functionality they have the option of creating a new transform using the center button. A simple filter creation screen is show in Figure 3(right). Upon submission, this filter is added to the transform database with an initial feedback rating of neutral.

Transform Execution and Interaction:
In general, transforms take *any* information object as input and convert it to another information object. The rules to make the conversion are created by the user with the aid of the tools provided by the information firewall. After creation, the rules are stored in the information firewall's database. Figure 3 (right) shows a simple example. The user selects options they would like in a transform and registers the transform with the information firewall which then stores the parameters in its database. In this case, it is a web page filter that removes undesired information from the page as well as changes interface parameters to present information in the desired way. While, in our current prototype, we implemented a limited number of transforms, we believe this to be a powerful concept. In any instance where there exists a reasonable algorithmic solution to convert one information object to another this algorithm may be included in the firewall as the basis of a transform. Human end users may then interact with these algorithms to create transforms which customize the information they receive.

Transforms are executed by selecting a hypertext link that calls a CGI script on the transform engine. This link includes a numeric parameter representing the transform to execute as well as the URL of the source website. First, the transform engine queries the database for details of the transform using the numeric parameter as an index. The transform engine then contacts the URL and requests the content in question. As the content is returned from the URL it strips or adds content and dynamically generates the resulting page for the user. For example, if the user requested no images, the transform engine will remove all image tags from the page. To facilitate trust, all transformed pages include a clearly delineated header that includes a link to the unaltered document and the author of the transform. A small, one-line form in the header allows the user to vote on the usefulness of the transform. Deliberately designed with simplicity in mind, the use of hypertext links to execute transforms is a key aspect of the system. As a result, users can share transforms easily via email and store them quickly as bookmarks.

3.3 System Implementation

The system was implemented entirely with open source / free software, with the exception of our host operating systems, Windows XP. The use of Windows XP was not a mandatory requirement, as the system could easily be implemented on a mainstream open source operating system, such as Linux, because equivalent software is readily available for all components. Information consumers used Microsoft Internet Explorer and the Mozilla Organization's Firefox browsers to view both original and transformed content. These specific browsers were not required for use of the system, any HTTP compliant browser would suffice. Similarly, any HTTP compliant web server would meet the system's requirements. We tested using the Apache web server package as our in-house analytic organizational server as well as conducted further tests using commercial Internet websites with undetermined server software. The transform processing engine was implemented using the Active State Perl package. We used Perl to dynamically create web pages needed by users as they created, modified, executed and searched for transforms. These transforms, including descriptions, were stored in a MySQL database accessed using standard calls from the transform engine's Perl application. Perl and MySQL were not specifically required, any CGI compliant programming language with the ability to interact with a backend database would meet the system's needs. The only, somewhat, non-standard tool that we used was the libwww-perl lwp Perl module [18] to allow the transform engine to easily request web content on behalf of the user. While the libwww-perl suite of tools was designed to provide an easy application programming interface to the World Wide Web, the minimum requirement is a CGI compliant language capable of handling HTTP transactions.

4. Results and Analysis

The information firewall system was designed to help analysts collect, assimilate and explore internal intelligence community and external open source information resources to better execute their responsibilities. To this end, we found the system's overall performance and capabilities to be technically possible, usable, efficient and effective. The quality of information presented to analysts improved dramatically by using the information firewall. Figure 4 illustrates this information gain by filtering extraneous elements from two large and complex web pages and passing the improved result to the information consumer. The following sections discuss the system's ability to meet our design goals as well as additional results and analysis.

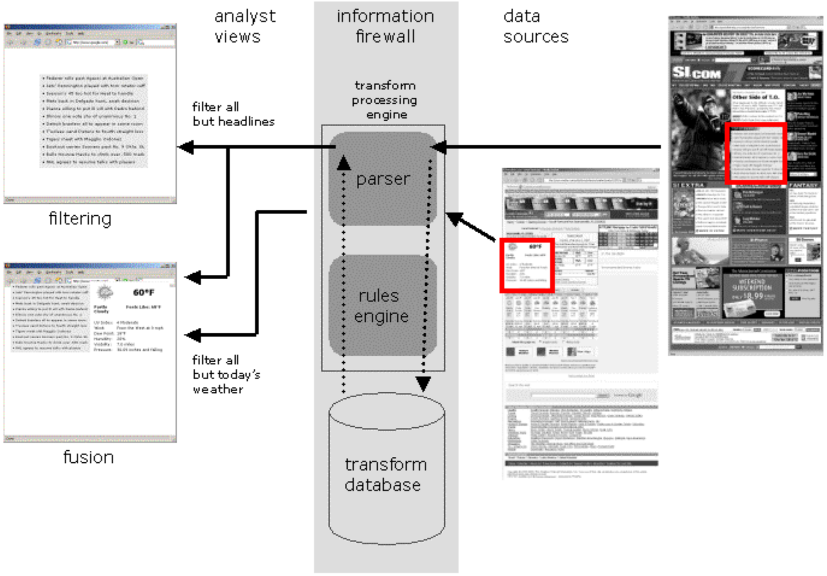


Fig. 4. Evaluation: Complex, noisy web pages are parsed and transformed in the transform processing engine. Pages with far higher information quality are returned to analysts. The system can filter information and optionally fuse together results from multiple data sources as well as modify interface components and navigation architectures

4.1 Information Gain

By coupling the strengths of human decision making with the information processing support provided by the system, we believe the use by analysts to collaboratively transform and improve information content was successful. Our results showed significant information gain in products that contained clear and consistent HTML markup. In these situations, we were able to create rules that transformed content with very fine-grained resolution. We found this to be true even in situations with high information entropy, as long as the structure of the document did not change. In other cases, we found that some dynamically generated pages and those that used significant amounts of JavaScript limited the success of the system. We were still able to transform consistent elements of these pages, but in other instances we were unable to effectively act upon the information, resulting in limited functionality. Ultimately, we believe the success of the system depends on the ability to reliably parse and semantically analyze information products. Adversarial information producers could actively frustrate the effectiveness of the system. In future versions of the system, we plan to allow users to submit feedback on the quality of content provided by information producers in addition to feedback on transform producers. We believe that these metrics, in combination, will strongly assist analysts in evaluating their source material.

4.2 Customer Feedback

During evaluation we were surprised to find that a clear pictured developed of desired and undesired information from the viewpoint of customers. By searching for transforms and examining their ratings, information producers could receive very clear feedback about which elements of their products were of value to customers. Developing satisfaction metrics and garnering feedback from intelligence customers is a continuing concern for all analytic organizations and we view this as a very promising advantage of the system. By using this feedback, information producers could reevaluate and adjust their intelligence products to better meet the needs of their customers while sparing their own organizations additional work.

4.3 System Usability

Our test users were pleased with the usability of the system. In particular, the simplicity of utilizing hyperlinks to request transformed information via the firewall proved to be quite effective. Our users liked the ability to share transforms via email, bookmark them in browsers and to use their browser's history function to recall recently used filters. We designed our application interfaces using web design best practices and our feedback indicated that our clean and simple approach was successful. In particular, we found that the inclusion of a simple header allowed users to quickly recognize they were viewing modified content, access original content and stimulated satisfaction feedback. While, technically, the system is capable of using high-resolution filters and transforming interfaces our initial work indicates that a more intuitive tool is required to streamline the process and help create task specific transforms. We believe a visual transform editing tool would significantly ease construction.

4.4 Privacy and Trust

As we built and evaluated the prototype system, we discovered several subtle interactions between privacy and trust. In an effort to build trust among users, we included the transform author's name as publicly available information. While our results indicated that this would build trust, we are concerned that, in bureaucratic organizations, office politics might prevent users from building the most candid and useful transforms for fear of political sensitivities. In the future, we plan to allow users to create pseudonyms or even anonymously post transforms. In this case, we believe the transform feedback system will provide sufficient protection from poorly designed transforms. This notion could be strengthened by the inclusion of a feedback system for transform authors, in addition to the current feedback system for individual transforms. Consider the feedback system used on Ebay [19]. On Ebay, every user is known only by a pseudonym, but the feedback system provides very insightful summary statistics on transaction satisfaction as well as individual comments for each transaction. The end result is a community of trust that succeeds despite a great deal of potential risk.

4.5 System Implementation

While our prototype was not put under the same stress as a production system, we believe the general approach is sound. Our desire to keep costs low by the use of open source software and COTS hardware was entirely successful in the current instantiation of the system. Clearly the centralized approach we present, will not scale well for very large systems. All information content is currently processed through a centralized information firewall. Despite this, we believe the underlying principles to be sound and plan to explore a decentralized approach using browser plug-ins. In essence, the information firewall would then exist at the client's personal computer which would handle the bulk of all information processing and communication.

5. Conclusions and Future Work

The information firewall we present is viable and useful within the Intelligence Community. It improves the efficiency and effectiveness of analysts by dramatically increasing the signal to noise ratio of intelligence data thereby easing the cognitive burden placed upon intelligence analysts as well as intelligence consumers. While the information firewall concept proved to be effective in the constrained environment of intelligence organization intranets, we plan to explore decentralized approaches to improve scalability. In particular, we believe that browser-based plug-ins combined with both peer-to-peer and centralized sharing of transforms will greatly increase the performance and flexibility of the system. In addition to intelligence community applications, we wish to apply the approach to the larger commercial Internet. For this to be feasible, we must explore the legal ramifications of filtering, fusing and transforming data with and without the permission of the information source. Finally, we plan on extending our work to include a generic information firewall for all digital content. For this to be feasible, we must explore ways to better access the embedded knowledge within available data using such technologies as XML. We envision that the information transformation techniques presented in this paper will work extremely well in a variety of additional applications including increasing accessibility, e.g. via custom presentation to color-blind or vision impaired users, conversion from text to audio, streamlined language translation and web-based intelligence monitoring and analysis. We also believe that the transformation of information and interfaces will allow use on a wide variety of computing platforms, including very small devices such as personal digital assistants and those with severe bandwidth constraints. The ultimate strength of the system lies in the ease with which individual analysts may create robust, high-resolution information transforms. In the future we plan to investigate intuitive tools to support transform construction, perhaps using the visual web page editing paradigm. Finally, we plan a formal usage study in an unclassified operational or training environment to gather additional feedback from analysts. Ultimately, this study would include assessment of information quality gain using best practice metrics from the information quality community.

Acknowledgments

We would like to thank Sandip Agarwala, David Dagon and Sven Krasser of the Georgia Institute of Technology for their thoughtful review and comment as well as Lieutenant Colonel Ron Dodge and the United States Military Academy's Information Technology and Operations Center (www.itoc.usma.edu) for their continual support.

References

1. Lyman, Peter and Varian, Hal. How Much Information 2003. http://www.sims.berkeley.edu /how-much-info-2003, last accessed on 1 January 2005.

2. Computer Emergency Response Team Coordination Center (CERT/CC). Denial of Service Attacks. http://www.cert.org/tech_tips/denial_of_service.html, last accessed on 2 January 2005.

3. Conti, Gregory and Ahamad, Mustaque. Countering Denial of Information Attacks. IEEE Security and Privacy. (to be published)

4. The Apache Software Foundation. HTTP Server Project. http://httpd.apache.org/, last accessed on 3 January 2005.

5. The Mozilla Organization. Firefox Web Browser. http://www.mozilla.org/products/firefox/, last accessed on 3 January 2005.

6. Active State. Active Perl. http://www.activestate.com /Products/ActivePerl/, last accessed on 3 January 2005.

7. Rennison, Earl. Galaxy of News: An Approach to Visualizing and Understanding Expansive News Landscapes. Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology, 1994, pp. 3 - 12.

8. Nielsen, Jakob. Designing Web Usability. New Riders, 2000.

9. Nielsen, Jakob. Homepage Usability: Fifty Websites Deconstructed. New Riders, 2002.

10. Levin, Raph. Attack Resistant Trust Metrics. Ph.D. thesis. University of California: Berkeley. Available online at http://www.levien.com/. Last accessed on 3 January 2005.

11. Google Mirror. http://www.alltooflat.com/geeky /elgoog/info/, last accessed on 3 January 2005.

12. Bug Me Not. Frequently Asked Questions, http://www.bugmenot.com/faq.php, last accessed on 3 January 2005.

13. The Mozilla Organization. DOM Inspector. http://www.mozilla.org/projects/inspector/, last accessed on 3 January 2005.

14. The Mozilla Organization. The Adblock Project. http://adblock.mozdev.org/, last accessed on 3 January 2005.

15. World Wide Web Consortium. Document Object Model. http://www.w3.org/DOM/, last accessed on 3 January 2005.

16. Privoxy Project Homepage. http://www.privoxy.org/, last accessed on 3 January 2005.

17. Webwasher. Webwasher CSM Appliance and Suite. http://www.webwasher.com/, last accessed on 3 January 2005.

18. libwww-perl Project Page. http://lwp.linpro.no/lwp/, last accessed on 30 January 2005.

19. Ebay Feedback. http://pages.ebay.com/services/forum/feedback.html, last accessed on 30 January 2005.

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Military Academy, the Department of the Army, the Department of Defense or the U.S. Government.

This work was supported in part by the National Science Foundation Information Technology Research award 0121643.