Event Pattern Discovery on IDS Traces of Cloud Services

Shin-Ying Huang
Research Center for IT Innovation
Academia Sinica
Taipei 11529, Taiwan.
smichelle19@citi.sinica.edu.tw

Yennun Huang
Research Center for IT Innovation
Academia Sinica
Taipei 11529, Taiwan.
yennunhuang@citi.sinica.edu.tw

Neeraj Suri
Dept. of CS, TU Darmstadt
Darmstadt, Germany.
suri@cs.tu-darmstadt.de

***Abstract*—The value of Intrusion Detection System (IDS) traces is based on being able to meaningfully parse the complex data patterns appearing therein as based on the pre-defined intrusion 'detection' rule sets. As IDS traces monitor large groups of servers, large amounts of network data and also spanning a variety of patterns, efficient analytical approaches are needed to address this big heterogeneous data analysis problem. We believe that using unsupervised learning methods can help to classify data that allows analysts to find out meaningful insights and extract the value of the collected data more precisely and efficiently. This study demonstrates how the technique of growing hierarchical self-organizing maps (GHSOM) can be utilized to facilitate efficient event data analysis. For the collected IDS traces, GHSOM is used to cluster data and reveal the geometric distances between each cluster in a topological space such that the attack signatures for each cluster can be easily identified. The experimental results from a real-world IDS traces show that our proposed approach can efficiently discover several critical attack patterns and significantly reduce the size of IDS trace log which needs to be further analyzed. The proposed approach can help internet security administrators/analysts to conduct network forensics analysis, discover suspicious attack sources, and set up recovery processes to prevent previously unknown security threats such as zero-day attacks.**

Keywords—forensic analysis, cloud services, intrusion detection system, internet security, growing hierarchical self-organizing map

I. INTRODUCTION

As networks and computing nodes are constantly being probed and attacked, network security products such as intrusion detection systems (IDS) are used to identify malicious activities and block the suspicious packets. Often, the IDS traces are stored on a backup resource and are not utilized unless a serious attack actually happens. This can keep the existing backdoors open where attackers can target a specific victim, and also allow the attackers to deploy advanced network attacks such as distributed denial of service (DDoS) attacks and advanced persistent threats (APT).

The Problem:

Intrusion detection system faces problems such as massive network traffic volumes, highly imbalanced data distribution, the difficulty to realize decision boundaries between normal and abnormal behavior, and a requirement for continuous adaptation to a constantly changing environment [21]. Current IDSs techniques have several limitations such as (1) IDSs provide unmanageable amount of real or false alarms overwhelming the security administrators, (2) it is very difficult to determine how well an IDS is set up, and (3) the degree of work and time required [6][20][22]. Therefore, the need is for IDSs to provide automated detection of malicious traffic behavior via techniques such as pattern recognition and anomaly detection.

There are several difficulties from the nature of the IDS trace data which makes it hard to be analyzed: (1) IDS traces are a series of log data which is often unstructured, and typically there is no relation information in it unless further analysis is done, (2) companies tend to ignore the information hidden in the IDS traces and overly depend on signature-based IDS, and (3) it is hard for the current IDSs to detect potential threats or formalize new attack patterns. Therefore, the key problem is the typical inability to be able to correlate the events recorded in the IDS traces because the attack behaviors have been concatenated into a set of events. A single event with a fragmentary signature cannot reveal too much information at first glance. Network traffic data tends to be data-rich but information-poor. Hence, developing techniques to handle the avalanche of IDS traces data into meaningful information and revealing previously unknown patterns in IDS traces forms the motivation for our research.

Practically, the network security products rely heavily on known signatures to detect malicious traffic, but still lack intelligence to digest what they have identified. Thus, an effective and efficient network forensic analysis mechanism is a key need to identify the leakage paths along with identifying the intruders. Furthermore, for identifying unseen attacks, the analysis of periodical network traffic data is essential to develop both a knowledge-based IDS and a behavior-based IDS.

The state of the art IDS has several limitations: (1) it is hard to identify novel attacks or minor variations of known attack patterns, (2) the tool generates alerts, which must be reviewed by a security analyst to ascertain the triggers behind the alerts , (3) it is having not possible to send error pages to clients as it does not work on HTTP(s), and (4) it is not possible just to write rules on outbound traffic because rules can be written either for both (inbound/outbound) or for inbound traffic.

Our Contributions:

To the best of our knowledge, little research has been focused on investigating the static event records because usually they are left as backup logs with little attention paid to this unstructured data to conduct network forensics. For the purpose of analyzing network traffic events with heterogeneous patterns and noises, unsupervised learning methods can help to explore groups of attack activities by correlating their input features horizontally (by time) and hierarchically (by features). This composite handling of time and feature characteristics in IDS traces is conspicuously missing in current approaches and defines our uniqueness.

Studies have shown that the Growing hierarchical self-organizing map (GHSOM) is useful for understanding network traffic data by visualizing the anomalous groups [12] [21], but there is no study that focuses specifically on event analysis and anomalous pattern discovery. Therefore, this paper proposes an approach based on GHSOM specifically for event pattern discovery that makes use of IDS traces in order to explore malicious activities which are either undergoing or have potential to become serous attack events in the future.

The paper's contribution is its focus on the event pattern discovery by using GHSOM to visualize the clustered groups of IDS traces in order to better identify and correlate various attack patterns. We use a set of real-world IDS traces to demonstrate how to analyze such data and we also apply data mining tools to discover group features, especially the missing event correlations.

Paper Outline:

On this background, we first present related work and a basic background on GHSOM in Section II and III before detailing our trace analysis approach in Section IV. Our approach is applied to actual system IDS traces and we detail both the experimental/validation setup along with the obtained results in Section V. Our summary discussions appear in Section VI and VII.

II. RELATED WORD

Multiple methodologies exist for facilitating network intrusion detection. However, they only address segments of the problem issues. We provide a brief discourse on them for the purpose of building and extending upon this body of knowledge.

Techniques of wavelets analysis provide a description for concurrent time and frequency, it can separate the time-localized anomalous signals from the noise signals by analyzing each spectral window's energy, the anomalies can be determined. Studies such as Kim et al. [8] proposed a detection technique based on creating a stable baseline profile to monitor the deviations in the traffic. An analysis was conducted to check the stability of the traffic with regards to different parameters. Significant differences in traffic patterns were found between different sites. A baseline profile that was based on different attributes was proposed for detection. Thapngam et al. [18] proposed a comparable detection methods based on the Pearson's correlation coefficient. Their methods can extract repeatable features from the packet arrivals in the DDoS traffics but not in flash crowd traffics. Choras et al. [3] proposed a novel framework for network security based on the correlation approach as well as new signal-based algorithm for intrusion detection on the basis of the matching pursuit (MP) algorithm, which is a known signal processing technique used for instance in audio compression, image and video compression [11].

Supervised learning methods are usually used in intrusion detection which is often regarded as a classification problem. For example, Tajbakhsh et al. [17] proposed a classification algorithm using fuzzy association rules for building classifiers. Particularly, the fuzzy association rule sets were exploited as descriptive models of different classes. The compatibility of any new sample (which is to be classified) with different class rule sets was assessed by the use of some matching measures and the class corresponding to the best matched rule set was declared as the label of the sample. Salama et al. [13] introduced a hybrid scheme that combined the advantages of deep belief network (DBN) and support vector machine (SVM). An application of intrusion detection imaging was been chosen and a hybridization scheme was applied, where DBN was used as a feature reduction method and SVM as a classifier. Sheikhan et al. [14] proposed a 3-layer recurrent neural network (RNN) architecture with categorized features as inputs and attack types as outputs of RNN. Experimental results showed that the reduced-size neural classifier improved classification rates.

The technique most useful for our needs (of diverse patterns, automation and accuracy) is likely unsupervised learning that groups data based upon their similarities without knowing the class-labeled information in the training phase. Therefore, it can be applied to anomaly detection and pattern discovery [21]. For example, Tjhai et al. [19] developed a two stage classification system using a self-organizing map (SOM) [9] neural network and K-means algorithm [10] to correlate the related alerts and to further classify the alerts into classes of true and false alarms. Stevanovic et al. [16] examined the use of two unsupervised neural network learning algorithms for web-log analysis, the SOM and Modified Adaptive Resonance Theory 2 (Modified ART2), to obtain a better insight into the types and distribution of visitors to a public web-site based on their browsing behaviors. Palomo et al. [12] used growing hierarchical self-organizing map (GHSOM) [5] to cluster network traffic data and the results confirmed that the GHSOM is very useful for a better understanding of network traffic data, making it easier to search for evidences of attacks or anomalous behaviors in a network environment. While the GHSOM approach offers a core basis, its ability to handle IDS traces is lacking. Addressing this gap forms the basis of our proposed approach.

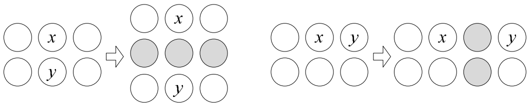
III. BACKGROUND & USAGE PERSPECTIVES ON GROWING HIERARCHICAL SELF-ORGANIZING MAPS

To efficiently deal with massive IDS traces in constantly changing environments, we consider the use of GHSOM as it is a scalable unsupervised learning method where the clustering process can help to divide these traces into smaller groups with preferred group size, horizontal and hierarchical structure; besides, it can provide topological location of the subgroups from the GHSOM so that their similarity can be explained and correlated. These properties are what we require for the purpose of event discovery of IDS traces. We first provide a brief synopsis on GHSOM operations before developing our techniques on it.

The classical GHSOM training process contains the following four phases [5]:

1. Initialize layer 0: Layer 0 includes a single node, the weight vector of which is initialized as the expected value of all input data. The mean quantization error of layer 0 (MQE_0) is calculated next. The MQE of a node denotes the mean quantization error that sums the deviation between the weight vector of the node and all input data mapped to the node.
2. Train each individual map: Under the competitive learning principle, only the winner and its neighboring nodes qualify for an adjustment of their weight vectors. The competition and training processes are repeated until the learning rate decreases to a certain value.
3. Grow each individual map horizontally: Each individual map grows until the mean value of the MQE for all nodes on the map, i.e., avg(MQE), is smaller than the MQE of the parent node MQE_p multiplied by τ_1, as in (1). If the stop criterion is not satisfied, we find the *error node* that owns the largest MQE and insert one row or column of new nodes between the error node and its dissimilar neighbor, as shown in Fig. 1. The notation *x* indicates the error node and *y* indicates the dissimilar neighbor.

$$\text{avg}(MQE) < \tau_1 \times MQE_p \qquad (1)$$



(a) insert a row (b) insert a column

Fig. 1. Horizontal growth of GHSOM.

4. Expand or terminate the hierarchical structure: The node with an MQE_i greater than $\tau_2 \times MQE_0$ will be used to develop the next layer, as in (2).

$$MQE_i < \tau_2 \times MQE_0 \qquad (2)$$

The GHSOM structure is shown in Fig. 2, each layer contains a number of maps (SOMs). Each map is grown individually when doing the hierarchical growth.

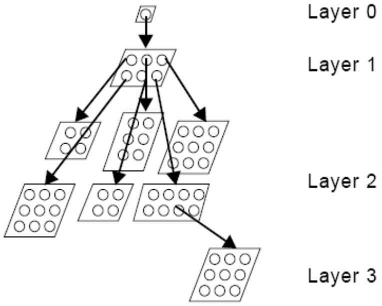


Fig. 2. The GHSOM structure.

The unsupervised methods such as SOM and its extension are suitable for ID tasks in that normal behavior is densely populated around one or two centers, while abnormal behavior and intrusions appear in sparse regions of the pattern space outside of normal clusters. Other classification algorithms, such as feed forward neural networks, were then trained on the clustering output [21].

Practically, GHSOM can be integrated into the higher level IDS because IDS can be layered in a hierarchy where the alert output of the lower stage is processed by a second IDS. The higher IDS is often used for correlation. It can also generate statistics, group alerts and detect outliers to provide a more succinct overview of the situation. This is especially useful when a large number of alerts are produced [4].

Comparing GHSOM with other trajectory-based clustering/outlier detection methods in spatial mining domains, Wu and Banzhaf [21] reviewed the use of computational intelligence in intrusion detection systems and observed that the SOM algorithms are suitable for intrusion detection as normal behavior is densely populated around one or two centers, while abnormal behavior and intrusions appear in sparse regions of the pattern space outside of the normal clusters.

In other words, GHSOM functions as a data pre-processor to cluster input data. For those IDS traces in the same clustered subgroups, we can apply other data mining techniques to help to analyze the correlation between the IDS traces. We believe that analyzing smaller groups of data with a certain similarity can come up with more meaningful mining results compared with directly analyzing a bunch of raw data. Data mining methods such as association mining for link analysis, and frequent episodes for sequence analysis can be used to derive header features for detecting some attacks. Feature selection can then be applied to the candidate set of features.

IV. PROPOSED APPROACH

On this basic background on GHSOM, we now present our anomaly detection approach where the individual steps are illustrated in Fig. 3. In the subsequent sections we progressively detail these to develop the validation. Our proposed approach primarily starts from the data clustering stage in Fig. 3. We apply GHSOM over the data clustering stage. Then, a series of visualization, feature observation and event pattern discovery are done based on the clusters of IDS traces. Also, the association analysis on event correlation and web graph analysis for IP correlation are applied to discover any intrigued event pattern.

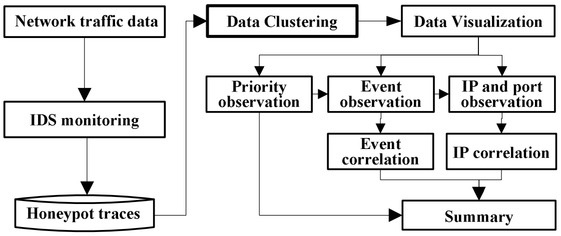


Fig. 3. Architecture of the anomaly detection approach.

When any network traffic triggers the IDS rule sets, it will be blocked and the flow information as well as the triggered event will be stored in the cloud. The IDS traces are mixed with all sorts of monitored IPs, so as the sources IPs. To help building up a focus for the internet security experts, we use GHSOM to cluster the data according to their inherent nature. Based on the visualized clusters, features such as priority, source IP, source port, destination IP, destination port, and the triggered event will be analyzed. Specifically, we will look at the map (means the clusters belong to same branch and locate in the same layer) with most frequent priority one samples or the map located in deeper layer to observe their features of IP and port. The frequent events are listed in descending order based on the amount of samples in which various event patterns are discovered and are left open for explanation. Besides, the information of IP and port can tell a basic profile of the attack path, for example, source IP *x* use port number *m* to connect to the destination IP *y* through port *n*. In order to find out the correlation of source IP and destination IP, for each investigated cluster, we apply the web graph analysis technique to obtain the strong links. Furthermore, we apply association analysis for each investigated cluster where a set of correlated events will be marked as an important attack feature.

A. *GHSOM setting*

The GHSOM development is primarily dominated by the breadth (τ_1) and depth (τ_2) parameters [5]. To achieve the goal of obtaining the multi-layer hierarchy features and preventing over-clustering, we predefine the following selection criteria:

1) There is more than one SOM layer in the GHSOM.
2) Samples of each node should not be overly clustered.
3) Each leaf node should contain at least one sample.

The one satisfied with the selection criteria and has the lowest MQE and fewer numbers of clusters are chosen. Following the GHSOM algorithm stated in Section II, the samples with similar patterns are expected to be grouped together and clusters are plotted in topological space.

B. *Feature observation*

We tend to target on the map with the most frequent *priority* equal to 1, or the map located in deeper hierarchical layer. The map with the most frequent *priority* equal to 1 means that lots of critical events have happened and these events are similar in a certain level, such concentration of events are worth of further exploration in terms of statistics signatures and behavior signatures. For the map with the most frequent *priority* 1, the other traces which are not *priority* 1 but belong to the same map can still represent a certain degree of similarity, so that any sign for future attacks can be inferred. The map located in deeper hierarchical layer is also worth of analysis because clusters in such map are more diverse than other clusters located in other branches. We believe that complex events often hide more valuable information about attack patterns.

C. *IP correlation*

We use web graph analysis to do the IP correlation. Web graph, plot nodes and histograms give users a visual representation of the strength of connection between the variables in the data. Graph is used to spot characteristics and patterns at a glance [15]. In a web graph, web nodes are used to show the strength of relationships between values of two or more symbolic fields. Connections are displayed in a graph with various types of lines to indicate connections of increasing strength. Directed web graphs show connections only from one or more *From* fields to a single *To* field. The connections are unidirectional in the sense that they are one-way connections. Here we choose two fields: *From* source IP *To* destination IP to explore the relationship between them and find out the strong connections.

The size of links in the output graph are varies continuously, which display a range of link sizes reflecting the variation in connection strengths based on actual data values. The strong connections are shown with a heavier line, which indicates that the two IPs are strongly related and should be further explored.

D. *Event correlation*

This study applies the Continuous Association Rule Mining Algorithm (CARMA) [7] to do the IP correlation. CARMA generates the *itemsets* in the first scan and finishes counting all the *itemsets* in the second scan. Specifically, CARMA generates candidate *itemsets* on the fly from every transaction. While reading a transaction, it increments the supports of all candidate *itemsets* contained in the transaction. A new candidate *itemset* contained of a transaction are generated if all of its subsets are relatively frequent with respect to the number of those processed transactions.

Note that the second scan of all the transactions is not needed, whenever the shrinking, deterministic intervals on the produced rules support and confidence suffice. Thus, CARMA can be used to continuously produce association rules from a list read from a network. Besides, CARMA can extract a set of rules without requiring user to specify the *In* (predictor) or *Out* (target) fields. This means that the generated rules can be used for a wider variety of applications. Compared to Apriori [1], CARMA offers build settings for rule support (support for both antecedent and consequent) rather than antecedent support, and rules with multiple consequents is allowed [15]. While not being faster in general, CARMA outperforms Apriori and Dynamic Itemset Counting (DIC) [2] on low support thresholds and is up to 60 times more memory efficient [7].

V. EXPERIMENTS

We now detail and demonstrate our proposed approach for IDS event analysis.

A. Dataset Description

The IDS dataset is collected from the Security Operation Center (G-SOC) of the Taiwanese government, for the entire year of 2012. This dataset includes the IDS triggered alerts from 61 government cloud services which contain 284412 types of events. The G-SOC uses Hadoop as the MapReduce implementation to filter data from different resources and organize the data. The pre-processed data with unified features will be used as our analyzed data sources. These types of events can be categorized into 39 classes, of which 33 classes of attack is defined by Snort IDS, the other 6 classes are custom by G-SOC (including: blacklist, high threat malware behavior and so on). The input features of an IDS trace in our research include the time*stamp*, *source IP address*, *destination IP address*, *priority*, *protocol*, *source port*, *destination port*, *event id*, and *sensor id*. The *priority* ranges from 1 to 3. Table I shows the categories of event.

TABLE I. SUMMARY COMPARISON OF THE ATTACK BEHAVIOR

No	Event	No	Event
1	attempted-dos	21	attempted-user
2	anti-botnet	22	policyviolation
3	malware-ip	23	unsuccessful-user
4	notsuspicious	24	default-login-attempt
5	string-detect	25	spamhaus-drop
6	trojan-activity	26	successfuluser
7	network-scan	27	attempted-recon
8	shellcode-detect	28	unknown
9	suspiciousbackdoor	29	web-application-attack
10	misc-activity	30	successful-reconlimited
11	protocol-command-decode	31	suspicious-filename-detect
12	web-application-activity	32	bot-c&c
13	highthreat-malware	33	attempted-admin
14	suspicious-host-name	34	le
15	successful-admin	35	malicious-ip
16	botnet-malware	36	rpcportmap-decode
17	suspiciousip	37	denial-of-service
18	misc-attack	38	suspicious-login
19	bad-unknown	39	system-call-detect
20	monitor-botnet		

B. Data Clustering

The GHSOM parameter τ_1 is adjusted from 0.5 to 0.9 per 0.1 scales, and the parameter τ_2 is adjusted from 0.01 to 0.05 per 0.01 scales. The results show that the parameterτ_1 = 0.7 and τ_2 = 0.03 meet the selection criteria stated in section III, so we pick this setting and visualize a series of clustering results.

1) Priority observation

The result of GHSOM contains 16 clusters. As shown in Fig. 4, the number in cycle means cluster number, the number in diamond means map number, the number in the left of the parenthesis means *priority*, the number within the parenthesis means the amount of samples for a specific *priority*, and the number within the square bracket means the sample size of a specific cluster.

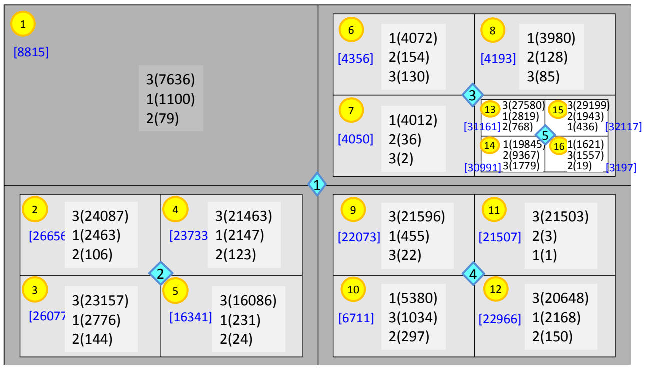


Fig. 4. Priority observation

According to Fig. 4, the cluster 6, 7, 8, 10, 14 and 16 whose most frequent *priority* is one, while other clusters' most frequent *priority* is three. In other words, map 3 and its sub map 5 have more clusters with *priority* one feature, so the clusters belong to map 3 and map 5 can be further analyzed.

2) Event observation

Because the map 3 contains the most number of priority one events and the branch of map 3 has more complex clustering structure, we take map 3 including its sub map – map 5 as an example to further analyze their attack patterns. Fig. 5 shows the event map of the map 3 and 5 of GHSOM, where the *event IDs* are sorted by sample size.

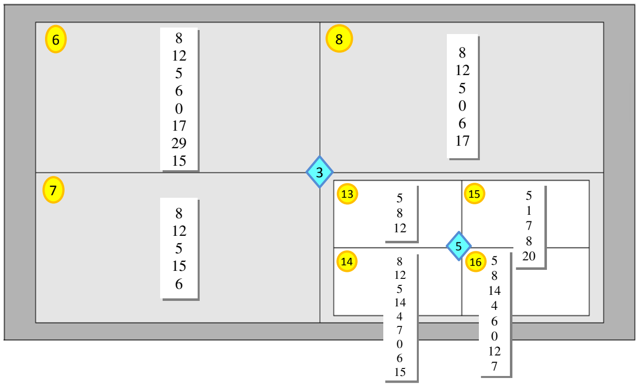


Fig. 5. The event map of the map 3 and map 5

The discussions about the event patterns in map 3 and map 5 are described as follows: First, take a look at cluster 6, 7 and 8, the first three events are event 8 (A Windows cmd.exe banner is detected in a TCP session), event 12 (Certain non-RFC chars are used in a request URI) and event 5 (Violate a corporate security policy). So, the common activities would be:

use of the Windows cmd.exe banner, use of suspicious URI format, and violate a corporate security policy. Note that cluster 14 in map 5 also has the same top three most frequent event categories. Cluster 13 also has the same top three most frequent event categories but the order is slightly different (event 5, 8, 12). Somehow the GHSOM can catch such event similarities and divide them appropriately in different levels, which is able to preserve their relationships by comparing their relative geographic distances.

Take a look at cluster 13 and 15. As shown in Fig. 6, although their *source IP* and *destination IP* are the same, but their event reports are much difference. Cluster 13 tends to use Windows cmd.exe banner in a TCP session (event 8). Cluster 15 tends to use VOIP-SIP register flood to attack the target server (event 1). The attacker from *source IP* 209.190.31.58 adopted two approaches to attack the same victim.

3) IP and port observation

Fig. 6 shows the information of most frequent *source IP*, *destination IP*, *source port* and *destination port* of map 3 including its sub set map 5 for IP and port observation.

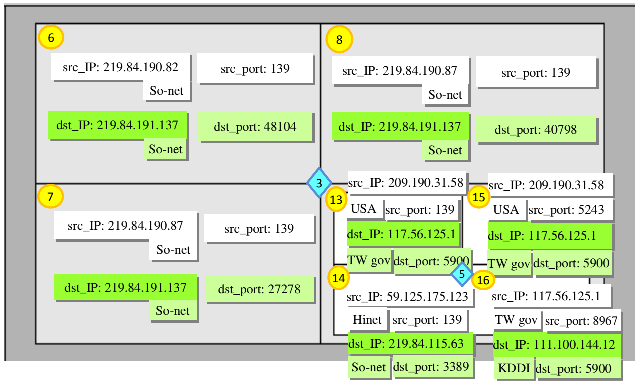


Fig. 6. IP and port observation

As shown in Fig. 6, the clusters 6, 7 and 8 have very similar features while the clusters 13, 14, 15 and 16 have clear differences and are more diverse compared with the clusters 6, 7 and 8. The clustering structure is reasonable which confirms that the GHSOM is good at clustering samples with high dimensionality and creating a hierarchical clustering adaptively.

Take a close look at cluster 6, 7 and 8, we found both three of them have very similar *source IP* and use the same *source port*. Moreover, these three clusters are mostly attacked by the same *destination IP* under different *destination port*. Therefore, it is worth noting that the attackers may come from one organization because they have slightly different *source IPs* (219.84.190.82, 219.84.190.87), but they all target on the same *destination IP* (219.84.191.137). The major differences of these three clusters would be the *destination port*. We suspect that the victims in these three clusters were intruded through different probing activities in order to search for any vulnerability.

The cluster 13 and 15 are almost the same except they use different *source port* (139 and 5243); in addition, the *source IP* comes from USA and the *destination IP* is a Taiwan government agency. Therefore, cluster 13 and 15 have high correlation. The *source IP* 209.190.31.58 is highly dangerous and is an obvious clue for further network forensic analysis.

C. *Event correlation*

In the stage of event correlation, we apply CARMA to generate association rules of events, and to mark out the correlation between events as an important feature for each cluster. The minimum rule size is set to be three, and we pick up the rules with confidence more than 90%.

In Table II, there are four rules in map 2 with support value larger than 10%, which are (12→5, 8→5, 0→5, [8 and 12]→5). Given by the results of event observation, the most frequent events of map 2 are event 5, 8, 0, 14, 12, the observed correlations can help to explain the causes and effects of alerts based on time factors. For example, before the attackers who violate a corporate security policy (event 5), they tend to use illegal URI (event 12), use a Windows cmd.exe banner in a TCP session (event 8), or use illegal FTP command (event 0).

TABLE II. ASSOCIATION RULES IN MAP 2

Consequent	Antecedent	Support %	Confidence %
5	12	43.48	100.00
5	8	15.94	100.00
5	0	10.14	100.00
5	8, 12	10.14	100.00

D. *IP correlation*

Take the map 5 as an example to further investigate its network pattern in order to do IP traceback as a part of network forensic process. Fig. 7 shows the result of IP correlation after doing the web graph analysis. The connection is set from *source IP* to *destination IP*. The square node means *source IP*, and the circle node means *destination IP*.

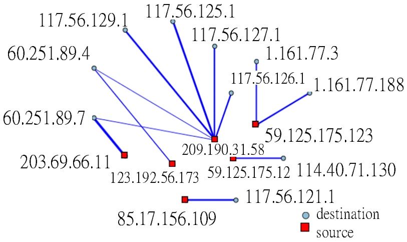


Fig. 7. The strong links in map 5

The strong connections are shown with a heavier line, which indicates that a group of *source IPs* and *destination IPs* are strongly related. Fig. 7 shows several connecting patterns. For example, there is a one-to-many pattern which is started from 209.190.31.58 and is connected to multiple destinations. Another one-to-many connection which is started from 59.125.175.123 to two *destination IPs*. There are two connections with many-to-one pattern. *Destination IP* 60.251.89.7 has two mainly sources, which are 203.69.66.11 and 209.190.31.58. *Destination IP* 60.251.89.4 has two mainly sources, which are 209.190.31.58 and 123.192.56.173.

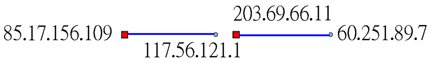
E. *Summary*

To summarize the obtained results and let the investigators understand where to focus and what the interesting finding is, we list the key features of each focused cluster in a bullet-point form. The clusters located in deeper layers of GHSOM are often worth of investigation because of their variety, we expect such variety has a positive impact on risk, such as a new form of attacks or other potential malicious activities undergoing. Table III shows the summarized results of map 5.

TABLE III. SUMMARIZED ATTACK PATTERN

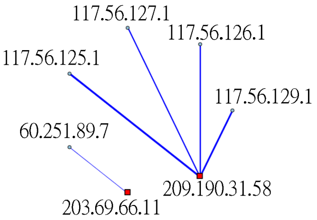
Cluster 13 *Profiling: Involved hacking activities through port 139*

- Most frequent event: 5, 8, 12. ●Most frequent *priority*: 3
- Event correlation: 8→5
- Most frequent *source IP* (port): 209.190.31.58 (139), a risky IP
- Most frequent *destination IP* (port): 117.56.125.1 (5900)
- Web graph:



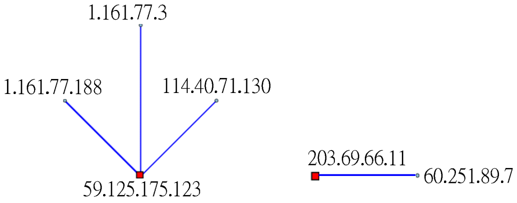
Cluster 14 *Profiling: Involved corporate privacy violations though inbound Teredo traffic*

- Most frequent event: 8, 12, 5. ●Most frequent *priority*: 1
- Event correlation: 8→5, 20→5
- Most frequent *source IP* (port): 59.125.175.123 (139)
- Most frequent *destination IP* (port): 219.84.115.63 (3389)
- Web graph:



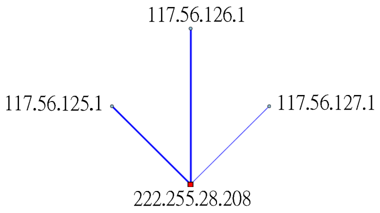
Cluster 15 *Profiling: Involved hacking activities through port 5243*

- Most frequent event: 5, 1, 7. ●Most frequent *priority*: 3
- Event correlation: 0→5→8
- Most frequent *source IP* (port): 209.190.31.58 (5243), a risky IP
- Most frequent *destination IP* (port): 117.56.125.1 (5900)
- Web graph:



Cluster 16 *Profiling: Involved internal corporate privacy violations*

- Most frequent event: 5, 8, 14. ●Most frequent *priority*: 1
- Event correlation: 20→5
- Most frequent source IP (port): 117.56.125.1 (8967)
- Most frequent *destination IP* (port): 111.100.144.12 (5900)
- Web graph:



F. *Comparison*

We compare the GHSOM clustering results with SOM because both of these methods can preserve the topology of nodes (i.e., subgroups). The mapsize is set to [8×8] in order to make it similar to the mapsize of second layer of GHSOM (see Fig. 4). Table IV shows the clustering results of SOM with priority information. The first column is node number. The second column is the features of each node including priority, event, and amount of samples.

TABLE IV. SOM WITH PRIORITY INFORMATION

No	feature	no	feature	no	feature	no	feature	no	feature	no	feature	no	feature	no	feature
1	(3, 29) 5264	9	(3, 17) 2906	17	(3 ,12) 3297	25	(3, 12) 1145	33	(3, 8) 18019	41	(3, 15) 22223	49	(3, 15) 13623	57	(3, 15) 15916
2	(3, 12) 855	10	(1, 15) 173	18	(1 ,8) 24	26	(3, 8) 766	34	(3, 8) 4969	42	(3, 8) 86	50	(3, 20) 1640	58	(3, 15) 14507
3	(1, 8) 34	11	(0, 0) 0	19	(0 ,0) 0	27	(3, 5) 735	35	(3, 7) 2213	43	(3, 14) 1956	51	(3, 14) 1525	59	(3, 14) 1878
4	(0, 0) 0	12	(2, 12) 1	20	(3 ,12), 8649	28	(3, 8) 7454	36	(3, 14) 2159	44	(3, 14) 1226	52	(3, 14) 316	60	(3, 14) 436
5	(2, 12) 8	13	(3, 12) 7914	21	(3 ,12) 8035	29	(3, 8) 5610	37	(3, 5) 6038	45	(3, 7) 5410	53	(3, 8) 35	61	(3, 17) 3383
6	(3, 12) 4898	14	(3, 8) 2906	22	(3 ,5) 5115	30	(3, 5) 1753	38	(3, 5) 1889	46	(3, 7) 19	54	(3, 17) 1812	62	(3, 17) 3214
7	(3, 5) 4502	15	(3, 5) 1583	23	(3 ,12), 3390	31	(3, 12) 6829	39	(3, 12) 6452	47	(3, 7) 1439	55	(3, 7) 9	63	(3, 17) 2128
8	(3, 7) 4795	16	(3, 12) 4094	24	(3 ,12) 6774	32	(3, 12) 9362	40	(3, 12) 32249	48	(3, 17) 2308	56	(3, 14) 3716	64	(3, 29) 2746

*Parenthesis: (priority, event); second raw: amount of samples.

According to Table IV, the node (i.e., cluster) 3, 10 and 18 have most frequent priority is one. We find that all samples are priority one and their event number are eight. The sample sizes are relatively small and pure which makes the following event correlation meaningless. As we look back to Fig. 4, the GHSOM can generate groups of nodes with hierarchical relationship with a certain heterogeneity (e.g., map 3 and map 5), which can help to explore meaningful features between neighboring nodes by data mining techniques.

VI. DISCUSSION

Based on the experimental results from the IDS traces as alerts triggered from the SNORT IDS rules, GHSOM can generate groups of nodes with hierarchical relationship with a certain heterogeneity which can help to extract meaningful features between the neighboring clusters. This visualization process is essential for analysis as GHSOM can help identify risky groups and reduce the size of IDS traces necessitating deeper analysis by up to 90.83% (see map 3 in Fig. 4).

The IDS alerts should be investigated but current IDSs are mostly rule-based systems and cannot provide such functions for alerts correlation and visualization. Our proposed approach leverages the advantage of unsupervised learning method GHSOM to organize these alerts into different subgroups, reveal their distinctive features and show where the vulnerability is. To further visualize the ID correlations, we first integrate GHSOM with the association rule method (CARMA) and Web graph tools to deeply analyze the embedded attack patterns within the investigated clusters.

VII. Conclusion

This paper proposes an approach for event pattern discovery by using GHSOM and other data mining tools to explore the IDS traces and turn numerous data into meaningful patterns. The results of event pattern discovery can help to do network forensics analysis through tracing back to the source of an attack and finding out the relationships among events as a set of attack patterns, which then can be used to create a prevention mechanism with dynamic filtering rules for IDS. Finally, the proposed approach is designed to be integrated with the current internet security deployments which all have the same purpose— to minimize the damage of the undergoing attack and prevent future attacks based on the knowledge learned from the analysis of the historical data.

The main contribution of this paper is that our proposed method is capable of facilitating event pattern discovery based on a very big set of IDS traces (collected from 61 Taiwan government cloud services in 2012) which is helpful in discovering the attack patterns within different subgroups, and significantly reducing the effort of security analysts to analyze such massive IDS traces. The relationships between each cluster are visualized in the typological space through lenses of dimension, where similar events are grouped together based on their nature so that more information can be achieved compared to empirical approaches. For example, (1) merely look at the events with *priority* one feature is not enough to find out the scenario of attack, (2) behavior-based events are important because the time factor can be considered in the behavior-based events, (3) relating the patterns among neighboring clusters is a key element to add more information to the knowledge pool.

Future works are described as follows: (1) to study the correlations among the features such as *port*, *protocol*, and *event* in the application level; (2) to include more input features such as frequency and duration, and refer to multiple data resources; (3) to study links in web graph for monitoring potential attacks; (4) to apply the proposed approach to other security issues, such as APT, botnet, malware, and social network; (5) to develop an incremental learning mechanism that helps to detect zero-day malware.

ACKNOWLEDGMENTS

We acknowledge the support from the Ministry of Science and Technology (No. NSC101-2221-E001-009-M) and TUD LOEWE CASED.

REFERENCES

[1] Agrawal, R. and Srikant, R. 1994. Fast Algorithms for Mining Association Rules in Large Databases. Proc. VLDB, 487-499.

[2] Brin, S., Motwani, R., Ullman, J. D., and Tsur, S. 1997. Dynamic Itemset Counting and Implication Rules for Market Basket Data. Proc. of ACM SIGMOD, 255-264.

[3] Choras M., Saganowski, L., Renk, R., and Hołubowicz, W. 2012. Statistical and signal-based network traffic recognition for anomaly detection. Expert Systems 29, 3, 232-245.

[4] Davis, J. J. and Clark, A. J. 2011. Data preprocessing for anomaly based network intrusion detection: A review. Computers & Security 30, 6-7, 353-375.

[5] Dittenbach, M., Rauber, A., and Merkl, D. 2002. Uncovering hierarchical structure in data using the growing hierarchical self-organizing map. Neurocomputing 48, 1-4, 199-216.

[6] Elshoush, H. T. and Osman, I. M. 2011. Alert correlation in collaborative intelligent intrusion detection systems—A survey. Applied Soft Computing 11, 7, 4349-4365.

[7] Hidber, C. 1999. Online association rule mining. Proc. ACM SIGMOD, 145-154.

[8] Kim, Y., Jo, J. Y., and Suh, K. K. 2008. Baseline profile stability for network anomaly detection. International Journal of Network Security 6, 1, 60-66.

[9] Kohonen, T. 1982. Self-organized formation of topologically correct feature maps. Biological Cybernetics 43, 59-69.

[10] MacQueen, J. B. 1967. Some methods for classification and analysis of Multivariate Observations. Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press 1, 281-297.

[11] Mallat, S. and Zhang, Z. 1993. Matching Pursuit with time-frequency dictionaries. IEEE Transactions on Signal Processing 41, 3397-3415.

[12] Palomo, E. J., North, J., Elizondo, D., Luque, R.M., and Watson, T. 2012. Application of growing hierarchical SOM for visualisation of network forensics traffic data. Neural Networks 32, 275-284.

[13] Salama, M. A., Eid, H. F., Ramadan, R. A., Darwish, A., and Hassanien, A. E. 2011. Hybrid Intelligent Intrusion Detection Scheme. Soft Computing in Industrial Applications 96, 293-303.

[14] Sheikhan, M., Jadidi, Z., and Farrokhi, A. 2012. Intrusion detection using reduced-size RNN based on feature grouping. Neural Computing and Applications 21, 1185-1190.

[15] SPSS Corp. 2013. http://www-01.ibm.com/software/analytics/spss/.

[16] Stevanovic, D., Vlajic, N., and An A. 2013. Detection of malicious and non-malicious website visitors using unsupervised neural network learning. Applied Soft Computing 13, 1, 698-708.

[17] Tajbakhsh, A., Rahmati, M., and Mirzaei, A. 2012. Intrusion detection using fuzzy association rules. Applied Soft Computing 9, 462-469.

[18] Thapngam, T., Yu, S., Zhou, W., and Makki, S. K. 2012. Distributed Denial of Service (DDoS) detection by traffic pattern analysis. Peer-to-Peer Networking and Applications, Published online: 25 October 2012.

[19] Tjhai, G. C., Furnell, S. M., Papadaki, M., and Clarke, Na. L. 2010. A preliminary two-stage alarm correlation and filtering system using SOM neural network and K-means algorithm. Computers & Security 29, 6, 712-723.

[20] Werlinger, R., Hawkey, K., Muldner, K., Jaferian, P., and Beznosov, K. 2008. The Challenges of Using an Intrusion Detection System: Is It Worth the Effort? Symposium on Usable Privacy and Security (SOUPS).

[21] Wu, S. X. and Banzhaf, W. 2010. The use of computational intelligence in intrusion detection systems: A review. Applied Soft Computing 10, 1-35.

[22] Xu, D. and Ning, P. 2008. Correlation analysis of intrusion alerts, in: R. Di Pietro, L.V. Mancini (Eds.), Intrusion Detection Systems, Advances in Information Security 38, 65-92.