TOWARDS EFFECTIVE COLLABORATIVE ANALYSIS FOR DISTRIBUTED INTRUSION DETECTION

Xianlin Hu
University of North Carolina at Charlotte
Charlotte, North Carolina, U.S.
email: xhu8@uncc.edu

Guanghua Song
University of the Pacific
Stockton, California, U.S.
email: shg6699@gmail.com

Lane Harrison
University of North Carolina at Charlotte
Charlotte, North Carolina, U.S.
email: lthtech@gmail.com

Aidong Lu
University of North Carolina at Charlotte
Charlotte, North Carolina, U.S.
email: alu1@uncc.edu

Jinzhu Gao
University of the Pacific
Stockton, California, U.S.
email: jinzhugao@gmail.com

Weichao Wang
University of North Carolina at Charlotte
Charlotte, North Carolina, U.S.
email: weichaowang@uncc.edu

ABSTRACT

This paper addresses the problem of collaborative analysis in a distributed setting via a network security application. Network security analysis often requires accurate and timely results, which is very challenging to achieve in large dynamic networks with a single user. To address this issue, we design and develop a collaborative detection mechanism for complex intrusion detection applications. We also establish a set of collaboration guidelines for team coordination with distributed visualization tools. These collaboration guidelines cover the designs of coordination roles, workflow, collaborative environments and human computer interactions. We apply them to generate a prototype system with interactions that facilitates collaborative visual analysis. According to the expert feedback acquired for assessing our approach, we propose directions for improving the efficiency of collaborative analysis.

KEY WORDS

Distributed collaborative analysis, computer supported cooperative work, human-work interaction design, guidelines and design heuristics

1 Introduction

Collaborative analysis can benefit many large scale applications where a small group of users discuss and negotiate their interpretations of the data with which they are working. These techniques are often required for application fields where task complexities can easily overrun computing power and algorithm intelligence [12, 16, 14]. Especially in security applications, collaboration mechanisms are crucial to provide time efficient solutions for processing a large amount of data in real time. While approaches have been explored for assisting individual analyst with detection tasks, the problem of collaborative analysis for such applications is still open.

In this paper, we concentrate on exploring a suitable solution for complex intrusion detection applications. Our motivation comes from the fact that important networking environments are always protected by security teams. Traditionally, such teams would use an algorithm-based Intrusion Detection System (IDS) to warn them of threats, but IDSes are prone to give false alarms. Since every alarm must be verified, security teams end up wasting much time investigating events that turn out to be false alarms. Collaborative analysis can provide a practical solution to overcome the ineffectiveness of automatic detection algorithms, limits of computing resources, and complexity of advanced malicious attacks. Especially for intrusion detection, where new or unknown attacks are often introduced, having a group of experts analyzing data in real time is crucial to provide accurate and time critical results.

Since most previous detection approaches have been designed from a single-user perspective, it is usually not possible to apply them directly to team coordination. Relevant studies, such as ones exploring teamwork theories, have been extensively performed in the fields of artificial intelligence and robotics. However, the proactive collaborative problem-solving feature of the security domain differentiates it from multi-agent coordination in their applications. To build an effective collaborative visualization model, various aspects related to the collaborative problem solving process, such as knowledge sharing and social factors [20], should be considered. However, it is not yet understood how interfaces and interaction techniques should be designed to specifically address the needs of distributed collaborative analysis.

In this paper, we design and develop a collaborative detection mechanism for defending against complex malicious attacks in wireless networks. The goal of our collaboration teams is to identify any hidden attacks and remove malicious nodes from the network. Here we concentrate on defending against a particularly harmful attack known as the Sybil attack, which has numerous variations. We first analyze the challenges and requirements for designing such coordinated systems. Later, we describe a web-based prototype system, which is built based on our design principles and heuristics. Our system supports multi-user input, shared and individual views on detection findings, and flexible workspace organization to facilitate group analysis.

The main contribution is that our work explores the area of collaborative analysis in a distributed setting, which to our knowledge has not been explored in significant depth. Our approach incorporates results from several research fields, including models of human behavior, teamwork theory, and interface design. We also provide a detailed discussion of different collaboration aspects. We give practical solutions for security applications in real-life for defending against various attacks, assuming that a reasonable detection algorithm is provided for each representative attack. The web-based prototype system and networking data collections provide a testbed for other researchers to explore and evaluate the effectiveness of different coordination aspects, which are hard to access without a working example.

2 Related Work

2.1 Distributed Collaborative Visualization

The concept of distributed visualization was first introduced by Anupam et al. [1] in 1994. The collaboration in distributed visualization environment is often based on systems [3] which can be deployed on different computers. On the human level, collaborative visualization is defined as multiple users working together using visualization systems to achieve the same goal [3]. Distributed collaborative visualization combines these two concepts both at the system level and human level. Many distributed collaborative visualization applications are web-based [15, 26, 24].

For security analysis, collaboration between multiple work groups is often required in time-critical situations. Thus it is necessary to improve the incident response process. When dealing with security events in large amounts of data, collaborative analysis can help analysts find meaningful information [25], which could then help administrators formulate a quick response [8]. However, based on our knowledge, we have found no effective collaborative solutions for security analysis.

2.2 The Mechanics and Social Aspects of Collaboration

Collaborative teamwork includes two important components: the mechanics and social aspects of collaboration. The mechanics of collaboration include common actions which team members must take to complete a shared task in the collaboration process. For example, Gutwin and Greenberg [13] identified several major actions including communication, coordination, planning, monitoring, assistance, and protection.

Ma and Wang [20] have pointed out that knowledge sharing and the social aspects of collaboration should be considered; particularly to better support collaborative work for large scientific projects using visualization. Social aspects are inevitable in collaborative work. The study of social aspects often involves exploring the structure of participant roles, awareness, and trust. Furthermore, social aspects include both social and cognitive presences. “Social presence reflects the ability to connect with members of a community of learners on a personal level. Cognitive presence is the process of constructing meaning through collaborative inquiry” [11]. Social and cognitive presences are also needed for online collaboration [22].

3 Design Guidelines and Heuristics

This section describes several guidelines for assisting a small team (3-15 members) on monitoring and defending a network collaboratively with a distributed environment. Our design references knowledge from multiple fields, including social science, psychology collaboration models and teamwork theory; and our observations of team dynamics in security applications.

3.1 Designing the Roles

The structure of participant roles can directly affect the efficiency of a collaborative problem-solving team. In real applications, the privilege of making final decisions can only be given to a small number of participants; thus we need to separate the roles of participants into at least two groups: *administrators* who supervise the collaboration process and *analysts* who handle individual detection tasks.

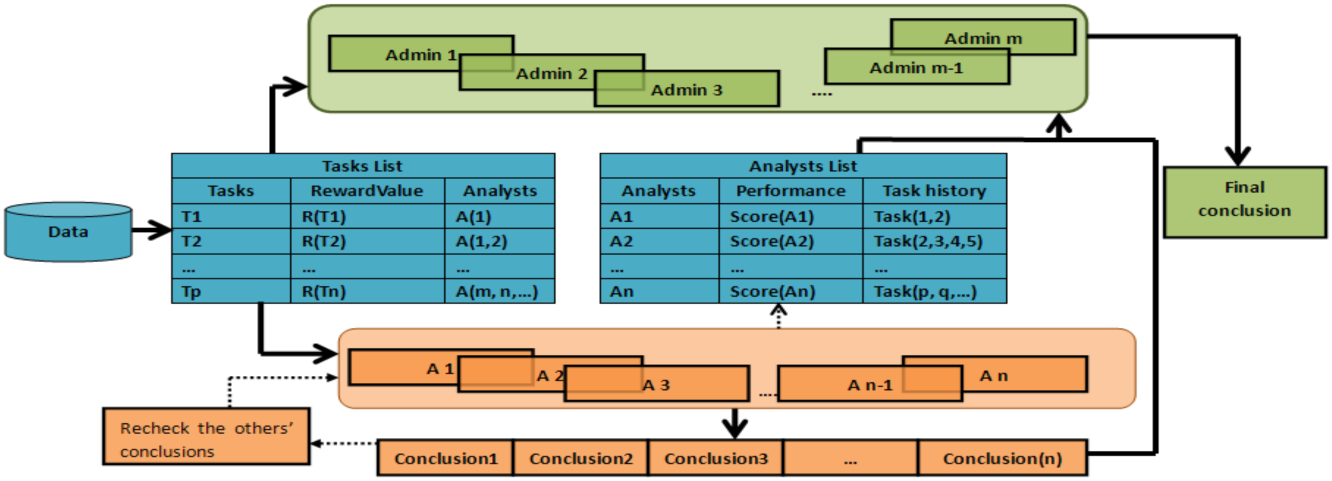
In our design, analysts are treated as the main players in the collaborative analysis process. Analysts can actively choose their tasks and coordinate with each other to complete the detection process. On the other hand, administrators have the responsibilities of monitoring team progress such as reviewing the results from analysts, adjusting task rewards, adding new tasks to the task list, monitoring analyst performances, drawing final conclusions, and removing malicious nodes from the network.

3.2 Designing the Workflow

Workflow is a powerful tool to guide collaboration and monitor overall team performance [9]. Our efforts are focused on designing a mechanism to smooth the coordination and communication among analysts.

As Figure 1 shows, the workflow starts from a server, which collects and stores data from the network in real time. For generality, we only collect network topology, which is one of the most commonly used information sources in network security. Once the detection process starts, the server automatically generates a list of tasks by dividing data into equal time durations. We define the set of tasks as $T := T_1, ..., T_p$, where p is the number of tasks defined in the system. Each task has an associated estimate cost of time and a reward value $R(T_i)$, which changes with time and detection results to promote early selection of important tasks.

In the workflow, administrators can access all the data and findings from individual analysts. The administrators



Tasks List		
Tasks	RewardValue	Analysts
T1	R(T1)	A(1)
T2	R(T2)	A(1,2)
...	...	...
Tp	R(Tn)	A(m, n,...)

Analysts List		
Analysts	Performance	Task history
A1	Score(A1)	Task(1,2)
A2	Score(A2)	Task(2,3,4,5)
...	...	...
An	Score(An)	Task(p, q,...)

Figure 1. The workflow of coordinated detection. A list of analysis tasks is generated in real-time at the server and can be accessed by both administrators and analysts. Analysts handle the detailed detection process, while administrators overview the team performance and make final decisions on action. The workflow provides administrators with the flexibility to monitor and adjust the team progress and analysts capabilities of collaborative analysis.

are the decision makers who review findings from analysts and draw final conclusions by considering the whole detection process. They are also responsible for monitoring overall team performance and improving the team efficiency by actively assigning tasks to analysts, modifying reward values, and suggesting the involvement of additional analysts.

Analysts must detect and explore hidden attacks by studying various matrix patterns generated from different time periods and investigate suspicious activity. Analysts first either select a task from the generated task list by considering its reward value or find that they have been assigned a set of tasks by an administrator. At this point, the analysts explore this assigned task via the provided visualization and interaction tools. During this analytical process, sharing information is needed. Analysts can share their conclusions and suggestions through images or text stored in a repository on the server. Thus, other analysts can reference historical findings for the task in order to make a more comprehensive decision regarding suspicious activity. Finally, the server automatically updates the task list and reward values based on the analysts' conclusions.

3.3 Designing Collaborative Detection

The following describes the interaction and collaboration regarding three aspects: detection, coordination, and communication.

3.3.1 Detection

Our design of collaborative analysis is applicable to general intrusion detection tasks. In this paper, we concentrate on detecting Sybil attacks, in which a malicious node either steals or generates false identities. The detection of such attacks often requires iterative exploration. In large networks, there are normally several interrelated attacks, which require collaborative analysis. In this section, we introduce the Sybil attack detection briefly. For more details, please refer to the appendix of Sybil attacks and the strategies of the three detection algorithms.

Due to the complexity of Sybil attacks, currently there are no algorithms that can detect them automatically. Instead, interactive visualization tools have been explored to suggest the existence of malicious nodes [19]. We use matrix visualization to represent network topology. As demonstrated in Figure 2, a matrix visualization of normal network topology generally has an appearance of random patterns like the left image. While the middle and right patterns, generated by reordering the node sequences, are indications of potential Sybil attacks. The white nodes located on the left bottom corners are suspicious in such patterns.

In this system, we adopt three independent detection algorithms which exploit different aspects of Sybil attack features to reorder the node sequences. Each algorithm generates an image like the examples in Figure 2. As long as one of the images demonstrates the suspicious patterns, users can identify Sybil attacks in a respective time period. Furthermore, what makes the detection task complicated is that Sybil attacks do not demonstrate anomalies in simple neighbor relationships at individual time steps. Analysts need to test different combinations of time durations and detection methods to explore potential attacks. The setup of this detection strategy allows us to introduce additional detection methods and detect other attacks in the future.

As the users of our system are security experts, we also provide a knowledge-based reordering interaction method, which allows users to input their assumptions or conclusions by specifying benign nodes in blue and ma-

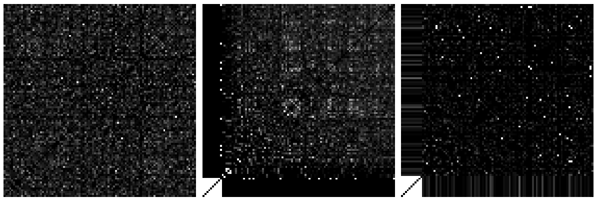


Figure 2. Left: General statistical topology matrix does not reveal any suspicious patterns; Middle and Right: A suitably reordered topology matrix with a certain time range can reveal traces to identify malicious nodes.

licious nodes in red. With the provided information, the algorithm reorders topology patterns automatically. This can often better reveal suspicious patterns and help users to gradually locate all suspicious nodes.

3.3.2 Coordination

As described in [17], "coordination is the attempt by multiple entities to act in concert in order to achieve a common goal by carrying out a script/plan they all understand." Thus building up such a good "script" for team members is the key point for the whole collaboration process. For distributed collaborative environment, Neale et al. [21] have pointed out that coordination should be defined by the combination of procedures, tasks, tools, and communication. In this section, we discuss the procedures of coordination including the division and allocation of task, the updating of reward values and performance scores, and decision making. In section 3.3.3, we discuss communication.

Collaboration style: We define two stages in the detection process: detection and monitor stages. The detection stage is the duration from the announcement of the first suspicious node to the time step in which all known attackers have been removed. The monitoring stage is time during which there are no obvious attacks occuring. In our distributed security environment, collaborators switch their collaboration styles between types of loose style in monitor stages and close style in detection stages. Collaboration style is also related to the division of tasks.

Task reward: In order support efficient collaboration among the analysts, providing a guide for them to target at time-critical tasks is necessary. Therefore, we design a reward value metric that is associated with each task. All the tasks are initially assigned reward values of 0s. The reward values are updated according to analysts' findings of suspicious nodes. Its value increases by 1 when any new or additional suspicious nodes are found, or conflicting conclusions for the same tasks are drawn.

Therefore, task reward value is the sum of the number of suspicious nodes and the times of all the conflict results. If the reward value is high, it indicates the task is in high risk and needs to remove the malicious nodes as soon as possible, or the task is too complex to make a clear conclusion by only two or three analysts, it needs more analysts to double check. Administrators can use reward value to control the task allocation. They can assign the task with high value to more analysts as well.

Division of task: Division of tasks is one of the most basic aspects in collaboration work. However, it is not trivial to parallelize an entire workflow into proper independent units [14]. The data we use in our system is temporal network data, so we divide tasks based on time duration.

The choice of task duration $d(t)$ at the time step t can be adjusted through two factors: the response duration of analysts and their collaboration styles. We require that each task be engaged by several analysts simultaneously. Thus response duration is the time costs between the selection and conclusion of a task from all the analysts for all the tasks during a recent history. We calculate the average $ar(t)$ of these response durations. Generally, a Sybil attack can cause severe damage within a certain time TD_{max}. The maximum change is defined as D_{max}. Thus, we design an equation 1 for division of tasks. The second factor is collaboration style, meaning that a longer duration for monitoring stages and a shorter duration for detection stages when communication is more frequently needed. We use half d for the detection stage.

$$d(t) = \begin{bmatrix} -D_{max} \times (\frac{2ar(t)}{TD_{max}} - 1)^3, if\ TD_{max} >= ar(t) >= 0 \\ -D_{max}, if\ ar(t) > TD_{max} \end{bmatrix} \tag{1}$$

Allocation of task: Effective division of task is not sufficient for successful collaboration, the efficient allocation of tasks is also necessary. In order to allocate tasks to proper individuals, analysts in our design can choose new tasks on their own with the information of reward values and their own task history. Allowing analysts to manage their work independently can bring benefits, in contrast to assigning their task passively [7]. The latter approach requires a central planner to control each analyst's workload, and thus the central planner must know much precise information about the whole network state and the analysts' respective productivity capacities. In such a management structure, even small mistakes made by the planner can drastically affect the entire organization. Allowing analysts to actively select tasks avoids this problem. Additionally, every analyst can concentrate on his or her own tasks without concerning about what tasks the other analysts are working on. Furthermore, administrators in this model are able to adjust and take action on the incoming results from the analysts. This model may slightly reduce the output of the analysts, since they must take time to select their own tasks. However, we believe that the time gained by administrators and the benefits brought by analysts having a say in their task allocation outweighs this small time loss for the analysts.

Task coordination: An important aspect of collaboration among analysts is referring to the regions deemed suspicious by the other group members through a spatial context [14]. Clark [4] grouped many forms of spatial reference into two categories: pointing and placing. Pointing means that using some vectorial reference to direct attention to specified regions or objects. Placing means that moving some information into one shared-space. In our system, we provide a task list and the corresponding suspicious node list to promote coordination among all participants. Analysts can point out their discoveries of suspicious nodes or regions in generated images after detection. Afterwards, they can upload their conclusions to the server, and others can verify these conclusions. Analysts test conclusions from others by reordering topology patterns according to their assumption of suspicious nodes.

Performance measurement: As described by Shipman and Wholey [23], "Performance measurement is the ongoing monitoring and reporting of program accomplishments, particularly progress towards pre-established goals". In our design, we build quantitative performance standards to improve the accountability of each analyst and the general effectiveness of coordination. To do this, we collect correctness and performance scores for each analyst. Correctness can be measured by comparing the analysts' conclusions for each task with the administrators'. The percentage of correctness increases when the conclusions are identical, and otherwise decreases. Thus the suspiciousness degree of any given task can be modified by administrators according to the correctness scores of the analysts who processed the task's data. Likewise, we measure the performance score by accumulating all the final reward values of the tasks an analyst has processed. We can assess productivity of each analyst by this score. The performance list is only available to administrators.

Decision Making: Decision making is a comprehensive procedure. In our approach, administrators can make final decisions about action by examining analysts' results, the task reward values and analysts' performance scores.

The administrators do not need to know the details of each analyst's work, but the system allows them to change task reward values and assign tasks to analysts when they can not make final conclusions by unclear sources such as the tasks contain uncertain suspicious nodes.

3.3.3 Communication

Sharing information: Sharing information is important in collaborative work groups [6]. Brennan et al. [2] built a collaborative framework among multiple analysts. In this framework, they focused on the idea of common grounded [5] communication, which allowed multiple analysts to share information, especially the reasoning behind the information, logically and graphically. Sharing information in our system is based on this framework. Analysts can point out their findings in generated images and send their findings with conclusion and suggestion to a sharing space in server. They can update the lists of malicious nodes information with confidence values. They are also permitted to write notes based on their own authority and expertise.

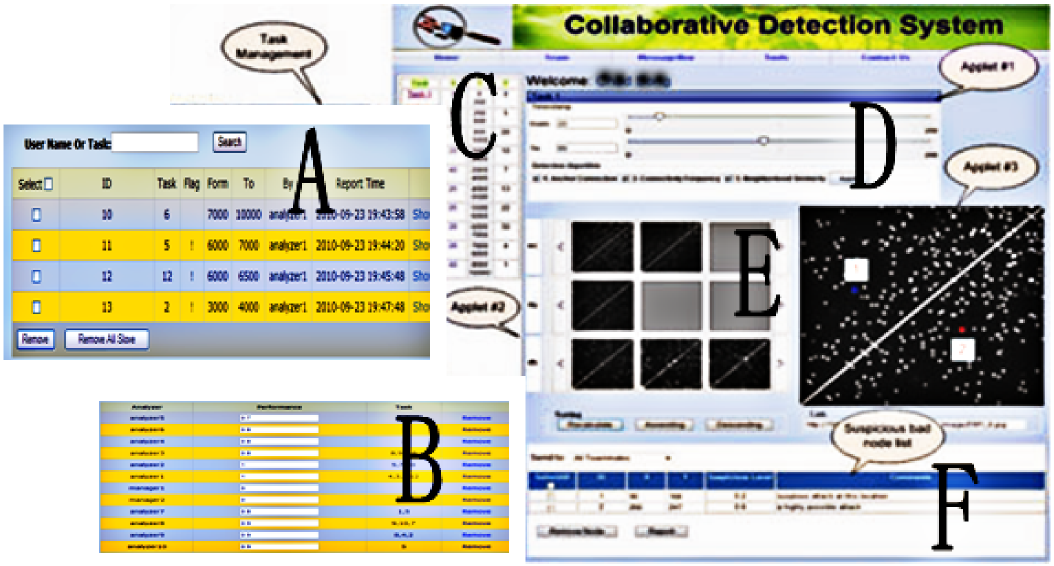
Awareness: One important aspect of communication is to provide the work status of each team member to the others. For distributed work groups, it is difficult to maintain awareness of the other members' work status [10] because of geographical distance. The traditional ways of maintaining awareness in distributed work groups (such as email) were demonstrated to be inefficient [18]. To mitigate this, we design a new way for analysts to maintain situational awareness. In our system, the ongoing task list and the analysts working on tasks are provided to further enhance awareness. Analysts can view the examined and unexamined tasks and the work progress of other analysts.

4 Collaborative Detection System

We apply the above design guidelines to develop a prototype system. We choose a web-based solution, as it is convenient for a group of people to monitor and defend a network collaboratively in a distributed environment. That is, through the web-based collaborative platform, multiple network analysts and administrators can work collaboratively towards identifying suspicious network events. Intelligent control mechanisms are also used for user management, task management, and collaborative decision making. The following describes the interface and implementation of such a web-based tool.

4.1 Interface Design

Figure 3 demonstrates the user interface design of the web-based collaborative detection system. The functionalities supported by the interface can be categorized into three types: (1) Administrative functionalities: For the purpose of effective user interaction, users are required to register and login before utilizing the functionalities supported by the system. An administrative interface is provided for administrators. Administrators can check the tasks detail information(Figure 3(C)), report time and suspicious nodes list (Figure 3(F)) and the performance list (Figure 3(B)). They make final decisions by considering all the information comprehensively. They can indicate the final conclusion of the tasks by adding 'Flag' in working processing board(Figure 3(A)). If the 'Flag' is '!', it means that suspicious nodes have been found. They also can remove '!' if they decide the task is safe. (2) Visual analytics functionalities: The interface displays visual representation of abstract network data to a group of users. Chart controls are provided to accept necessary user interactions, from moving the mouse over a 2D location to clicking or double-clicking on that location, for marking suspicious nodes under attack. Analysts can also adjust sliders in Figure 3(D) to select different time ranges of the detected task and to select different algorithms. This interaction make analysts to get the precise location of the suspicious nodes. In Figure 3(E), the generated images by three different detection algorithms locate at different rows. Analysts can get a scaled image by double clicking the small one. In the scaled image, they



Select □	ID	Task	Flag	Form	To	By	Report Time	
□	10	6		7000	10000	analyzer1	2010-09-23 19:43:58	Sho
□	11	5	!	6000	7000	analyzer1	2010-09-23 19:44:20	Sho
□	12	12	!	6000	6500	analyzer1	2010-09-23 19:45:48	Sho
□	13	2	!	3000	4000	analyzer1	2010-09-23 19:47:48	Sho

Figure 3. A demonstration of the graphical user interface for collaborative detection. (A) working processing board. (B) performance list for administrators. (C) the task list. (D) a panel for selecting time ranges and detection algorithms. (E) topology pattern window. The topology patterns are arranged into three rows which are results from three algorithms. When analysts double click one pattern, an enlarge image will be shown in the right window. (F) suspicious nodes list which is generated automatically based on suspicious nodes identified by analysts. Analysts can identify suspicious level and write comments here.

can point suspicious nodes in red or good nodes in blue by clicking them. (3) Communication and coordination functionalities: The findings of each task can be shared among analysts. They can click 'report' button to upload suspicious node list with images and notes(Figure 3(F)) to tell the other analysts the reasons of their findings. Our interface also helps both analyzers and administrators manage and maintain lists of ongoing tasks, and for each task, keep its allocation status, its current reward value, and a list of suspicious bad nodes.

4.2 Implementation Detail

Following the Model-View-Controller design pattern, for reasons of flexibility, the implementation of the interface supporting visual analytics functionalities consists of three modules: the Data module, the Control module, and the Visualization module. For each user request, these three interacting components always work together to produce visual representations of the network data in a user-specified range. Specifically, the Visualization module sends requests to the Control module for the display content while the Control module sends requests to the Data module for the network data that is required for satisfying the display requests.

5 Discussion

Here we discuss several different scenarios of attacks: no attack, simple attack, and complex attack. It is important that a distributed collaboration system is able to handle all the cases as each case has potential pitfalls.

In the case that there are no attacks in a network, all analysts should be made aware of this fact. In a traditional setting that relies on automatic intrusion detection systems, false alarms are common. As a result, many analysts are kept busy with verifying that false alarms are indeed false alarms. In our system we rely on the strength and diversity in abilities of a group of analysts to assess the state of a network. As such, when the network is in a safe state, our collaborative tools allow this fact to propagate to all analysts and administrators, who may then reduce the number of analysts actively working on tasks and let them either explore historical data or devote their time to other tasks.

The second case is where there is a simple attack. A potential pitfall in this situation, particularly when several analysts are processing the network data, is repeated work. That is, in a system that employs several analysts but fails to have adequate communication capabilities, several analysts may go through the process of identifying the simple attack. However, since our distributed system provides communicative functionality to both administrators and other analysts, the analysts who have not yet processed the attack will be made aware of the fact that an attack has been identified, and may then assist in verifying this conclusion. By verifying the conclusion, the analysts are updating the suspiciousness degree, which will prompt final action by an administrator more quickly than traditional communication methods.

Finally, the case of a complex attack is when attack nodes are migrating and exhibiting other complex behaviors. Such an attack may be first discovered by an analyst. However, it is more likely that the administrators, who are concerned with processing conclusions from the analysts, will identify patterns based on the results submitted by the analysts. At this point, administrators can produce and deploy a verification and response plan using the provided visualizations and tools. Our system gives administrators the ability to respond rationally (with the correct amount of analysts) and in a timely manner.

An additional example of a related complex scenario is when conflicts arise among different analysts' conclusions for a task. For some nodes of the task, different analysts may draw different conclusions based on their respective expertise. For example, analyst A finds a malicious node in some task's data. As a result, analyst A will increase the reward value of this task. Before A sends his results to the server, analyst B processes the same task. However, analyst B identifies the node that has been identified as malicious by A to be benign. As a result, this task is viewed as having conflicting results and the reward value is increased further. Our goal then becomes to eliminate the conflict. Administrators can identify such situations and assign more analysts to process this task while updating the respective performance and correctness scores of A and B. Based on the results from additional analysts and their respective performance scores, administrators will be able to draw conclusions about the task in question.

Another interesting situation arises when the reward value of the task is the same but different malicious nodes are found. To address this in our design, we provide a task list and corresponding suspicious node list with suspiciousness degrees to all participants. The administrator can assign nodes with high suspiciousness degrees as tasks to analysts for verification, and finally make decisions based on the analysts' conclusions.

6 Expert Feedback

As an important component of the evaluation procedure, we have provided the prototype system to four researchers: two visualization researchers, one wireless network security researcher and one web-based collaborative analysis researcher. The following summarizes the positive feedback from three aspects. Limitations and future work are summarized in the next section.

First, the feedback shows that the hierarchical organization of the user roles matches the scenarios of many real-life applications. A system administrator often has several assistants in managing a complex wireless network. Once the distribution of the work responsibilities among the assistants is determined, they usually have a great degree of flexibility in accomplishing their tasks. At the same time, the administrators have the authority to integrate the results from the analysts and make the final decisions. One feature that distinguishes the proposed approach from several voting based attack detection schemes is that the analysts can choose their own tasks based on their expertise, processing capabilities, and rewards. It liberates the administrators from the overhead of task assignment so that they can focus more on the result integration procedure.

Second, the proposed approach provides a powerful and convenient vehicle for communication among the analysts. The system provides two channels for the analysts to share their observations and localized detection results. First, they can identify the suspicious areas in the network so that other analysts can conduct detection at a finer granularity in the areas. Second, the analysts can directly share the suspicious that they identify and assist other analysts in their tasks. Sharing only the suspicious areas and the detection results will greatly reduce the communication overhead among the analysts.

Third, the proposed approach provides methods to measure the performance of analysts. The schemes include the reward incentives and the performance monitoring procedures. The reward incentives inspire the rational analysts to carefully conduct the attack detection tasks to maximize their evaluation effectiveness. At the same time, the cross-comparison between the analysts' results and the final decisions of the administrators prevents the analysts from trading detection accuracy for response time. The two schemes together can reduce the false positive and false negative alarms. At the same time, the degradation of the performance of any analysts can be easily discovered by the administrators.

7 Limitations and Future Work

One limitation of the present work is that the communication among distributed team members is limited. We plan to improve it for time-critical applications by allowing analysts to share ongoing results through introducing uncertainty visualization to our system. Another limitation is that it is still challenging for administrators to make decisions when conflict results occur. Thus additional decision making tools are necessary for such case. We plan to study relevant work from social science to improve the workflow for this purpose.

We hope that our work will bring about a discussion on exploring collaborative analysis methods for information security applications, especially in distributed settings. Our future work plan includes exploring the effects of different interaction and algorithm design aspects. We plan to design a series of simulations for use in user studies. The results of user studies will be used to adjust our approach and provide design guidelines for general collaborative analysis in distributed environments.

References

[1] V. Anupam, C. Bajaj, D. Schikore, and M. Schikore. Distributed and collaborative visualization. *Computer*, pages 37–43, 1994.

[2] S. Brennan, K. Mueller, G. Zelinsky, I. Ramakrishnan, D. Warren, and A. Kaufman. Toward a multi-analyst, collaborative framework for visual analytics. *Symposium On Visual Analytics Science And Technology*, 0, 2006.

[3] K. W. Brodlie, D. A. Duce, J. R. Gallop, J. P. R. B. Walton, and J. D. Wood. Distributed and collaborative visualization. *Computer graphics forum*, 23:223–251, 2004.

[4] H. H. Clark. Pointing and placing. In S.Kita(Ed),Pointing, Where language, culture, and cognition meet, 2003.

[5] H. H. Clark and S. E. Brennan. *Grounding in communication.* Perspectives on socially shared cognition, 1991.

[6] J. N. Cummings. Work groups, structural diversity, and knowledge sharing in a global organization. *Manage. Sci.*, 50(3), 2004.

[7] M. Egidi and L. Marengo. Division of labour and social coordination modes - a simple simulation model, 1993.

[8] D. Ferebee and D. Dasgupta. Security visualization survey. In *Proc. of the 12th Colloquium for Information Systems Security Education*, 2008.

[9] C. for Technology in Government University at Albany / SUNY. An introduction to workflow management systems models for action project:developing practical approaches to electronic recordsmanagement and preservation, 1997.

[10] S. R. Fussell, R. E. Kraut, F. J. Lerch, W. L. Scherlis, M. M. McNally, and J. J. Cadiz. Coordination, overload and team performance: effects of team communication strategies. In *Proc. of the 1998 ACM conference on Computer supported cooperative work*, pages 275–284.

[11] D. R. Garrison. Online collaboration principles. *Asynchronous Learning Networks*, 10, 2006.

[12] I. J. Grimstead, D. W. Walker, and N. J. Avis. Collaborative visualization: A review and taxonomy. In *Proc. of the 9th IEEE International Symposium on Distributed Simulation and Real-Time Applications*, pages 61–69, 2005.

[13] C. Gutwin and S. Greenberg. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. In *Proc. of the 9th IEEE International Workshops on Enabling Technologies*, pages 98–103, 2000.

[14] J. Heer and M. Agrawala. Design considerations for collaborative visual analytics. In *Proc. of the 2007 IEEE Symposium on Visual Analytics Science and Technology*, pages 171–178.

[15] J. Heer, F. B. Viégas, and M. Wattenberg. Voyagers and voyeurs: Supporting asynchronous collaborative visualization. *Commun. ACM*, 52(1), 2009.

[16] P. Isenberg and S. Carpendale. Interactive tree comparison for co-located collaborative information visualization. *IEEE Transactions on Visualization and Computer Graphics*, pages 1232–1239, 2007.

[17] G. Klein. Features of team coordination. New Trends in Cooperative Activities:Understanding System Dynamics in Complex Environments, 2001.

[18] R. E. Kraut and P. Attewell. Electronic mail and organizational knowledge: Media use in a global corporation. Technical report, 1993.

[19] A. Lu, W. Wang, A. Dnyate, and X. Hu. Sybil attack detection through global topology pattern visualization. *Information visualization*, 2010.

[20] K.-L. Ma and C. Wang. Social-aware collaborative visualization for large scientific projects. In *International Symposium on Collaborative Technologies and Systems (CTS)*, pages 190–195, 2008.

[21] D. C. Neale, J. M. Carroll, and M. B. Rosson. Evaluating computer-supported cooperative work: Models and frameworks. In *Proc. of the 2004 ACM Conference on Computer Supported Cooperative Work*, pages 112–121, 2004.

[22] R. Noss, C. Hoyles, J.-L. Gurtner, R. Adamson, and S. Lowe. Face-to-face and online collaboration: appreciating rules and adding complexity. *International journal of Continuing Engineering Education and Lifelong Learning*, 12:521–540, 2002.

[23] S. Shipman and J. Wholey. Performance measurement and evaluation: Definitions and relationships, 1998.

[24] H. Wang, T. Liu, L. Qiao, and S. Huang. Implementation of a web-based collaborative process planning system. In *Proc. of the 6th international conference on Cooperative design, visualization, and engineering*, pages 19–26, 2009.

[25] Y.-S. Wu, B. Foo, Y. Mei, and S. Bagchi. Collaborative intrusion detection system (cids): A framework for accurate and efficient ids. In *Proceedings of the 19th Annual Computer Security Applications Conference*, ACSAC '03, 2003.

[26] Y. Zhao, C. Hu, Y. Huang, and D. Ma. Collaborative visualization of large scale datasets using web services. In *Proc. of the Second International Conference on Internet and Web Applications and Services*, page 62, 2007.

Appendix

Sybil attacks [19] are particularly harmful on distributed systems and wireless networks. This attack has been demonstrated to be detrimental to many important network functions. Malicious nodes play the roles of multiple legitimate members of a network by impersonating their identities or claiming fake IDs. These fake nodes do not have real physical devices like legitimate nodes and they often claim to have direct or indirect connections with the malicious nodes which generate them.

In our system, users study the accumulated global network topologies. They detect Sybils by locating anomalies in neighbor relationships and movement patterns of wireless nodes. With the tight integration of interactive visualization and security algorithms, the system can be used to detect Sybil attacks under more sophisticated scenarios.

Here we briefly describe the three detection algorithms. **Algorithm 1–Anchor Connection**: The first algorithm is designed based on the fact that there is usually a lack of direct connectivity between Sybil and legitimate nodes and long-time connections among Sybil nodes. This method results as the middle pattern in Figure 2, empty regions on upper left and bottom right.

Algorithm 2–Connectivity Frequency: The second algorithm is designed according to the high connectivity feature among fake identities. This method results a highly connected region on the bottom left, which is shown to be white blocks in target patterns in Figure 2.

Algorithm 3–Neighborhood Similarity: The third algorithm is designed according to the neighbor similarities of a node group. This feature results regions with band patterns on the upper left and bottom right as the right pattern in Figure 2.