A Visualization Scheme for Network Forensics Based on Attribute Oriented Induction Based Frequent Item Mining and Hyper Graph

Jianguo Jiang[1], Jiuming Chen[1,2], Kim-Kwang Raymond Choo[3], Chao Liu[1], Kunying Liu[1], and Min Yu[1,2(✉)]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
yumin@iie.ac.cn
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX, USA

Abstract. Visualizing massive network traffic flows or security logs can facilitate network forensics, such as in the detection of anomalies. However, existing visualization methods do not generally scale well, or are not suited for dealing with large datasets. Thus, in this paper, we propose a visualization scheme, where an attribute-oriented induction-based frequent-item mining algorithm (AOI-FIM) is used to extract attack patterns hidden in a large dataset. Also, we leverage the hypergraph to display multi-attribute associations of the extracted patterns. An interaction module designed to facilitate forensics analyst in fetching event information from the database and identifying unknown attack patterns is also presented. We then demonstrate the utility of our approach (i.e. using both frequent item mining and hypergraphs to deal with visualization problems in network forensics).

Keywords: Visualization · Big data analysis · Network forensic · Hypergraph

1 Introduction

In our increasingly Internet-connected society (e.g. smart cities and smart grids), the capability to identify, mitigate and respond to a cyber security incident effectively and efficiently is crucial to organizational and national security. Existing security products include those that enforce policies and generate situational intelligence [1, 2]. However, existing solutions are generally not designed to deal with the increasing volume, variety, velocity and veracity of data generated by existing security solutions [3].

Thus, in this paper, a visualization analysis scheme based on attribute oriented induction based frequent item mining and hyper graph is proposed. The choice of attribute oriented induction based frequent item mining algorithm and hyper graph is as follows. In network attacks, for example, using frequent item mining algorithm only allows us to extract records whose attributes meet the frequent character. In a host scan attack, however, the destination port number may varies. Thus, we use attribute

P. Matoušek and M. Schmiedecker (Eds.): ICDF2C 2017, LNICST 216, pp. 130–143, 2018.
https://doi.org/10.1007/978-3-319-73697-6_10

oriented induction based frequent item mining algorithm instead of only frequent item mining (FIM) algorithm to process network traffic data and security logs. This allows us to effectively filter out the redundant data and discover interesting patterns hidden in the data. In addition, several commonly seen attacks have a one-to-many relationship, which could be visualized and distinguished [7] Thus, using hyper graph, we can clearly display multi-attribute associations and the specific attack event information. In our scheme, we also include an interaction module for the forensics analyst to manual analyze the visualized event and obtain the original information of these events.

Our scheme is designed to handle both network flow data and security logs, and therefore, a forensic analyst can easily understand the behavior of hosts or users from the visualization graph when an attack occurs. Specifically, our scheme allows the forensic analyst to identify new anomaly and attack patterns using the graph visualization and the interaction module. The scheme can deal with big dataset using attribute oriented induction based frequent item mining, where multi-attribute relationship of parameters such as source IP, destination IP address, port number, and time, can be explored to provide in-depth information about malicious cyber incidents or events.

The remaining of this paper is structured as follows. In the next section, we review related literature. In Sects. 3 and 4, we describe our visualization scheme and demonstrate the utility of our scheme using real-world dataset, respectively. Finally, we conclude our paper in Sect. 5.

2 Related Work

Many approaches designed to handle and display complex data in networks have been proposed in the literature. Such approaches facilitate humans in recognizing abnormal events in the network [4–7]. Parallel coordinate, a commonly used visualization method proposed by Inselberg [8], displays multi-dimensional data. Specifically, in a parallel coordinate, each vertical axis represents a different attribute and the lines display the associations between two coordinates. There are a number of published parallel coordinate based visualization schemes and tools, such as NFSight [9], VisFlowConnect [10], and PCAV [11].

There are several other visualization tools, such as Nfsen [12], FlowScan [13], FloVis [14], NFlowVis [15] and Fluxoscope [16], which use a range of visualization methods (e.g. histograms, pie charts, scatterplots, heavy hitter lists, and tree maps). A key limitation in parallel coordinate based approaches and several other visualization approaches is that the lines they use in the graph can only indicate associations between two linked parameters. However, these approaches cannot visualize multi-attribute associations. In addition, the parallel coordinate approach cannot display the quantitative characteristics.

Krasser and Conti [17] used parallel coordinate for real time and forensic data analysis. Their approach displays both malicious and non-malicious network activities, but the approach does not scale well to deal with big dataset. The plane coordinate diagram, another popular approach used in the literature, can only represent the association between two attributes.

The hyper graph approach, however, can effectively display the association between the multi-attributes and facilitate a forensic analyst to reconstruct the event [18]. For example, Glatz et al. [19] proposed a hyper graph based approach to display traffic flow data. While the proposed approach in [19] visualizes dominant patterns of network traffic, the authors did not explain how their approach can be used to distinguish attacks features (i.e. a visualizing approach, rather than a visualizing analysis method). Unlike existing approaches, in this paper, we first analyze the "one-to-many relationship" in popular attack patterns that allows us to distinguish between the attacks. We then add an interaction module in our visualization scheme so that a forensics analyst can easily interact with the database and the hyper graph to discover unknown attack patterns.

There have also been efforts to using signature based methods, such as hash function, to handle the traffic flow data and mining interesting patterns [20]. However, these methods need to know the characteristics in advance and these methods' efficiency is limited when dealing with big dataset. Therefore, in our research, we use attribute oriented induction based frequent item mining algorithm to extract attack patterns hidden in the data. Frequent item mining algorithm is widely used in the field of data mining, but to the best our knowledge, our work is the first to leverage both hyper graph and frequent item mining to network forensic visualization.

3 Proposed Network Forensic Visualization Scheme

The complex and noisy network traffic and security logs can be simplified using visualization techniques or tools, which allows network forensic analysts to have an in-depth insight into the network and the activities (e.g. attack event information). For example, using visualization, we can deduce some new or unknown attacks in the network; thus, enabling unknown attack(s) to be detected. Key challenges in security visualization include data volume and the correlating methods. In order to mitigate existing limitations, we propose a data extraction method based on a frequent item mining algorithm to reduce the volume of the noisy data set. Specifically, we propose a hypergraph based method that allows the correlation of several parameters such as source address, destination address, source port, destination port, packet length and time.

3.1 Attack Features

There is no one size fits all visualization method, but we can design the graph on a case-by-case basis to fulfill specific needs. There are mainly four attack types, namely: scan attacks, denial of service attacks, worm attacks and other attacks (e.g. botnet facilitated attacks). In order to design an effective visual method for most popular attacks, their features must be considered. For example, these popular attacks have one common characteristics, which is "one–to-many relationship" between network attacker and victim/victim machine(s) [7]. The characteristics can inform the design of detection algorithm for maximal accuracy.

Network scanning attack is generally (one of) the first step(s) in an attack, such as a host or port scan to probe and identify vulnerable host(s) in the network and available service(s) of the target(s) host for exploitation. In both scanning processes, there is a "one to many relationship", in the sense of one attack host with one or many victim hosts and many ports.

Denial of service (DoS) and distributed denial of service (DDoS) attacks are another popular type of attack, seeking to exhaust and overwhelm the target network's bandwidth and computational resources. Similarly, such attackers have a "one-to-many relationship".

Worm is a self-propagating code whose propagation process is similar to botnet attack. After detecting the vulnerable machines in the network, an infected host may propagate the worm code to one or more target hosts. Therefore, we have a "one-to-many relationship" between the source infected host and the target hosts. Similar relationship is in botnet attacks, where the attacker propagates the malicious code to an infected host, and builds a relationship between controller and infected client hosts.

Attackers may also change or hide the parameters and find new vulnerability(ies) to increase the possibility of success and reduce the probability of detection. Such efforts will compound the challenges in detection. Thus, we need to identify avenues that can be used to effectively mitigate such efforts. One such avenue is in their (common) characteristics, as discussed above.

In this remaining section, we will show how to extract and display the characteristic, and when combined with the use of AOI-FIM algorithm and hyper graph, facilitates forensics analysis.

3.2 Attack Parameters

When an attack occurs, there are many logs (e.g. security device logs, system logs, web server application logs and network logs) containing information related to the event and could be used to reconstruct the event. For example, network flow data and security logs are two main sources in network forensics. In this paper, we use network flow data and security logs to collect the data for the following analysis. There are many parameters within flow data and network security logs, such as IP, port, time and alert type. We need to choose parameters that will be helpful to visualize the "one-to-many" relationship and distinguish the type of attacks for forensics analysis.

Firstly, the source IP address and the destination IP address are selected as parameters, which could be used to specify the victim and attacker. Secondly, Internet worms and botnet attacks may choose one or more ports to propagate the malicious code. Therefore, the port number is another parameter to be considered. Thirdly, network scanning and DDoS attacks generally make use of packets without payload or with fixed length such as 40 or 48, but Internet worms and botnet attacks generally have a fixed length payload of more than 48 (i.e. due to the malicious payload). Thus, we can use the packet length parameter to distinguish between different types of attacks. Finally, to distinguish one flow or multiple flows with the same value, we add the time of the flow to display the quantity characteristic of the network flow.

The following is a sample of a normalized record analyzed using the proposed visualization scheme. The SIP represents the source IP, and DIP represents the destination IP address. The SPort and DPort represent the source and destination port number of the flow data, respectively. The pSize represents the packet size when the data is flow data. The alert type represents the type of alert from a security log.

{SIP: x.x.x.x, DIP: y.y.y.y, SPort: t, DPort: p, time: xx-xx-xx, pSize: m, Alert Type: IRC}

3.3 AOI-FIM Algorithm

Analyzing network packets, logs and systems events for forensics has always been a challenge, due to the large data volume. Thus, we apply frequent item set mining algorithm to extract patterns hidden within the data, and visualize them using hyper graph to show the relationship between each parameter.

Frequent item set mining algorithm, a process that extracts patterns from transactions, is one of the most used methods to create association rules. Let $I = \{i_i \ldots, i_n\}$ be a set of parameters and the sum of parameters is n, and $D = \{t_{1\ldots}\ t_m\}$ be a set of transactions, where the sum of transactions is m [21]. Each transaction t_i contains the subset of parameters in I. A frequent item set is a transaction that appears at least s transactions in D. The parameter s determines the threshold size of frequent item sets. The parameter k determines the minimum number of parameters in each frequent item. In our context, we use network flow and security logs as transactions, and the parameters consist of source IP, destination IP, etc. The frequent item sets are a set of some traffic parameters which frequently appear in the data, such as {SrcIP = a.a.a.a, Sport = x, DestIP = b.b.b.b, DestPort = y}. The result of the frequent item mining process is a collection of the IP, port, and other parameters in the flow data attributes and security logs.

For a port scan attack, the port may be various and there is a frequent pattern between the varied port number and source IP. Using the traditional FIM algorithm, the various port numbers may not be detected as a single port does not occur frequently. In addition, directly using only a conventional FIM algorithm (e.g. Apriori) does not allow us to distinguish the types of attack automatically. Although many popular attacks meet "one-to-many" relationship, the frequent distribution of data cannot be extracted directly by many classical frequent mining items algorithms. To merge some records with multiple port numbers or multiple IP address into one frequent pattern, we apply the attribute oriented induction method [22] into the frequent item mining algorithm (AOI-FIM), which improves the detection precision.

Attribute oriented induction (AOI) algorithm is a useful method for knowledge discovery in relational database, which uses a machine learning paradigm such as learning-from-examples to extract generalized data from original data records [23]. The attribute-oriented concept tree ascension for generalization is the key to the AOI method, which can reduce the computational complexity of the database. Figure 1 is an example application of the AOI concept tree.

In this paper, we use the AOI method to redesign the FIM algorithm so that it can extract some specific and unknown attack patterns. Using the AOI method, the AOI-FIM algorithm can promptly extract attack patterns from normalized records that

have a "one-to-many" relationship. For some special frequent item set that meets the threshold requirement of frequent pattern but the value distribute of the parameters may vary, we use 'Vary' to merge the items and represent the distributed frequent pattern of the special parameter.

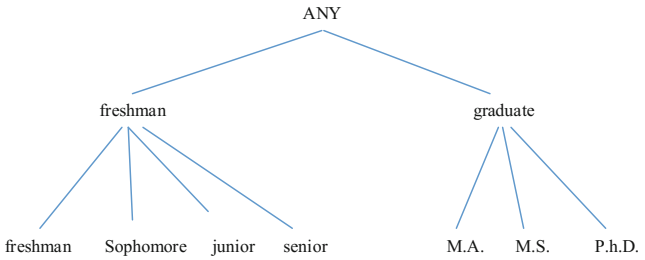

Fig. 1. A sample concept tree of AOI method

The attribute oriented induction based mining algorithm [24] can be used to generalize some records with a different parameter value into one frequent item set when the generated patterns meet the frequent rule. In the next subsection, we explain how the hypergraph theory can be used to visualize the frequent item sets for forensic investigations.

3.4 Hyper Graph

Existing network security log visualization approaches generally use parallel coordinates to represent the relationship between the network parameters. However, parallel coordinate cannot correlate the relationship between multiple attributes, or reflect the quantitative characteristics of various transactions. To overcome these limitations, we use hypergraph [25] to reflect the multi-attribute associations within the frequent item sets. Hypergraph also allows us to distinguish some specific attacks based on the displayed characteristics.

Mathematically, a hypergraph is a generalization of a graph, where an edge can connect any number of vertices [26]. Formally, a hypergraph H is a pair H = (X, E), where X is a set of elements, called nodes or vertices, and E is a set of non-empty subsets of X called hyper edges or links. In this paper, the nodes or vertices represent the parameters consisting frequent item sets and each hyper edge represents a frequent item set extracted from the flow data and security logs.

To display the quantitative characteristics of data, we add a circle to each hyper edge and the circles is varied based on the size of frequent item sets. We display the number within the circle node to represent the quantity characteristic within the frequent item set. In this paper, we seek to automatically extract the one-to-many relationship and find the hyper edges that are potentially associated with an attack.

Figure 2 shows some typical hyper edges that represent some popular attacks. Attacks that have a "one-to-many relationship" can be easily visualized and distinguished using hypergraph. What's more, the combination of some different frequent

items which meet one-to-many relationship can also infer some specific attacks such as botnets and Trojans. To display some frequents patterns with different parameter values, we use the attribute oriented induction based frequent item mining algorithm (AOI-FIM) to extract the attack patterns, and "Vary" to represent the parameter whose distribution of values is dispersed but the parameter and other frequent item sets have a one-to-many relationship. For example, we use a port-fixed DDoS attack as an example, and the graph shows that in this example, three distributed hosts are attacking target hosts y_1. y_2. y_3. y_4 via n ports. We use "Vary" to represent the multiple source hosts in the hyper graph to show the induced frequent quantitative relation.

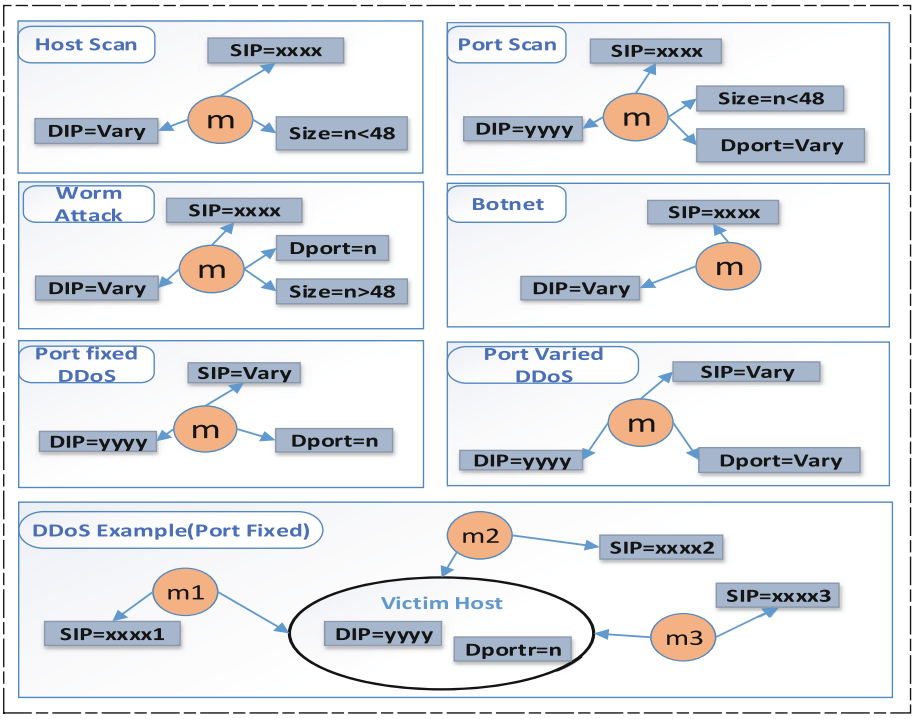

Fig. 2. Attack pattern visualized by hyper graph

3.5 Visualization System for Network Forensic

In order to automate the data correlation and displaying of event information for forensic investigation, we design a system based on frequent item mining algorithm and hyper graph. The architecture of the system is presented in Fig. 3, where the system consists of data collection, data pre-process, frequent item set mining, attack detection, hypergraph visualization, and manual inspection.

The system also has five main modules, namely: a collection module, analysis module, visualizer module, data store module and forensics interaction module. The collection module receives network traffic flow data and security logs using some

sensor routes and security devices. It extracts parameters such as IP, port from the data set. The collection module stores the raw data into database that can be used in subsequent forensic investigation. The analyzer module uses the frequent item mining algorithm to extract the association rules and detect attacks. The visualization module uses hypergraph to display the frequent item set and attack information. According to the attack information within the hypergraph, forensic investigators can use the interaction module to manually correlate the event and extract original event information from the database for in-depth analysis.

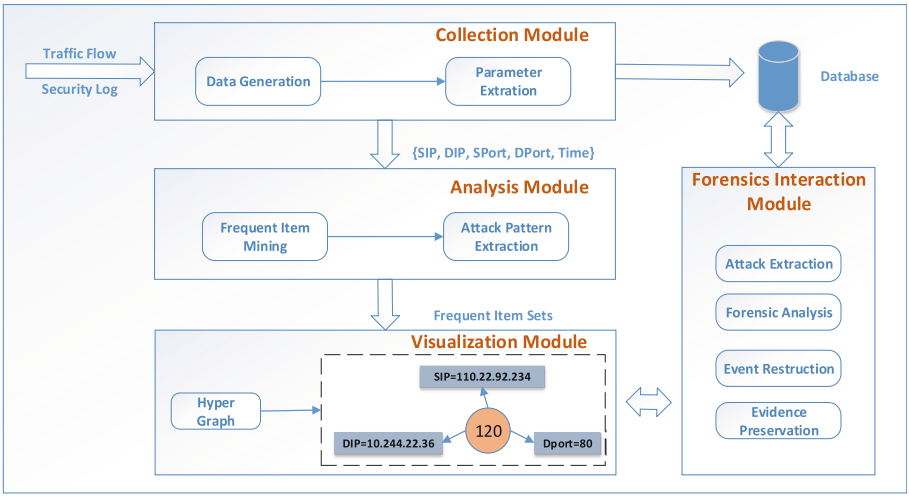

Fig. 3. Proposed visualization system architecture

4 Evaluation

In this section, we extract and display the one-to-many relationship from the flow data and security logs using two different data sets, namely: a local network traffic flow data and the VAST 2012 [27] data set. We implement the visualization system so that it could extract and display the characteristic automatically. We firstly collect the raw data in CSV format, and filter out the other parameters to retain IP, Port and alert type data. Then, we implement the AOI based FIM processing program using the Java language, which extracts the one-to-many relationship from the data set and build the frequent item set for the visualization. After that, we build the hypergraph for these frequent item sets and design a specific hypergraph template for each popular attack described in Sect. 3.4. We also assign values for these templates and display them based on the frequent item set extracted by the preceding step. For each frequent item set that represents the characteristic of an attack, we design a link for the raw data so that the forensic analysts can carry out detailed inspection.

We will now describe the experiments on both data sets. Specifically, the local network traffic flow data set consists of traffic data collected over a total of seven days

in the local network. The VAST data set is from the VAST contest, which has a variety of visualization tasks and data source for researchers to analyze each year. In this paper, we use the VAST 2012 data set to show the effectiveness on firewall and IDS logs. In this particular data set, the context is a virtual international bank's network of nearly 5,000 hosts, and the data set contains log information of IDS and firewalls in its network over a period of time.

4.1 VAST Firewall and IDS Logs

The logs in the VAST data set are in csv format. For the pre-processing, we first transform them into normalized format records, which consist of source IP address, source Port number, destination IP address, destination Port number, alert type and timestamp information. We then filter the firewall logs by manually building a white list according to the BOM network configurations and operation policies. As IDS already filter the logs based on suspected attacks, we label them as anomalies. There are a total of 17,530 records after pre-processing. Then, we extract the necessary parameters from the records.

We extract the one-to-many relationship within the records using the attribute oriented induction based frequent item mining algorithm. The frequent items are presented in Table 1, where each frequent item extracted from the records represent a suspected attack that has a one-to-many relationship.

Table 1. Frequent item mining result for VAST data set.

Frequent item sets	Frequency
SIP = Varied, DIP = 10.32.5.57, type = IRC-Malware Infection SIP = {172.23.231.69-80, 172.23.127.100-120}	696
SIP = 10.32.5.57, DIP = Varied, Sport = 6667, type = IRC-Authorization DIP = {172.23.231.69-80, 172.23.127.100-120}	1464
SIP = Varied, DIP = 10.32.5.57, type = Attempted Information SIP = {172.23.231.69-80, 172.23.127.100-120}	687
SIP = Varied, DIP = 172.23.0.1, type = Misc-activity SIP = {172.23.236.8, 172.23.234.58, 172.23.234.4, 172.231.69}	466

We visualize the frequent item sets using hypergraph – see Fig. 4. We compare the frequent item with the hypergraph template and display them using a hyper edge. The first frequent item and the second meet the Botnet template, and the third frequent item meets the DDoS attack template. From the first and second frequent item sets, we determine that the two main malicious attacks are Botnet behavior and some illegal connections. We also locate a large number of IRC connections from different hosts to IP address 10.32.5.57. These hosts include 172.23.231.69-172.23.231.80, 172.23.127.100-172.23.127.120, which suggest that most of the hosts are infected via the IRC traffic. The second frequent item shows that the host 10.32.5.57 replies to the infected

hosts with IRC authorization messages; thus, indicting a potential Botnet attack. We also found a number of attempted information alerts between 10.32.5.57 and the infected hosts, which suggest a need for further forensic investigation to determine whether data has been exfiltrated.

From the last frequent item, 172.23.0.1 is determined to be the external interface at the firewall, and there have been a number of attempted connections. This suggests the presence of potential DDoS attack or remote services.

Forensic investigators can use the forensic interaction module to fetch and analyze the original data for further investigation.

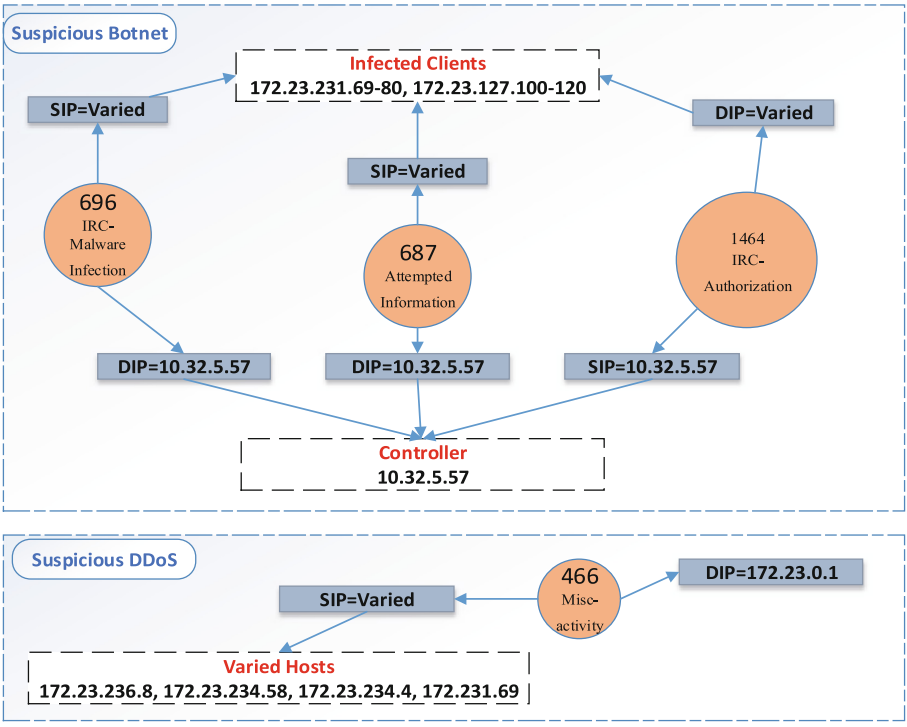

Fig. 4. Visualization of the frequent items using hyper graph

4.2 Local Network Traffic

In order to evaluate the performance of our visualization scheme on the traffic flow data, we collect the flow data from an internal monitoring environment. The environment would generate an alarm in the event of a suspected attack, as well as retaining the flow data of the alert event. We collect 1096 alert data and fetch their flow data to build the records. After pre-processing, the formatted log information is obtained from the data set, which includes 1,096 records. The association rules mining algorithm is used to process the log to extract the attack patterns hidden in the data. We choose the support threshold to be 5%. The frequent item sets are presented in Table 2:

Table 2. Frequent item mining result for local network traffic

Frequent item sets	Frequency
SIP = 193.169.244.73, DIP = 114.255.165.142, Sport = Varied Sport = {80, 443, 5430}	226
SIP = 114.255.165.142, DIP = 193.169.244.73	107
SIP = 221.130.179.36, DIP = 202.106.90.2, SPort = 8888	54

We will now use hypergraph to visualize the above frequent items sets. We use circular nodes and hyper edges with some property rectangles to represent a frequent item with a one-to-many relationship. We identify their quantity characteristics in circular nodes. The rectangle nodes are linked to a circular node indicating the relationship between an associated rule and its attributes. Figure 5 shows the malicious attack patterns extracted by the scheme visualized using hyper graph.

In Fig. 5, SIP denotes the source IP address, and DIP denotes the destination IP address. SPort denotes the source port used by the source host, and DPort denotes the destination port. The number represents the occurrences of a frequent item with a one-to-many relationship. The following information can be found from the visualization results.

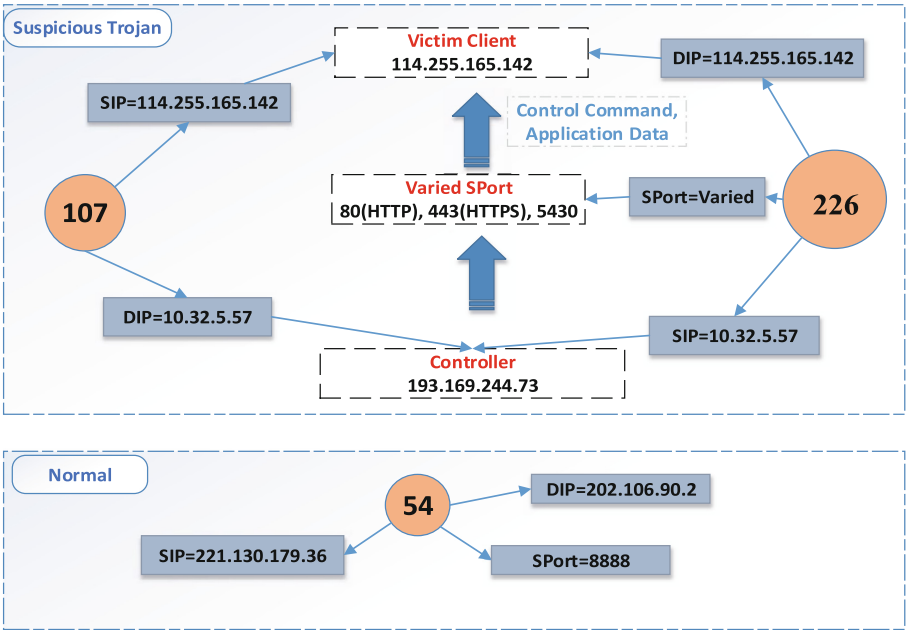

Fig. 5. Visualization of frequent items using hyper graph

(1) Host with IP address 114.255.165.142 connects to host with IP address 193.169.244.73 several times, and the former is an internal host and the latter is an external host.
(2) The external host with IP address 193.169.244.73 mainly connects through ports 80,443 and 5430 with the internal target host 114.255.165.242.
(3) There are a large number of connections between IP addresses 221.130.179.36 and IP 202.106.90.2 via port 8888. Both the source host and destination host are determined to be internal hosts. A large number of connections without a corresponding large data transportation may indicate a specific business need of the network.

From the above findings (1) and (2), forensic investigators can easily determine the malicious connections between internal host 114.255.165.142 and external host 193.169.244.73 and that this is most probably a Trojan attack. The external host continually sends information to the internal infected host via ports 80, 443 and 5430. Both port numbers 80 and 443 are often used for HTTP and HTTPS communication, which could indicate that the controller host sent some commands or some application data to the target host. The event can then be reconstructed based on the attack pattern and original traffic data using the forensic interaction module.

5 Conclusion and Future Work

Network forensics and forensic visualization will be increasingly important in our networked society. Extracting and analyzing anomaly and damage from large scale network data remains a key challenge in network forensics. In order to extract attack patterns hidden in large volume traffic data and security logs and visualize the multi-attribute associations within the attack events, we designed a visualization scheme based on AOI-FIM and hyper graph. Using two real-world data sets, we demonstrated the effectiveness of our proposed scheme in distinguishing attacks and obtaining event-relevant information.

Although frequent item mining based algorithms can be used on big dataset, the processing speed will be affected by significant increases in the data volume. Therefore, future research includes extending our proposed approach to improve the processing speed, and consequently improve the efficiency of network forensics. In addition, research on automated classification and distinguishing methods to extract unknown attacks such as 0-day attacks will be on the agenda.

Acknowledgment. This work is supported by National Natural Science Foundation of China (No. 91646120, 61402124, 61572469, 61402022) and Key Lab of Information Network Security, Ministry of Public Security (No. C17614).

References

1. Zuech, R., Khoshgoftaar, T.M., Wald, R.: Intrusion detection and big heterogeneous data: a survey. J. Big Data **2**(1), 3 (2015)
2. Bhatt, S., Manadhata, P.K., Zomlot, L.: The operational role of security information and event management systems. IEEE Secur. Priv. **12**(5), 35–41 (2014)
3. Cardenas, A.A., Manadhata, P.K., Rajan, S.P.: Big data analytics for security. Secur. Priv. IEEE **11**(6), 74–76 (2013)
4. Tassone, C., Martini, B., Choo, K.K.R.: Forensic visualization: survey and future research directions. In: Contemporary Digital Forensic Investigations of Cloud & Mobile Applications, pp. 163–184 (2017)
5. Tassone, C.F., Martini, B., Choo, K.R.: Visualizing digital forensic datasets: a proof of concept. J. Forensic Sci. (2017)
6. Quick, D., Choo, K.K.R.: Big forensic data management in heterogeneous distributed systems: quick analysis of multimedia forensic data. Softw. Pract. Exp. **47**(8), 1095–1109 (2016)
7. Choi, H., Lee, H., Kim, H.: Fast detection and visualization of network attacks on parallel coordinates. Comput. Secur. **28**(5), 276–288 (2009)
8. Inselberg, A.: Multidimensional detective. In: IEEE Symposium on IEEE Information Visualization, Proceedings, pp. 100–107 (1997)
9. Berthier, R., et al.: Nfsight: NetFlow-based network awareness tool. In: International Conference on Large Installation System Administration USENIX Association, pp. 1–8 (2010)
10. Yin, X., et al.: VisFlowConnect: NetFlow visualizations of link relationships for security situational awareness. ACM Workshop on Visualization and Data Mining for Computer Security, pp. 26–34. ACM (2004)
11. Choi, H., Lee, H.: PCAV: internet attack visualization on parallel coordinates. In: Qing, S., Mao, W., López, J., Wang, G. (eds.) ICICS 2005. LNCS, vol. 3783, pp. 454–466. Springer, Heidelberg (2005). https://doi.org/10.1007/11602897_38
12. Krmíček, V., Čeleda, P., Novotný, J.: NfSen plugin supporting the virtual network monitoring. Virtual networks; monitoring; NetFlow, NfSen (2010)
13. Plonka, D.: FlowScan: a network traffic flow reporting and visualization tool. In: Usenix Conference on System Administration USENIX Association, pp. 305–318 (2000)
14. Taylor, T., et al.: FloVis: flow visualization system. In: Cybersecurity Applications & Technology IEEE Conference for Homeland Security, CATCH 2009, pp. 186–198 (2009)
15. Fischer, F., Mansmann, F., Keim, D.A., Pietzko, S., Waldvogel, M.: Large-scale network monitoring for visual analysis of attacks. In: Goodall, J.R., Conti, G., Ma, K.-L. (eds.) VizSec 2008. LNCS, vol. 5210, pp. 111–118. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85933-8_11
16. Leinen, S.: Fluxoscope a system for flow-based accounting (2000)
17. Promrit, N., Mingkhwan, A.: Traffic flow classification and visualization for network forensic analysis. In: IEEE International Confrence on Advanced Information Networking and Applications. IEEE, pp. 358–364 (2015)
18. Yang, W., Wang, G., Bhuiyan, M.Z.A., Choo, K.-K.R.: Hypergraph partitioning for social networks based on information entropy modularity. J. Netw. Comput. Appl. **86**, 59–71 (2017)
19. Glatz, E., et al.: Visualizing big network traffic data using frequent pattern mining and hypergraphs. Computing **96**(1), 27–38 (2014)
20. Hirsch, C., et al.: Traffic flow densities in large transport networks (2016)

21. Borgelt, C.: Frequent item set mining. Wiley Interdisc. Rev. Data Min. Knowl. Discov. **2**(6), 437–456 (2012)
22. Cai, Y., Cercone, N., Han, J.: Attribute-oriented induction in relational databases. Knowl. Discovery Databases **15**(7), 1328–1337 (1989)
23. Han, J., Cai, Y., Cercone, N.: Knowledge discovery in databases: an attribute-oriented approach. In: International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc. 547–559 (1992)
24. Warnars, S.: Mining frequent pattern with attribute oriented induction high level emerging pattern (AOI-HEP). In: International Conference on Information and Communication Technology IEEE, pp. 149–154 (2014)
25. Guzzo, A., Pugliese, A., Rullo, A., Saccà, D.: Intrusion detection with hypergraph-based attack models. In: Croitoru, M., Rudolph, S., Woltran, S., Gonzales, C. (eds.) GKR 2013. LNCS (LNAI), vol. 8323, pp. 58–73. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-04534-4_5
26. Zhou, D., Huang, J.: Learning with hypergraphs: clustering, classification, and embedding. In: International Conference on Neural Information Processing Systems. MIT Press, pp. 1601–1608 (2006)
27. Cook, K., et al.: VAST challenge 2012: visual analytics for big data. In: 2012 IEEE Conference on Visual Analytics Science and Technology (VAST), 251–255. IEEE (2012)