Towards Efficient Collaboration in Cyber Security

Peter Hui Joe Bruce Glenn Fink
Michelle Gregory Daniel Best Liam McGrath
Pacific Northwest National Laboratory
{peter.hui, joseph.bruce, glenn.fink, michelle.gregory, daniel.best, liam.mcgrath}@pnl.gov

Alex Endert
Virginia Polytechnic Institute and State University
aendert@cs.vt.edu

ABSTRACT

Cyber security analysts in different geographical and organizational domains are often largely tasked with similar duties, albeit with domain-specific variations. These analysts necessarily perform much of the same work independently— for instance, analyzing the same list of security bulletins released by largely the same set of software vendors. As such, communication and collaboration between such analysts would be mutually beneficial to the analysts involved, potentially reducing redundancy and offering the opportunity to preemptively alert each other to high-severity security alerts in a more timely fashion. However, several barriers to practical and efficient collaboration exist, and consequently, no such framework exists to support these efforts. In this paper, we discuss the inherent difficulties which make efficient collaboration between cyber security analysts a difficult goal to achieve. We discuss preliminary ideas and concepts towards a collaborative cyber-security framework currently under development, whose goal is to facilitate analyst collaboration across these boundaries. While still in its early stages, we describe work-in-progress towards achieving this goal, including motivation, functionality, concepts, and a high-level description of the proposed system architecture.

KEYWORDS: Cyber-security systems, collaborative software frameworks, collaborative security frameworks, computer security

1. INTRODUCTION

Although distributed geographically and often across different organizations, cyber-security analysts often face a similar set of tasks and, in many cases, analyze much of the same data. To this end, with some qualifications, it would be beneficial for cyber-security analysts to be able to share such information amongst each other. For example, suppose several software companies release bulletins documenting security vulnerabilities in each of their respective products, with some more severe than others. Cyber-security analysts across many different organizations, each charged with the similar tasks of defending their respective network infrastructures, will typically analyze a large common subset of these reports, prioritizing the more severe reports for action over those that are less pressing. Our studies [3][4][5] suggest that analysts across many different organizations scrutinize large numbers of similar reports on a daily basis, resulting in a significant amount of redundant analysis. A framework to support the communication of high-priority warnings amongst peer analysts would help to reduce the amount of redundant work, but currently no such collaborative framework exists. Secondly, the ability to communicate such high-priority security bulletins between peers efficiently has the potential to bring such warnings to analysts' attention in a more timely fashion than would otherwise be possible, potentially offering a higher probability of preempting future attacks, an effect from which all collaborating analysts would benefit collectively.

However, the design of such a meaningful and effective collaborative framework is not without its challenges. For one, the *types* of data in which analysts are interested will almost certainly vary in some regard between peers; an analyst charged with defending a network of Linux systems would almost certainly be interested in a different subset of data than that of an analyst defending a Windows network. On the other hand, peers of Linux administrators might expect to be interested in a largely common subset. Security is an inherent issue as well. Although peer analysts are collaborating in this sense, with the introduction of cross-domain data sharing, a collaborative

framework must provide a guarantee to prevent the flow of proprietary data between otherwise competing organizations. Another concern arises with respect to security policies even within individual organizations; a collaborative framework must be sure to preserve the security policies of individual organizations once the organizations are integrated into the framework. Concerns of this latter type have been investigated before [6][7], and these results will be helpful in addressing these types of issues in our final framework.

A third and significant challenge lies in automating these tasks such that the system operates as non-intrusively as possible so that the system provides meaningful feedback to analysts with minimal disruption.

We are currently developing a framework that allows for efficient and secure collaboration and communication of relevant information amongst cyber-security analysts across distributed locations and organizations, with the aim of non-intrusively facilitating their daily activities. Our work is still very much in its early stages, with requirements and, to some extent, the precise formulation of the problem to be addressed, still to be formalized. Nonetheless, we believe that we have developed the ideas and concepts behind our proposed collaborative security framework to a level of maturity to which they may be of interest to the COLSEC community. With this in mind, we focus primarily on the motivation and concepts behind our proposed framework in the hopes of stimulating a fruitful workshop discussion.

The remainder of the paper proceeds as follows: In Section 1.1 we discuss motivation for our proposed collaborative framework. In Section 2, we discuss related work. In Section 3, we discuss, at a high level, the various proposed components of our system. We briefly discuss security-related concerns in Section 4, and we conclude and discuss open problems in Section 5.

1.1 Motivation

There are a number of challenges that make it difficult for cyber-security analysts to collaborate effectively. Analysts are often reluctant to share unverified theories about evolving situations via official channels for a variety of reasons. For example, inaccuracy might reflect poorly on their organizations. Information posted to official channels might be later redacted or otherwise rendered out of date, and content from "trusted" commercial data sources is sometimes tainted by hype and influenced by competitive market advantage. In contrast, our studies [3][4] show that cyber-security analysts rely instead on informal sources of information such as blogs and other social media for up-to-date information in order to share and discuss unsubstantiated hypotheses. Hence, many important technical details can only be found in the contents of these social media, including a rich set of indicators of previously unknown vulnerabilities. But cyber-security professionals cannot effectively monitor more than a dozen or so unofficial information sources—typically checking these sources once or twice a day. Furthermore, it is not always clear to what extent these informal sources can be trusted, and this discourages collaboration. Finally, our studies suggest that contributing to these sources of information requires effort that many analysts are not willing to expend. To this end, one of our goals is to encourage such participation by reducing the amount of effort needed to do so.

The recent example of GhostNet [22] demonstrated that successful cyber defense can require collaboration across traditional boundaries within and between organizations and across national boundaries. The study, whose field work began in a Tibetan embassy, ultimately uncovered a network of compromises that spanned 1,295 machines in 103 countries (**Figure 1**), but collaboration across boundaries like these is difficult using today's tools and methods; without collaboration support, political and organizational obstacles make true cooperative cyber security across geographic and corporate boundaries extremely difficult.

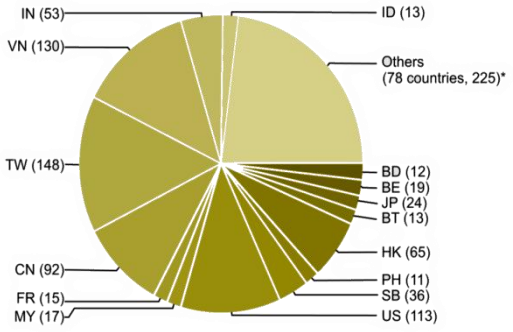

Figure 1. GhostNet: compromised machines by country (report at http://www.tracking-ghost.net)

2. BACKGROUND AND RELATED WORK

Collaborative computing frameworks have been well studied, dating back to as early as 1984 with the study of *Computer Supported Cooperative Work* [8], and since this time, the field has matured significantly. In [13], Kouzes et. al. give a survey of various collaborative frameworks

used by scientists in various fields including biology, astronomy, and environmental science, along with the challenges and other factors to be considered by designers of such frameworks. These *collaboratories* are all essentially scientific laboratory emulators, which allow scientists across various distributed locations to communicate using various software tools such as whiteboards, electronic notepads, chat tools, and videoconferencing tools to facilitate communication across various locations. Along similar lines, the authors in [19] discuss their efforts in creating *CCF*, a *Collaborative Computing Framework* consisting of a suite of tools, protocols, and software enabling scientific collaboration across a set of distributed locations, but with respect to a set of criteria differentiating it from previous work in this regard, including a specialized multicast data protocol, a completely distributed architecture (whose primary benefit is an improved performance), and the view of collaboration as specifically including computation and data manipulation with specific support for a high-performance, parallel, distributed environment.

When considering collaboration across such distributed environments, the question of security and access control immediately comes to mind, and indeed this area has been the topic of much work as well. In [6], Gong and Qian formalize a notion of a distributed system, viewed as a collection of users, machines, data objects, and others, with security policies specified using access control lists (ACLs); thus users are explicitly permitted or denied access to resources within a system, and a collection of collaborating systems is deemed *secure* if, taken together, the resulting unified security policy does not contradict that of any individual system. The problem considered by the authors in [6] is then to compute the *largest* possible collaboration between a collection of systems which still remains secure, and the authors show that this problem is in fact **NP**-complete, thus showing that it is impractical to compute this largest set for a large array of distributed systems. Shehab, Bertino, and Ghafoor [18] expand on a variant of this model; their work presents a framework similar to that modeled by Gong and Qian,[6][7], but in the absence of any trusted third-party mediator having a global view of access control policies and thus avoiding associated bottleneck issues associated with such a mediator.

There has been other work in the area of secure collaborative frameworks as well. For instance, in [24], Zhang, Nakae, Covington, and Sandhu present a security framework for collaborative computing systems, coupling an access control model based on the UCON model of Park and Sandhu with Sandhu's layered PEI [17] framework. The result is a usage control based security framework for collaborative applications which successfully models collaboration between resource users and resource providers within *virtual organizations* responsible for managing interaction between the two parties. Their framework bridges policy with implementation using the PEI model [17] of Sandhu, et. al., and the authors provide a prototype implementation in which access control policies are specified using an open access control language (XACML) and demonstrate their framework by providing implementations of various access control policies.

Our work differs from each of the above cited works. Principally, whereas previous collaborative frameworks [8][13][19] have been developed with very general-purpose usage in mind, our proposed framework is targeted specifically to the domain of cyber-security analytics, being developed with the specific intent of helping such analysts in various organizations to collaborate effectively and securely across a distributed environment. While other collaborative cyber security frameworks have been developed [11], these focus specifically on incident response. The work we describe here enables the communication of *potential* attacks and vulnerabilities (e.g. based on security vulnerability reports released by software vendors) with the intent of preempting such attacks. While both of these approaches could theoretically be addressed simultaneously in a single framework, they are fundamentally different problems, and in the interest of focusing our efforts on investigating a single problem, our framework is focused specifically on the latter.

Mendeley[1] is a social networking application for shared research. Mendeley's user-facing portion is comprised of a desktop application and a personal web account. Mendeley allows users to store, index, and annotate their library of PDF documents and paper references and share their library with the world. It imports much information directly from bibliographic software or PDF documents themselves, but to use it most effectively, users must enter a fair amount of information. For instance, to find documents similar to those in one's own collection, a user needs to enter keyword tags for each document. Even without these, the web part of Mendeley allows users to discover interesting statistics such as the most read articles and authors both overall and in a given discipline, most popular article tags in a discipline, and the distribution of disciplines of Mendeley users. One key difference between this approach and our framework is our intent to provide effortless collaboration by sharing information gathered from normal analytic activities and filtering them into the user's domain. If Mendeley were to do something

[1] http://www.mendeley.com/

similar to what we intend, the very act of reading a PDF would generate information about the user's interests and intent. This information would be used to suggest colleagues, suggest additional reading material, etc. Additionally, the act of writing a document might generate lists of potential citations from the context of where the cursor is located and the content of the document itself.

3. SYSTEM DESIGN

While collaboration solutions are effective in many domains, given the scope, problem size, and scalability of cyber problems, cyber-security analysts are often reluctant to adopt collaborative solutions [4][5]. We are developing a framework designed specifically for cyber-security analysts and their unique concerns. Our framework is being designed for use by domain experts in a fixed, non-mobile setting. There have been relatively few studies done on the activities, needs, and methods of cyber-security analysts. However, ethnographic studies have recently been published on the activities of system administrators [1][5], whose jobs often involve cyber security analytics. One major goal of our effort is to better understand the methods and work flows of cyber-security analysts. To this end we have conducted formal observations and working sessions with cyber-security analysts to understand how to better assist them in performing their duties. In this section, we present the outcomes of our user studies and provide a high-level description of the proposed system design and its functionality.

We interviewed eight cyber-security analysts at Pacific Northwest National Laboratory (PNNL) to discover more about the analysis process and how to better assist these analysts. From these studies, we learned that these analysts and their peers base much of their analytical activity on data which falls into two broad data categories. One of these categories is data from the broad category of *social media*, which includes data such as postings from security blogs, software vendor bulletins, and RSS feeds which aggregate data such as these. The other of these categories is data which can be roughly categorized as *internal log data*. This latter category includes log data from the analyst's own network such as web server logs, network traffic analysis, and server access logs. The implications of this distinction will become more apparent in Section 3, but from our analyst interviews, we learned that regardless of the data category, the analysts would benefit greatly from knowing what their peers were observing within their systems and from informal external sources. For example, if large numbers of an analyst's peers begin reading blog posts and security bulletins related to a new software vulnerability, it is likely that the analyst might be interested in reading more about this topic as well. Similarly, if one analyst detects a sudden pattern of network traffic which is characteristic of a known virus, it is likely that her peers may also be interested in this information, in case the peers may want to check for this virus on their own networks as well. However, our studies suggest that analysts are often reluctant to actively share their own queries and sources because of time constraints and privacy concerns. In addition, the cyber-security analysts we interviewed expressed concern over adding another analytic tool that will increase their workload. To this end, we are designing our system to be built on passive collection of analyst data, with support for analysts to choose the data to be collected. Another guiding principle is for the framework to behave as nonintrusively as possible, so as to minimize disruption of the analysts' daily routines.

3.1 Architecture

In this section, we discuss a high-level design of our proposed framework.

The proposed system architecture is designed with several key goals in mind. First, from the users' perspective, the system should be as nonintrusive as possible so as not to unnecessarily burden the analysts. Second, the system is not designed to replace the analyst. On the contrary, an integral part of the system architecture is a feedback loop through which analysts will contribute data for input into the system. It is through this feedback loop that analysts will ultimately share important findings with their peers. Third, the user must have control over what data is to be shared and the degree to which their identity is known to other users. We discuss this latter point more in Section 4. **Figure 2** provides a conceptual overview of the system towards which we are we are working:

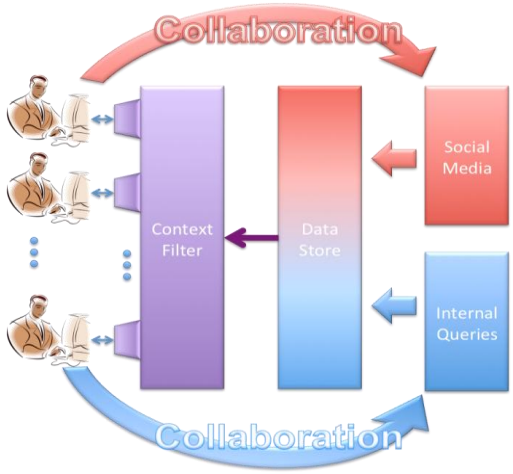

Figure 2 - System Overview

We briefly describe this figure first, and the remainder of this section describes each system component in more detail.

The analysts are represented in **Figure 2** as the set of users. As analysts perform their tasks, we envision the framework monitoring the analysts' activities, specifically the data being analyzed. As discussed earlier, our studies show that the data studied by analysts falls roughly into two categories— *social media data* and *internal log data*. In the case of *social media* data, the data equates roughly to a list of cyber-security-related URLs (RSS feeds, blog links, software vendor security bulletins, etc.) available to the user for daily analysis, and the activity monitored by the framework equates roughly to the specific links prioritized and selected ("clicked") by the user for further study. The motivation in this case is that the analysts collectively have access to a largely common set of such data for analysis, and if a large number of users select the same few topics for further investigation (e.g., if several analysts are suddenly reading social media posts related to a newly discovered security vulnerability in the latest Linux kernel), this is likely indicative that these select topics may be of higher priority, a fact which to which peer analysts should be alerted. The feedback loop in this case represents the specific links selected ("clicked") by individual analysts, and is represented by the top "Collaboration" arrow.

In the case of *internal log data*, the data equates roughly to the analysts' server log data, network traffic packet traces, server access log data, etc. In this case, however, we do not monitor the precise data being analyzed (e.g. sniffed packet contents, log file contents, access log contents) as one might initially expect. Instead, our framework will take a novel approach; instead of monitoring the log data itself, we will note the *types of queries* the users run over this data. For example, an analyst may query a trace of network packet data to search for patterns or signatures indicative of a particular virus attack. Our framework will not note the entire packet trace being analyzed by the user, but rather the *queries* the user is running over this data. This approach has a few benefits: For one, with potentially millions of packets traversing each analyst's network each day, and similarly with no guaranteed upper bound on server log sizes, the size of this data makes storing all such data analyzed by all analysts impractical. Secondly, security is a major concern in this regard— for obvious reasons, our framework cannot store raw packet data (which would include packet contents, header information, etc.) or server log data verbatim in a framework accessible by others outside the analysts' own organization. Storing *query* data, however, provides an abstraction of the issues about which analysts are actively concerned, without actually storing sensitive data. For example, suppose CERT [21] issues a warning about a new virus which steals user passwords and sends this data to a known central machine, `evil.org`. Analysts querying their network traffic would likely test for the presence of this virus on their system by searching for packets bound for `evil.org`, and our framework would make note of this query, without having to store any sensitive data. A large number of users querying for packets bound for this destination could be flagged as being indicative of a surge in the presence of this virus, a fact to which peer analysts should be alerted. This query summarization feedback into the system is represented by the bottom *collaboration* arrow in the diagram.

The rest of this section describes each of the components in **Figure 2** in more detail.

3.1.1 Data, Transformations, and Summarization

As we have discussed above, our framework will initially support two types of data— *social media* and *internal log* data. In both cases, this data is summarized and transformed, with the results placed in a data store for later delivery to the users. The summarizations and transformations differ between the two classes of data, and we discuss these next.

3.1.1.1 Social Media

The social media data consists primarily of links to web sites, RSS feeds, blog postings, etc. of interest. However, a major goal of our framework is to alert peer analysts to major trending *topics* of interest, such as an increase in the frequency of analysts reading articles related to the occurrence of a particular virus or software vulnerability. Thus, in order to perform any meaningful analysis over the topical content of this data, the links must be followed, and the link *content* retrieved and analyzed. In order to effectively deliver meaningful topical summaries back to the users the system must understand the content of the documents gathered. To enable this, our framework will employ natural language processing (NLP) components to analyze the content of the harvested documents. We envision an NLP pipeline consisting of the following components to achieve this goal, using open-source components wherever possible.

A Topic Detection [1] component will label an entire document or portions of a document with one or more topics. Topic labels can be defined by users training statistical models by supplying sample documents, or chosen by the system based on a discourse analysis of the importance of terms or phrases in the document. Topics can be used as a course-grained basis for describing the content of new documents and the interests of users.

An Entity Recognition [9][10][12][23] component will tag entities mentioned in a document and categorize them by

type. Entity recognition and categorization will be driven by statistical models defining common entity types (PERSON, ORGANIZATION, etc.) and domain-specific gazetteers listing names of entities of types of special interest (APPLICATION, SERVICE, etc.). These entities can then be used to describe document content such as topic labels, which can then be used for semantic queries.

A Sentiment Analysis [2][15][16] component will tag entire documents, portions of documents, or entities mentioned with an expressed opinion or mood. Lexicons are used to identify the polarity, subjectivity or strength of an expressed opinion, while grammatical dependencies are used to identify the target entity. When there are many documents of a particular topic or containing a particular entity, sentiment analysis can help narrow to the most severe or urgent issues.

Finally, an Interesting Phrase Recognition component will tag phrases in a document are statistically unexpected in a context, for example, within a certain topic or at a certain point in a timeline. This can be used to find the most potentially interesting occurrence of a topic or entity mentions.

Once the system understands the content of documents, techniques such as Relevance Feedback or Personalized Search can be used to build a profile of the user from the content, guiding analysts to topics of heightened interest.

This process of taking links clicked by the users, fetching the corresponding link content, passing this content through the NLP pipeline, and placing the resulting summarized data in the data store, is represented in **Figure 2** as the box labeled "Social Media", along with the arrow leading from that box into the box labeled "Data Store".

3.1.1.2 Internal Log Data

The internal log data consists primarily of network-related data internal to an analyst's network— for example, network packet traces, web server logs, proxy access logs, and so on. Based on our user studies, we expect that analysis over this data will be performed as queries over this data (e.g., using a relational database), and, for reasons discussed earlier in this section, our framework will store not the data itself, but rather abstractions of these queries.

However, this poses certain logistical challenges. Chief amongst these is the need to provide a query abstraction layer capable of representing the types of queries which might be asked by the analyst over the various types of internal log data.

At one level, the abstraction layer will need to unify the various tools used for similar tasks in different organizations. For instance, one organization may use Apache as its web server, while another uses Microsoft's IIS, while a third may use `lighttpd` for the same purpose. In all cases, however, analysts will look at their respective server logs to note any suspicious activity, and moreover, the types of data available in these logs is largely common between all three cases— in all cases, for instance, a system administrator can determine the IP address of a visitor, the time of the access, any user credentials provided, etc., although the format of this data will clearly vary between the different implementations. One logistical issue we will need to address is to develop an abstraction of the data fields likely to be queried by the analysts, independent of the tool being used. Once this has been formalized, the users' queries over their data can be expressed using this abstraction, and these abstracted queries can then be stored in our data store.

At a higher level, the abstraction layer will need to be expressive enough to express queries over all different *types* of log data. In other words, the layer must be able to encapsulate queries over not only log data, but network packet data and other server log data as well.

Our framework will then use this "common query protocol" to encapsulate the analysts' queries for feedback into the system, and this process is represented in **Figure 2** as the box labeled "Queries", along with the arrow leading from that box into the box labeled "Data Store".

We envision that user queries will be collected by a thin proxy process, residing between the user and the system being queried (e.g., perhaps a relational database), monitoring the queries executed by the user and passing copies of these queries to the transformation and summarization module.

The existence of a collaborative framework will likely incur additional overhead, and we anticipate that the majority of this overhead will result from these data summarization and transformation modules; the feedback path (indicated by the "Collaboration" arrows in **Figure 2**) incurs minimal overhead, as this can be implemented using a simple message-passing mechanism, and the data store and context filter can be implemented efficiently using a properly designed relational database, for example. The summarization and transformation, however, require comparatively expensive operations— for example, fetching documents from the World-Wide Web in the case of the social media data, and we must take special care to implement these pieces efficiently.

3.1.2 Data Store

Once the analysts' data has been transformed and summarized, it will be placed in a relational database where it will be available for feedback to the analysts. This database is represented in **Figure 2** by the box labeled "Data Store", and this database will be queried by the Context Filter (Section 3.1.3) to deliver specialized content to individual analysts. Once data has been entered into the data store, it will have been normalized as to be queryable by a common query language (e.g., SQL). The breadth of these normalization rules makes being comprehensive nearly impossible. A rule sharing component would be one possible method of addressing this issue.

3.1.3 Context Filter

An analyst profile drives the gathering, analysis and contextualization of the summarized data. These profiles, tailored specifically to each analyst, represent the system's understanding of each analyst's environment (e.g., the systems and infrastructure for which the analyst is responsible, the analyst's *social media* data sources, etc.). While the data store contains all of the summarized data from all analysts, most analysts will only be interested in a small subset of this data. For example, a Linux administrator will likely not be as interested in Microsoft Windows security bulletins, and vice-versa. The purpose of the context filter is to selectively filter relevant data from the data store, custom tailored to the needs of each analyst.

We envision the users' profiles to be dynamic, adjusting to reflect changes in the analyst's environment and data-needs based on the analyst's interactions with the system, although the details, such as the precise inputs, user interface, and the model by which the profiles adapt, remain yet to be determined.

The context filter will be based on three primary sources of information. The first is a model of the analyst's system, specified by the analyst; for example, the user might specify that, as a Linux system administrator, Linux-related topics take precedence over all others.

The second will be the users' query and browsing history. We envision the framework to adaptively learn topics of interest to the analyst. For example, if the user begins to browse links related to Linux kernel security issues with higher frequency, we expect for our framework to interpret this as a "new interest", and consequently for the context filter to begin to suggest such topics to this analyst with a higher priority. Problems such as these have been well studied in the field of Machine Learning [14], and as such, we expect to draw heavily from past work in this area to help to implement this functionality.

The third factor for contextualization will be based on the information consulted by users with similar interests; if one Linux analyst is concerned with topics related to a new kernel vulnerability, chances are likely that other Linux analysts will be interested in the same topic as well. There are several open problems with respect to this problem. For one, what is the optimal method of quantifying similarity of interest between analysts? Clearly not all Linux analysts will be interested in exactly the same topic set. A major challenge to implementing a meaningful contextualization lies in the answer to this last question.

3.1.4 User Interface

As we are still in the process of designing our framework, the user interface itself remains largely undecided. However, from our analyst interviews, we do know that the interface must be designed around several guiding principles. Our system design is based on two competing constraints: The need for cyber analysts to collaborate on one hand, but with the constraints preventing them from doing so on the other. The cyber analysts we interviewed have cited privacy and security as a major road blocks to collaboration, but additionally the analysts do not want another analytic tool that will increase their workload. Keeping these issues in mind, the system must facilitate collaboration as nonintrusively as possible, yet with users having complete control over what information is shared. Thus, we face the usual tradeoff between convenience and security; one challenge we currently face is the question of how best to automate the process of selecting analysts' data for use in our system as nonintrusively as possible (convenience), while at the same time satisfying the analysts concerns that sensitive, proprietary data remain out of our system (security).

4. PRIVACY AND SECURITY

As with any collaborative framework, security is a major issue. There has been much work in the past with respect to access control and information flow in a collaborative network (for instance, the need for the security policies of individual networks to remain intact even under composition [6][7]), and we discussed these in Section 2. These concerns will certainly arise in our framework as well, but, as this particular problem is relatively well-studied, this prior work will serve to guide us in this regard.

There are other security-related issues as well. For instance, we must be careful to guarantee that proprietary

information will not escape organizational boundaries. In the case of summarized social media data, the feedback data (marked "Collaboration" in **Figure 2**) is derived entirely from publically accessible data, so this concern is largely not an issue. In the case of summarized query data, however, the distinction is not as clear-cut. For instance, if the *common query protocol* of Section 3.1.1.2 is not properly defined, the potential exists for proprietary information about an analyst's network to be leaked into our framework.

Another issue arises with respect to revealing the identity of individual analysts; our studies suggest that some degree of anonymity is a major concern for the users. To this end, we envision our framework supporting an identity spectrum; at one end, the analyst may elect to share his full identity, or at the other extreme, choose to remain anonymous. In between, the analyst may choose to be known by a pseudonym, or only by the organization to which she belongs.

Another concern lies in the ability of the analyst to select the peers with whom her data can be shared. For example, corporate policy may expressly forbid any information sharing of any sort with a specific competitor, yet this should not preclude the ability of the analyst to collaborate with other analysts with no relation to the competitor. A whitelist or blacklist would be one possible solution to this issue.

Other more subtle security issues which will need to be addressed lie in the possibilities of users attempting to use the system for purposes other than those intended— for instance, social engineering or industrial spying. We believe the key to addressing such concerns, with the latter in particular, lies in making careful decisions restricting the type of data that can be gleaned from the system.

Trust, in this framework, is a bidirectional issue; new analysts joining a system for the first time must be assured that the system monitoring and analyzing their queries is not doing so maliciously, and conversely, the system must ensure that new analysts joining it are not doing so in an attempt to glean sensitive information. To this end, we envision a bidirectional web-of-trust. Collaborating organizations would maintain their own known servers to which new analysts would connect, ensuring new analysts of the integrity of the system to which they are connecting. Similarly, collaborating organizations, by maintaining credentials of peer servers, form a web of trust whereby analysts connecting to peer servers are implicitly trusted.

5. CONCLUSION AND OPEN PROBLEMS

In this paper, we have described the motivation behind a proposed collaborative framework, currently in development, to support and facilitate the daily activities of cyber-security analysts distributed across geographical and organizational boundaries. Through interviews of such analysts, we have determined that although communication and collaboration between such analysts would be mutually beneficial to the analysts involved, several barriers to practical and efficient collaboration exist, and as such, no such framework currently exists to support such efforts. We have described our progress towards developing such a framework, which is still very much in its early stages. We described a high-level view of the architecture for our proposed framework, consisting of a feedback loop between the analysts and the system, with the users selecting data of higher priority, and the system observing the global selections of all of its users, using these observations to make tailored recommendations to its users on the next iteration.

5.1 Open Problems

While we have laid out the foundations for developing a collaborative framework for cyber-security analytics, there is still much left to be done, and in addition to development of a prototype, several questions of interest remain open. In Section 1.1, we outline ways in which our framework will be used to observe analysts' activities in order to better assist the analysts in performing their duties. Logistically, however, this opens an entirely new set of questions. For instance, in sharing information between analysts across different organizations, the question of security and information flow becomes immediately relevant; to give one obvious example, our framework must not allow proprietary information to escape from one organization to another. To a certain extent, the general problem of security in collaborative frameworks has been studied in depth [6][7][20], but in general, these studies deal primarily with the concern of violating individual entities' security policies once these entities are joined by a collaboration framework, whereas one of our major challenges involves the prevention of, for instance, proprietary information flow between collaborating entities.

As discussed in Section 3.1.1.2, the *common query protocol* we describe in that section must be carefully designed so as to guarantee that no proprietary or otherwise sensitive data is leaked into our framework.

At a lower level, although we distinguish between *social media* and *internal log* data, it would be desirable to be able to unify these two types of data to allow for the handling and analysis of all data in a uniform way.

However, it is not immediately clear how or if such a unification is possible.

The context filter described in Section 3.1.3 relies on the ability to meaningfully quantify the similarity in interests between two analysts, a largely subjective concept. At a higher level, the general question exists of how to make our interface as nonintrusive as possible yet still remain effective in aiding collaboration. While these open questions are difficult, we believe that a modified work environment where speculative collaboration is encouraged and fostered will increase the participation and decrease the risks of collaboration. Our hope is that our framework will be a stepping-stone into a truly collaborative cyber security environment.

REFERENCES

[1] Allan, J. 2002 *Topic Detection and Tracking: Event-Based Information Organization*. Kluwer Academic Publishers.

[2] Choi, Y., Kim, Y., and Myaeng, S. 2009. Domain-specific sentiment analysis using contextual feature generation. In *Proceeding of the 1st international CIKM Workshop on Topic-Sentiment Analysis For Mass Opinion* (Hong Kong, China, November 06 - 06, 2009). TSA '09. ACM, New York, NY, 37-44.

[3] Fink, G. A., D. McKinnon, S. Clements, and D.Frincke. "Tensions in collaborative cyber security and how they affect incident detection and response" chapter in Collaborative Computer Security and Trust Management. IGI Global, to appear.

[4] Fink GA, North CL, Endert A, and Rose SJ, "Visualizing Cyber Security: Usable Workspaces." In Proceedings of the 2009 Workshop on Visualization for Computer Security (VizSEC 2009). PNNL-SA-66416

[5] Fink, G., R. Correa, and C. North. "System Administrators and their Security Awareness Tools." 2005. Available at http://people.cs.vt.edu/~finkga/Research%20Defense/System%20Admins.html.

[6] Gong, L. and Qian, X. 1994. The Complexity and Composability of Secure Interoperation. In *Proceedings of the 1994 IEEE Symposium on Security and Privacy* (May 16 - 18, 1994). SP. *IEEE Computer Society*, Washington, DC, 190.

[7] Li Gong, Xiaolei Qian, "Computational Issues in Secure Interoperation," *IEEE Transactions on Software Engineering,* pp. 43-52, January, 1996

[8] Grudin, J. Computer-Supported Cooperative Work: History and Focus. *IEEE Computer,* vol.27, no. 5, pp.19-26,

[9] Guo, H. L., Zhang, L., and Su, Z. 2006. Empirical study on the performance stability of named entity recognition model across domains. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (Sydney, Australia, July 22 - 23, 2006). ACL Workshops. Association for Computational Linguistics, Morristown, NJ, 509-516.

[10] Guo, H., Zhu, H., Guo, Z., Zhang, X., Wu, X., and Su, Z. 2009. Domain adaptation with latent semantic association for named entity recognition. In *Proceedings of Human Language Technologies: the 2009 Annual Conference of the North American Chapter of the Association For Computational Linguistics* (Boulder, Colorado, May 31 - June 05, 2009). Human Language Technology Conference. Association for Computational Linguistics, Morristown, NJ, 281-289.

[11] Khurana, H., Basney, J., Bakht, M., Freemon, M., Welch, V., and Butler, R. 2009. Palantir: a framework for collaborative incident response and investigation. In *Proceedings of the 8th Symposium on Identity and Trust on the internet* (Gaithersburg, Maryland, April 14 - 16, 2009). K. Seamons, N. McBurnett, and T. Polk, Eds. IDtrust '09, vol. 373. ACM, New York, NY, 38-51.

[12] Kozareva, Z., Ferrández, O., Montoyo, A., Muñoz, R., Suárez, A., and Gómez, J. 2007. Combining data-driven systems for improving Named Entity Recognition. *Data Knowl. Eng.* 61, 3 (Jun. 2007), 449-466.

[13] Kouzes, R.T.; Myers, J.D.; Wulf, W.A., "Collaboratories: doing science on the Internet," *IEEE Computer* , vol.29, no.8, pp.40-46, Aug 1996

[14] Mitchell, T. *Machine Learning*. McGraw-Hill. 1997

[15] O'Hare, N., Davy, M., Bermingham, A., Ferguson, P., Sheridan, P., Gurrin, C., and Smeaton, A. F. 2009. Topic-dependent sentiment analysis of financial blogs. In *Proceeding of the 1st international CIKM Workshop on Topic-Sentiment Analysis For Mass Opinion* (Hong Kong, China, November 06 - 06, 2009). TSA '09. ACM, New York, NY, 9-16.

[16] Pang, B. and Lee, L. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.* 2, 1-2 (Jan. 2008), 1-135.

[17] Sandhu, R., Ranganathan, K., and Zhang, X. Secure Information Sharing Enabled by Trusted Computing and PEI Models. In Proceedings of the ACM Symposium on Information, Computer, and Communication Security.

[18] Shehab, M., Bertino, E., and Ghafoor, A. 2005. Secure collaboration in mediator-free environments. In *Proceedings of the 12th ACM Conference on Computer and Communications Security* (Alexandria, VA, USA, November 07 - 11, 2005). CCS '05. ACM, New York, NY, 58-67.

[19] Sunderam, V., Cheung, S. Y., Hirsch, M., Chodrow, S., Grigni, M., Krantz, A., Rhee, I., Gray, P., Olesen, S., Hutto, P., and Sult, J. 1998. CCF: Collaborative Computing Frameworks. In *Proceedings of the 1998 ACM/IEEE Conference on Supercomputing (Cdrom)* (San Jose, CA, November 07 - 13, 1998). Conference on High Performance Networking and Computing. IEEE Computer Society, Washington, DC, 1-6.

[20] Tolone, W., Ahn, G., Pai, T., and Hong, S. 2005. Access control in collaborative systems. *ACM Comput. Surv.* 37, 1 (Mar. 2005), 29-41.

[21] United States Computer Emergency Readiness Team (US-CERT) website. http://www.us-cert.gov/current/

[22] Walton, G., et al., *Tracking GhostNet: Investigating a Cyber Espionage Network*. 2009, Information Warfare Monitor. http://www.tracking-ghost.net

[23] Whitelaw, C., Kehlenbeck, A., Petrovic, N., and Ungar, L. 2008. Web-scale named entity recognition. In *Proceeding of the 17th ACM Conference on information and Knowledge Management* (Napa Valley, California, USA, October 26 - 30, 2008). CIKM '08. ACM, New York, NY, 123-132.

[24] Zhang, X., Nakae, M., Covington, M., and Sandhu, R. 2008. Toward a Usage-Based Security Framework for Collaborative Computing Systems. *ACM Trans. Inf. Syst. Secur.* 11, 1 (Feb. 2008), 1-36.