# Visual Mining Intrusion Behaviors By Using Swarm Technology

Xiaohui Cui, Justin Beaver, Thomas Potok
*Computational Science and Engineering Division*
*Oak Ridge National Laboratory*
*Oak Ridge, TN 37831, USA*
*Email: cuix@ornl.gov*

Li Yang
*Department of Computer Science and Engineering*
*University of Tennessee at Chattanooga*
*Chattanooga, TN 37403, USA*
*Email: Li-Yang@utc.edu*

***Abstract*—The alerts produced by the real time intrusion detection systems, e.g. Snort, can be difficult for security administrators to efficiently review and respond to, due to the enormous amount of messages generated in a short time frame. In this research, we developed a technique, the swarm based visual data mining approach (SVDM), to help user gain insight into the alert event data of the intrusion detection system, come up with new hypothesis, and verify the hypothesis via the interaction between the human and the system. The SVDM system can efficiently help security administrators detect anomaly behaviors of malicious user in a large volume of high dimensional time-dependent state spaces. The visual representation from this system exploits the human being's innate ability to recognize patterns and utilizes this ability to help security administrators understand the relationship between the seemingly discrete security breaches.**

***Keywords*-swarm; intrusion; data mining; visualization;**

## I. Introduction

Intrusion detection systems (IDS) are used to monitor network traffic and behaviors of information systems for the possible security breach, such as Vulnerability scanning and denial of service. IDSs generate alert messages to report abnormal network actions upon detection of it. These alert messages (alerts) are reported to security administrator who will make decision and respond to alerts. However, the real-time IDS, such as Snort, may generate enormous number of security alerts with large amount of security attributes on high-traffic networks. These alerts can make it difficult for security administrators to efficiently review and respond to the intrusions. Based on these alert messages, it is also difficult for the current IDSs to provide adequate defenses against sophisticate attacker attack [15]. Further more, the IDSs have difficulty detecting "low and slow" attacks designed specifically to evade the IDS detection [14]. The underlying reason is that these IDSs cannot deal with a large number of security alerts with high number of attributes due to their shortcomings in knowledge discovering of the unknown pattern or behaviors from large scale high dimensional security alert messages. The alerts generated by IDSs can be represented as stream datasets containing collections of data objects that alter in real time, due to continuously time-dependent data updating [1]. The detection of cyber threat and sophisticate attack is equivalent to the knowledge discovery from the large scale of high dimensional data streams. To facilitate early and accurate detection of the cyber threat, it requires uncovering the previously unknown relationships, the changing trends, or the cyclic behaviors that are hidden within the data streams.

## II. Visual Intrusion Detection Concept

We believe the challenge for intrusion detection in cyber attack data streams can be addressed by processing alerts using data clustering techniques and then visually presenting the dynamic graphic results to the network administrator. These dynamic graphic results can provide useful, easily understandable output and can result in a quicker response for decision making. In the present research, we developed a swarm based visual data mining technique (SVDM) that can dynamically transform the cyber security alert data stream into a large number of animated software agents actively moving in a virtual space to represent the evolving trend and behaviors of the data stream.

The SVDM system provides a real time visualization function for the raw IDS alert data. It assists network administrators in understanding the current state of a network and quickens the process of reviewing and responding to intrusion attempts. The system presented in this paper consists of two primary visualization components. The first component provides a visual mapping of the network topology that allows the end-user to easily browse clustered alerts. The second component is based on the swarm flocking behavior of birds. Each alert is represented as an individual particle featured with swarm flocking behavior. The particles can self organize themselves into different clustering to achieve data clustering result. This component allows the end-user to real time view the clustering process and provides an efficient means for reviewing alert patterns. The objective of this research is to use the SVDM approach to help user gaining insight into the network based IDS alert event data stream, coming up with new hypothesis, and verifying the hypothesis via the interaction between the human and the system. Our research have demonstrated that the principles of self-organization and swarm behavior simulation can be used to represent the dynamic evolution of the time-varying dataset and fuzzy relationship within the data streams from

different resources. Since the human is directly involved in the exploration process, the exploration goals can be easily shifted or adjusted if it is necessary. SVDM will target on analyzing the high dimensional alert messages generated from IDS, in which little pre-knowledge about the data is known and the exploration goals are vague. To start data visual analysis, firstly, SVDM IDS automatically extracts the alert event messages from the IDS dataset, the Splunk data warehouse [19]. Secondly, our prototype system will provide the user with an interactive interface that allows user test the hypothesis by modifying agents' motion parameters and/or observing the resulting change in behaviors. Furthermore, it is not necessary for the users to know in advance what kind of phenomena they should observe, since the visual representation of the data automatically emphasize all of the current important events or deviations, such as the abnormal behavior of a cyber system user. Moreover, future SVDM can provide novel interactive solutions to many knowledge discovery problems, including the discovery of frequently occurring patterns (motif discovery), anomaly detection, and query by content.

This novel IDS can efficiently facilitate the security administrators to detect anomaly behaviors of malicious users in the high dimensional time-dependent state spaces. The system's visual representations exploit the human being's innate ability to recognize patterns and utilize this ability to help the security administrator understand the relationship between the seemingly discrete security breaches. Security managers can look at the combination of data collected from firewalls, IDS, and host audit logs in one picture, which is more meaningful to interpret and explore. Furthermore, by combining the mission-impact analysis with the time-based representations, this novel system will be able to help organizations to efficiently and effectively predict the target computer and network, improving the respond speed to the system security threats.

In this paper, our research focus on how to use the SVDM approach to improve the network based IDS. Network based IDS have been deployed for a wide variety of purposes, ranging from identifying the anomalous users in a system to analyzing alerts in order to assist network administrators in determining the proper response procedure. However, the enormous amount of alerts generated on the high-traffic networks make it more difficult for the network administrators to efficiently review and respond to the data produced by the network-based IDS. In this paper, we demonstrated that processing and visualizing alerts by using SVDM approach will greatly improve the existing IDS with useful, easily understandable output for a quicker responsive decision to the alerts. Detailed analysis on how the visualization of IDS alert data assists network administrators in understanding the current topology of a network and on how network based IDS knowledge discovering capability can be improved by using this SVDM system for analyzing output will be presented in this paper.

In the following sections, we will first provide a brief background and related works of the IDS alert visualization research. After that, the mathematical theory of the SVDM will be described. In Section 4, the SVDM Framework will be presented. The SVDM framework consists of two primary components. The first component provides a visual mapping of the network topology that allows the end-user to easily browse clustered alerts. The second component is based on the flocking algorithm to demonstrate different attack behavior discovering models. The two components will be described in Section 5 and Section 6 will be the discussion and conclusion.

## III. Background and Related works

Many research on analyzing the large amounts of IDS alerts, detecting multistage attacks and visualizing them have been carried out in recent years. To understand multistage attacks, Mathew et al. [13] developed a technique using attack-track based visualization of heterogeneous event streams. They studied event correlation to track the multistage attack. But event correlation considers every occurred event to find the relation between the alerts, which is not an effective way due to the false detection by IDS. There are many techniques to reduce the large set of IDS alerts by considering only alerts, which generated frequently. In [10], Koike and Ohno have done exceptionally well to visualize the alerts in order to find the false detection. Though they have considered only the essential data to visualize, there is a need to analyze them further to make it more useful for network administrators to find future attacks. Where as in [18], they have grouped alerts on a temporal basis. Because it was totally focused on archiving snort alerts, further analysis has to be done to detect multi-stage attacks. The works in [7], [17] and [12] have furthered the research to visualize the essential events from a large set of alerts considering different information related to each alert using different techniques, such as trees and graphs. However, none of them concentrated on temporality, the flow of the alerts, and the source and destination at the same time. In [11], [4] they have showed, by applying data mining techniques on these alerts, that patterns could be generated effectively with a credibility for future use. We were motivated by [11], [4], [5] to use data mining techniques to find the rules and patterns through studying the essential features from each alert. Lee and Stolfo [5] have used data mining techniques to find user behaviors by considering system calls to develop Host based IDS (HIDS).

## IV. Mathematical Theory of SVDM

The SVDM technique is mainly based on the mathematical flocking simulation of birds or fish [3]. For the IDS time series, data streams collected from different sources. We assume one data stream is mirrored as one particle in

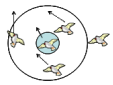


Figure 1. Alignment

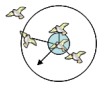


Figure 2. separation

a two dimensional virtual scene. Each particle's feature is a data vector that represents the value of the data stream the particle mirrored. As the bird flock in the real world, the particles that share similar features will automatically group together as a particle flock and other particles that have different feature will keep away from the flock. The motion of the particle represents the dynamic evolution of the alert data stream value. The faster the data stream value change, the fast the particle moves in the space. The screen output displayed to the user is a large number of animated data object particles representing the evolution trend of the data stream. The movement velocity of a particle A with position $P_A$ depends on all the data objects $X$ with position $P_X$ in its neighborhood. It is driven by a set of local behavior rules, including alignment rule, cohesion rule, and separation rule. The particle's velocity is also impacted by the feature similarity compared to the features of other particles. These impacts react upon successive data updates, thereby generating distinct emergent motion typologies, which can be easily visually interpreted by users. The three behavior rules and the feature similarity impacts can be illustrated by following mathematical equations:

### A. Alignment Rule

The alignment rule, as shown in Fig. 1, acts as if the active flock particle (outlined triangle located in the centre of the diagram) tries to align its velocity vector with the average velocity vector of the flock in its local neighborhood. The degree of locality of the rule is determined by the sensor range of the active flock particle and is represented diagrammatically by the circle. The mathematical implementation is:

$$\begin{aligned} &d(P_x, P_b) \leq d_1 \cap d(P_x, P_b) \geq d_2 \Longrightarrow \\ &\overrightarrow{v_{ar}} = \frac{1}{n} \sum_x^n \overrightarrow{v_x} \end{aligned} \tag{1}$$

Where $v_{ar}$ is the velocity driven by alignment rule, $d(p_x, p_b)$ is the distance between particle B and its neighbor X, $n$ is the total number of particle B's local neighbors, $v_x$ is the velocity of particle X, $d_1$ and $d_2$ are pre-defined distance values and $d_1 > d_2$.

### B. Cohesion Rule

The cohesion rule, as shown in Fig. 3, acts as if the active flock particle tries to orient its velocity vector in the direction of the centroid (average spatial position) of the local flock.

Figure 3. Cohesion

$$\begin{aligned} &d(P_x, P_b) \leq d_1 \cap d(P_x, P_b) \geq d_2 \Longrightarrow \\ &\overrightarrow{v_{cr}} = \sum_x^n \overrightarrow{(P_x - P_b)} \end{aligned} \tag{2}$$

Where $v_{cr}$ is the velocity driven by cohesion rule, $d_1$ and $d_2$ are pre-defined distance and $\overrightarrow{(P_x - P_b)}$ calculates a directional vector point.

### C. Separation Rule

The separation rule, as shown in Fig. 2, acts as if the active flock particle tries to pull away before crashing with each other. The mathematical implementation is:

$$d(P_x, P_b) \leq d_2 \Longrightarrow \overrightarrow{v_{sr}} = \sum_x^n \frac{\overrightarrow{\overrightarrow{v_x} + \overrightarrow{v_b}}}{d(P_x, P_b)} \tag{3}$$

Where $v_{sr}$ is the velocity driven by separation rule, $d_2$ are pre-defined distance, $v_b$ and $v_x$ is the velocity of particle B and X.

### D. Feature Similarity Rule

The flock particle tries to stay close to other particles that have similar features and to stay away from other particles that have dissimilar features. The strength of the attracting force is proportional to the distance between the particles and to the similarity between the particles' feature values. And the strength of the repulsion force is inversely proportional to the distance between the particles and the similarity between the particles' feature values. In this research, rather than directly use the feature similarity/dissimilarity rule, we nullified the alignment and cohesion rules when $S(B, X) < T$. $S(B, X)$ is the similarity value between the features of particle $B$ and $X$ and $T$ is similarity threshold. Thus, for dissimilar particles (The similarity value $S$ smaller than the threshold $T$), separation rule is the only active rule, causing them to repel one another. The similarity between two alert particles is expressed as a number within the range 0 to 1, with 0 for no similarity to 1 for identical. We use Q-grams algorithm [6] to determine similarity which approximate string matching by "sliding" a window of length $q$ over the characters of a string to create a number of $'q'$ length grams for matching a match is then rated as number of q-gram matches within the second string over possible q-grams. The intuition behind the use of q-grams as a foundation for approximate string processing is that when two strings s1 and s2 are within a small edit distance of each other, they share a large number of q-grams in common. Consider the following example. The positional q-grams of length q=3 for string "sam chapman" are $f(1, \#\#s)$, $(2, \#sa)$, $(3, sam)$, $(4, am)$, $(5, mc)$,

$(6, ch)$, $(7, cha)$, $(8, hap)$, $(9, apm)$, $(10, pma)$, $(11, man)$, $(12, an\%)$, $(13, n\%\%)$, where $'\#'$ and $'\%'$ indicate the beginning and end of the string. So getting the q-grams for two query strings allows the count of identical q-grams over the total q-grams available. This method is frequently employed as a 'fuzzy' search in databases to allow non exact matching of queries. A rating is also often given by the ratio of the q-grams distance metric.

## V. Visualization Framework

In the present project, the log processing and alert clustering process is accomplished using server-side Java code, and the alerts are presented to the end-user using two separate front-end applications developed in Adobe Flex. The front-end applications communicate with the server-side application using the open-source Blaze DS framework. One front-end application presents a mapping of each node on the network that has generated an alert. The administrator can quickly get an overview of the traffic flow on the network and is able to see the quantity alerts each of these nodes has produced, the time when they were produced, and a detailed list of each alert. The second front-end application presents a visual flocking model where each alert is represented by a color-coded circle. Our project uses network traffic captured over a five day period involving over 300 different network nodes.

### A. Snort Alert Pre-Processing

The IDS utilized in our project is Snort [20], a well-known and widely-used rule based IDS. Snort generates alerts by matching incoming packets against the built-in rules. The alerts generated by Snort are indexed by Splunk. We developed a software that can retrieve the alert message through the Splunk API and place the retrieved output into an output file for further analysis. Each entry in the output file lists detailed information associated with an alert, including the alert type, the alert generation rule, the source and destination of the IP addresses and ports, and the time at which the alert was generated to millisecond precision. As we mentioned in the introduction section, one shortcoming of Snort is the tremendous amount of alerts that are generated when the tool is deployed on a busy network. The large number of alerts makes it difficult for security administrators to quickly decipher and respond to intrusion attempts. It is necessary to reduce the number of alerts that the network administrator is presented with by grouping the similar alerts together and reducing each alert size.

It is very important to analyze snort alerts and convert them into a generalized form by considering only essential attributes such as source and destination IP address, source and destination port numbers, destination port number, time stamp of the alert and the type of the alert. Below is a sample alert from Snort.

```
[**] [1:2000537:6] ET SCAN NMAP -sS [**]
[Classification: Attempted Information Leak] [Priority: 2]
07/17-09:30:09.298097 192.168.101.66:33966 -> 192.168.101.53:175
TCP TTL:49 TOS:0x0 ID:27814 IpLen:20 DgmLen:44
S* Seq: 0x3C25204F  Ack: 0x0  Win: 0x800  TcpLen: 24
TCP Options (1) => MSS: 1460
```

Figure 4. Sample Alert from Snort

```
(portscan)TCPPortscan , 07/17-10:03:27.114495,
192.168.101.66,3387, 192.168.101.54,4497.
```

Figure 5. Snort Alert with Reduced Features

In one of our experiments, we used around 3GB of network traffic dataset that consists of millions of network packages. Out of those millions of packets we could generate 150K snort alerts of size 63MB. As we need to extract common details in each alert, less significant information details such as protocol information are ignored. After extracting common and essential attributes, 50K alerts were reduced to 3K in number and 360KB in size. A sample alert is in the format (Attack type, timestamp, Source IP, Source port, Destination IP, Destination port) mentioned above. The following is an example of our Snort alert:

We chose to group similar alerts together on temporal basis. We assume that alerts can be grouped into one if they share the same Source IP address, alert type, and occur in a specific and short time span. This approach is different from the one in [11], where they group all the alerts with same source. Our approach is effective in eliminating false detections and also to find sequential pattern of attacks on temporal basis, which plays significant role in finding potential multistage attacks. After the grouping has been done, we have added two more columns to alerts one as count, the number of alerts grouped together, and the other one is the behavior code to help while visualizing the alerts. The behavior code indicates similarity of current alerts with other alerts, which help to cluster similar alerts. A sample alert after grouping looks like the one below with the last attribute 50 as the number of grouped alerts.

### B. Alerted Network Traffic Topology

The network traffic visualization component processes the raw Snort alert data into a visual format, displaying each node on the network and its associated alerts and connections. Details of alerts, such as full network addresses, service, port numbers, durations, can be presented to the network security administrator on demand. Our model, as

```
192.168.101.56, 192.168.72.1, 07/17- 10:59:06,
ETSCANNMAP, 70, 50
```

Figure 6. Grouped Snort Alert

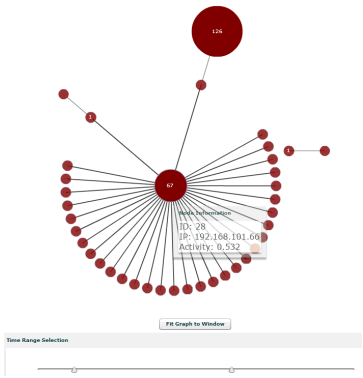



Figure 7. Alerted Network Traffic Topology

shown in Fig. 7, is able to visualize a network topology and its alert traffic based on the raw Snort alerts. A network topology map is generated to allow the end user to efficiently access details of the network nodes and their associated alerts.

## VI. Visual Data Mining the Attack Pattern

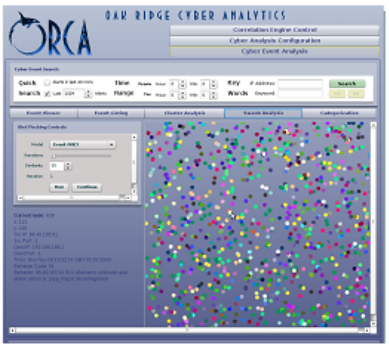
The SVDM system provides multiple visual models to offer the users a full view of current attack status. It is suitable for the enterprise level cyber security information control.

### A. Event Only Model

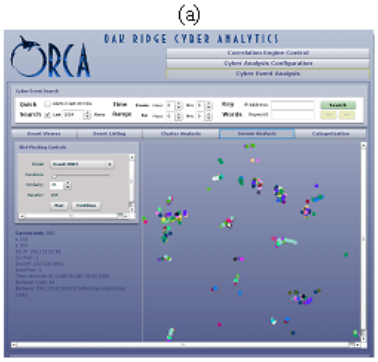
As shown in Fig. 8(a), in this model, each event represented as a dot in the visual space. Moving mouse on top of each dot, the system will automatically display the detailed information about this alert event. The system also provides an interface for adjusting the parameters of the particles' movement. To be easily visualized by the human eyes, as shown in Fig. 8, the system uses randomly generated colors representing different alert events. By using the flocking algorithm presented in the previous section, all alert events can be quickly self-organized into multiple groups. The alert events that swarmed together are attack events with features similar to each others. It gives the user a high level view about the diversity of the attack behaviors. The visual results displayed in Fig. 8(b) indicated that although the system received thousands of alert events, there are actually only 10 to 12 different attack behaviors. Most events are generated by the similar attack behaviors.

### B. Cyber Attack Behavior Construction Model

Most of the attacks compose a serious of malicious behaviors. These malicious behaviors will be represented as individual events in IDS alert event message information if they are discovered by the IDS. Discovering the dependency of these events will help us building the attack link models for predicting the potential attacks and assessing the attack damage (Fig. 9). In the current stage, we manually discover




(a)




(b)

Figure 8. Events Only Model

the malicious events from the customer's sample IDS data and build the attack behavior patterns for different classes of attacks. The patterns are stored as reference patterns and used for model 3 and 4 discussed in the next sections.

### C. Simultaneous Attack Discover Model

In this model, the SVDM system focuses on analyzing the attacking behavior of each individual IP address. We hypothesize that some attackers or hack organization launch simultaneous cyber attacks from different machines with different IP addresses. Although the attacks are launched on different machines, the attack strategies, methods, and tools are most likely the same. The alert sequences triggered by these different IP address should be similar. These alert sequences can be considered as the attacking behavior of the IP address. In the demo figures, each colored dot represents one single IP address and the attacking behaviors of this IP address have been conducted during a period of time. The system can quickly group the different IP addresses with the similar attacking patterns into one group, as shown in Fig. 10. If one group in the system appears significantly larger than the other groups, it might indicate the organization are under simultaneous attack from different IP addresses.

### D. History Behavior Discovery Model

Most experienced criminals have a relatively fixed behavior pattern when they conduct crime, similar to the cyber

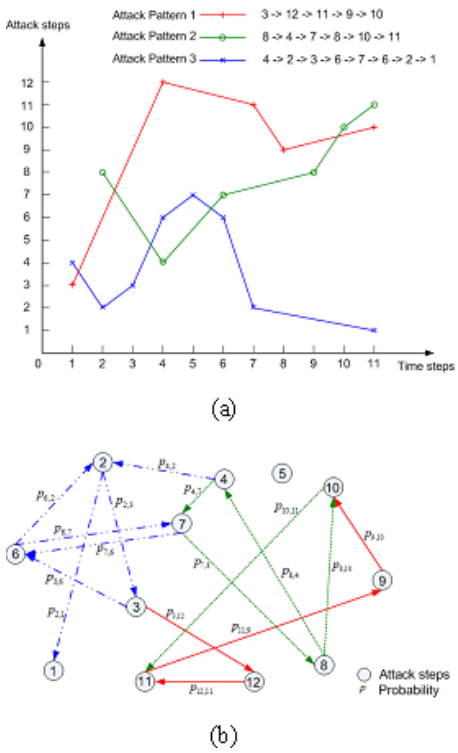



Figure 9. Cyber Attack Behavior Construction Model

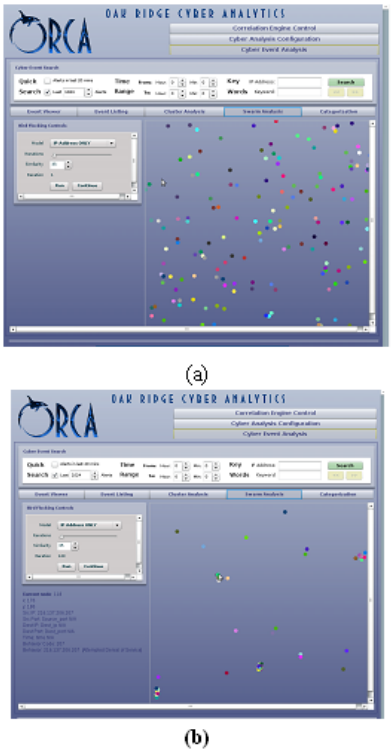



Figure 10. Simultaneous Attack Discover Model

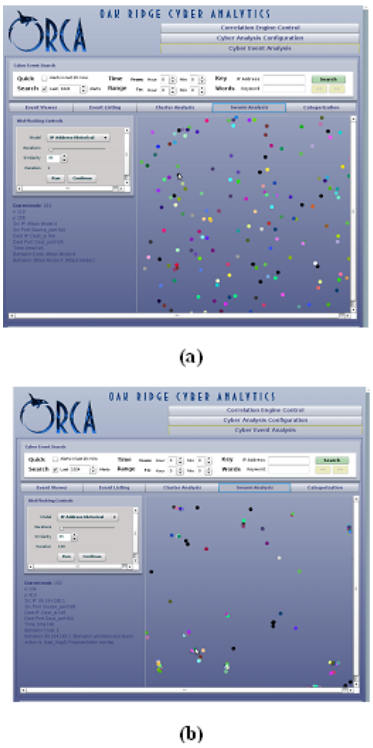



Figure 11. History Behavior Discovery Model

attacker. They intend to use similar methods or tools to conduct attacks on different machines. We can analysis the history data to build different attacker behavior models and compare them with current data to find the similarity between the patterns from current data and the history model. In Fig. 11, each colored dot represents one single IP address and the attacking behaviors on this IP address have been conducted in recent times. The black dot represents attacker behavior models that we build up from the history data. When the color dots grouped with the block dot, it indicates the color dot IP is using a similar attack method as that described in the history dataset represented by the black dot that is near the color dots.

## VII. Conclusion

SVDM is a novel approach for the cyber alert information analysis. First, SVDM will automatically extract all of the important events or deviations and detect specific patterns and properties from the high dimensional data streams. However, in conventional information analysis system, the human user has to generate a hypothesis prior to the moment, which will introduce bias. Second, our prototype system will provide the user with an interactive interface to allow user test the hypothesis by modifying agents' motion parameters and observing the resulting behavior changes. Third, users do not need to know in advance what kind of phenomena they should observe, since the visual representation of the

data automatically emphasize all of the current important events or deviations, such as the abnormal behavior of a cyber system user.

Our project has demonstrated how principles of self-organization [16] and swarm behavior [9] simulation can be used to represent the dynamic evolution of the time-varying dataset and fuzzy relationships within the data streams from different resources. Since the human is directly involved in the exploration process, shifting and adjusting the exploration goals can be easily done if it is necessary. SVDM will target on analyzing the high dimensional data streams, in which little knowledge about the data streams is known and the exploration goals are vague.

## ACKNOWLEDGMENT

This work was supported in part by the Lockheed Martin Corporation DashNet program. The views and conclusions contained in this document are those of the authors. This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

## REFERENCES

[1] Bass T., Intrusion Detection Systems and Multisensor Data Fusion, Communications of the ACM, Vol. 43, No. 4, pp 99-105, April, 2000.

[2] Copeland A., Tool update: high alarm count issues in IDS rainstorm, Proceedings of the 3rd international workshop on Visualization for computer security, Alexandria, Virginia, USA, 2006.

[3] Cui X., Gao J. and Potok T., A Flocking Based Algorithm for Document Clustering Analysis, Journal of System Architecture, June,(2006)

[4] Dokas P., Kumar L., Lazarevic V., Srivastava J., Tan P., Data Mining for Network Intrusion Detection, USA, 2004.

[5] Ghosh A. K., Schwartzbard A. and Schatz M., Learning Program Behavior Profiles for Intrusion Detection. Proceedings of the 1st conference on Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California, 2002.

[6] Gravano L. and Ipeirotis P. G., Jagadish H. V., Koudas N., Muthukrishnan S. Pietarinen L. and Divesh Srivastava. Using q-grams in a DBMS for approximate string processing. IEEE Data Engineering Bulletin. pp.28-34, vol. 24, 2001.

[7] Hertzog P., Visualizations to improve reactivity towards security incidents inside corporate networks, Proceedings of the 3rd international workshop on Visualization for computer security, Alexandria, Virginia, USA, 2006.

[8] Han J. and Kamber M., Data mining concepts and techniques. Academic Press, San Diego, California, 2001

[9] Kennedy J. and Eberhart R. C., Swarm Intelligence. San Mateo, CA: Morgan Kaufmann, 2001.

[10] Koike H., Ohno K., SnortView: Visualization System of Snort Logs, Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, Washington DC, USA, 2004.

[11] Lee W., Applying data mining to intrusion detection: the quest for automation, efficiency, and credibility, ACM SIGKDD Explorations Newsletter, December, 2002.

[12] Mansmann F., Fisher F., Keim D. A., North S. C., Visual support for analyzing network traffic and intrusion detection events using TreeMap and graph representations, Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology, Maryland, USA, 2009.

[13] Mathew S., Giomundo R., Uoadhyaya S., Sudit M., Slotz A., Understanding multistage attacks by attack-track based visualization of heterogeneous event streams, Proceedings of the 3rd international workshop on Visualization for computer security, Virginia, USA, 2006.

[14] Matzner S., Sinclair C., Pierce L., An application of machine learning to network intrusion detection. In Proceedings of the 15th Annual Computer Security Applications Conference, pages 371?377, Phoenix, AZ, 1999.

[15] McHugh J. Intrusion and Intrusion Detection, International Journal of Information Security, 1: 14-35, 2001

[16] Ramos V., Merelo J., Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning, 1st Spanish Conference on Evolutionary and Bio-Inspired Algorithms, E. Alba, F. Herrera, J.J. Merelo et al. (Eds.), pp. 284-293, Merida, Spain, 2002, 6-8 Feb.

[17] Shabtai A., Klimov D., Shahar Y., Elovici Y., An intelligent, interactive tool for exploration and visualization of time-oriented security data, Proceedings of the 3rd international workshop on Visualization for computer security, Alexandria, Virginia, USA, 2006

[18] Soleimani M., Khosrowshahi E., Doroud M., Damanafshan M., Behzadi A., Abbaspour M., A Reliable Analyzer and Archiver for Snort Intrusion Detection System, Proceedings of the 2007 ACM symposium on Applied computing, Korea, 2007.

[19] Splunk Inc. Splunk: Search and analysis engine for IT data. [online], 20052009. http://www.splunk.com/

[20] The snort. http://www.snort.org.

[21] Tan P., Stenbach M., Kumar V.. Introduction to data mining. Pearson Addison Wesley. 2006.