Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 6; April-June, 2015 pp. 564-568
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/acsit.html

Identifying Network Attack Strategies by Correlating Alerts

Arushi Jain

Ambedkar Institute of Advanced Communication Technologies and Research Delhi, India
E-mail: aashi0702@gmail.com

Abstract—*With the extreme popularity of the Internet and widespread use of automated attack tools, attacks against Internet-connected systems have become commonplace. Not only has the number of incidents increased, but the attackers are using more and more sophisticated methods. Security issues have become major concern for organizations that have networks connected to the Internet. One of the approaches to study the attack strategies is to extract them from the alerts that are generated by IDSs.*
However, as intrusion detection systems are increasingly deployed in the network, they could generate large number of alerts with true alerts mixed with false ones. Manually manage and analyze these alerts is time-consuming and error-prone. Alert correlation helps in automating alert clustering, which then groups logically interconnected alerts into one groups and lets security to analyze the attack easily.

Keywords: *Attack strategies, alerts, correlation, networks, alert clustering.*

1. INTRODUCTION AND RELATED WORK

Businesses that accepttransactions via the Internet can gain a competitive edge by reaching a worldwide customer base at relatively low cost [1]. Systems are employed to detect internet anomaly play a vital role in internet security [2].Since the Intrusion Detection Systems (IDS) are easily deployed in the network, they could generate overwhelming number of alerts with true alerts mixed with false ones. But the Internet poses a unique set of security issues due to its openness and ubiquity.Many of these attack alerts are raised independently, making it hard for network administrator or intrusion response system to understand the real security situation of the network and provide response to the intrusions. Consequently, alert correlation has become a critical process in intrusion detection and response.Alert correlation can be very beneficial especially for intrusion response. Firstly, it reduces the number of alerts that needs to be handled. IDS may generate thousands of alerts per day. One of the tasks of alert correlation is to aggregate duplicate alerts and filter low-interest alerts. After these steps, the number of alerts that is presentedto the network administrators will be greatly reduced.Secondly, because of the problem of false positives, it is impossible to response to every alert that is reported by IDS, only those which are detected with high confidence will be considered for response action.

Generally, for attacks that have great security impact on the protected network such as DOS/DDOS and worms, response will be taken without observing any correlated alerts [4]; this is because these kinds of attacks themselves are not likely to be false alerts due to their complexity. Moreover, the severity is high and therefore should be handled with high priority. However, there may be a possibility that certain type of alert can have high false positive rate, such as password-guessing. In that case, if a response unit receives such an uncorrelated alert, it knows that it could be a false alert, by considering its severity; it might choose not to respond at this time. But, if it receives this alert right after it receives a port scan alert from the same source targeting the same host, it will have higherconfidence that this is a true alert and therefore takes appropriate response actions. One other important use of alert correlation is to recognize the strategy or plan of different intrusions and inference the goal of attacks. Suppose that the ultimate goal of an attacker can be identified by looking at the pattern of the intrusive behavior, we can take action to prevent the attack from escalating and therefore minimize the damage to the asset. Classification may be used togroup various cyber-attacks and then use the profiles to detect an attack when it occurs [5]. Alert correlation provides a way to group different logically-connected alerts into attack scenarios, which allows the network administrator to analyze the attack strategies.Section 2 discusses the problem definition along with the phases of correlation process. Section 3 includes correlation techniques with main emphasis on alert correlation matrix and multilayer perceptron based on neural networks. Subsequent section gives time consideration between different alerts. This paper is lastly concluded by future work and conclusion.

2. PROBLEM FORMULATION

Proposed a new alert correlation technique that is based on neural network approach. The distinguished feature of this approach is that it uses supervised learning method to gain knowledge from the training examples. Once trained, the alert correlation can determine the probability that two alerts should be correlated. Assigning probability of correlation can help constructing hyper-alert graph and attack graph that represent the real attack scenario.Proposed an Alert Correlation Matrix (ACM) that can encode correlation knowledge such as correlation strength and average time interval between two typesof alerts. This knowledge is gained during the training process and is used by correlation engine to correlate future alerts. And besides, different attack graphs can be generated out of ACM, which can help security analyst to study the strategies or plans of attackers. Other benefit of using ACM is that it must be incrementally updated after training process;this enables it to discover the changes in the existing attack patterns or the newly emerging patterns.

2.1. Correlation phases for generated alerts

We divide the process into three stages based on the general data processing procedure:

2.1.1. Preprocess

The primary aim of the preprocess stage is to convert the alerts to a generic format and reduce the number of alerts to be correlated. False alerts need to be handled at an early stage as they will have negative impact on the correlation results. This step includes data normalization, data reduction, alert aggregation, alert filtering and reducing false alerts.

2.1.2. Correlation techniques

Generally, alerts are raised independently by IDSs, however, they may have some kind oflogical connections with each other, and these interrelated alerts might represent attack that contains multiple stages starting from probing to compromising and escalating, or a large scale of cooperative attack. The correlation reconstructs attack scenarios from alerts reported by intrusion detection systems. Measuring the feature similarities is the central part of this approach, in order to find out the overall similarity between two alerts, four metrics are considered: feature overlap, feature similarity, expectation of similarity and minimum similarity.

- **Feature Overlap:** Some common features are shared by two alerts.
  Example:
  1. Same source IP address.
  2. Same class of attack.
  3. Same time information.
     4. **Feature Similarity:** Each of the common features has different similarity function that computes the similarity of the particular feature. A few data mining techniques have used feature selection techniques [6]. For example, the similarity between two source address can be calculated in terms of the higher bit of the IP addresses, while the similarity between attack classes is obtained by consulting an incident class similarity matrix which encodes prior knowledge about this feature.

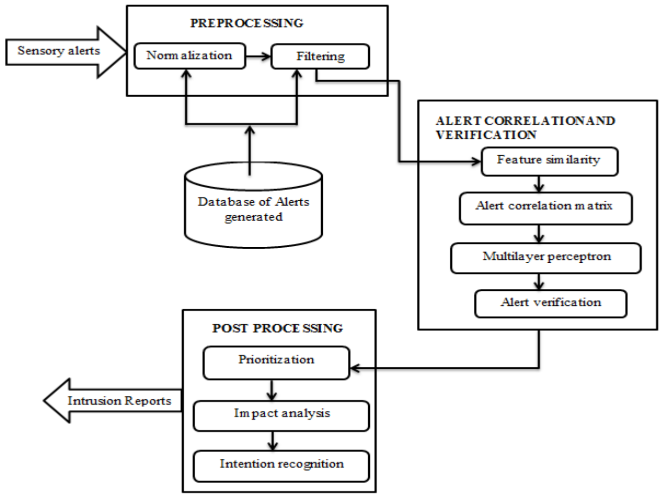

Fig. 1: Alert Correlation phases

- **Minimum Similarity:** The minimum degree of the similarity that must be met for certain features.To determine which feature selection measure, top feature selection and classification algorithm are the best [7]. It expresses the necessary, but not sufficient conditions for correlation. If thesimilarity value of anyfeatureis lower than the corresponding minimum similarity, then the match is rejected regardless of the overall similarity.

The overall similarity of two alerts is computed using following formula:

$$\mathrm{SIM}\,(X^i, Y) = \frac{\sum_{i=0}^{N} E_j\ \mathrm{SIM}\,(X_j^i, Y)}{\sum_j E_j} \qquad (1)$$

Where
X^i = similar candidate Meta alert i for matching
Y = new alert
j = index over the alert features between the alerts
E_j= expectation of similarity for feature j
Xj, Y_j= values for feature j in alerts X and Y alerts.
N= total number of non-redundant similar alerts.

The goal is to find an isuch that SIM(Xi; Y) is maximum. IfSIM(X^i; Y) is greaterthan or equal to minimum similarity, $X^i$and Y will be correlated;otherwise, Y willbecome a new Meta alert.

- **Alert Correlation Based on Prerequisite and Consequence Relationship:** This class of approaches is based on the assumption that most alerts are not at all isolated, but related to different stages of attacks, with the earlier stages preparing for the later ones. Based on this observation, several works propose to correlate alerts using prerequisites and consequences of corresponding attacks. These approaches require specific knowledge and understanding about the attacks in order to identify their prerequisites and consequences. Alerts are considered to be correlated by matching the consequences of some previous alerts and the prerequisites of later ones. For example, if sad mindpingattack is found to be followed by a buffer overflow attack against the corresponding sad mind service, then two alerts may be correlated as part of the same attack scenario.

2.1.3. Post process

- **Alert Prioritization:** Intrusion detection systems often produce numerous alerts each day. Many of these alerts are related to failed attacks and false alarms caused bynormal network traffic. Only a few numbers of the alerts are causedby successful attacks. Even for those real alerts, not all of them are equally important in terms of theirseverity andcriticality of the target being attacked. Therefore, it is essential to separate theimportant alerts from the rest.
  For example,
  DDOS attack requires more priority as compared to password attack for preventing the damage to the information security.
- **Intention Recognition [8]:**Intention or plan recognition is the process of inferring the goals of an intruder by observing theactions. It is of great importance in terms of providing early warning and preventing intrusion from escalating. However, it is also a difficult task since the behavior of intruders is unpredictable; they tend to change their behavior so that they can not to be identified. Knowing the strategies or plans of an intruder is essential to intention recognition. Using alert correlation, the intruders' relevant behavior can be grouped into attack scenarios, and later on, their attack strategy or plans can be extracted.

3. ALERT CORRELATION TECHNIQUES

Some methods such as Probabilistic or neural networks methods classify or correlate the alerts based on the similarity of their chosen common features [9]. Attack strategies are normally represented as attack graphs. These graphs can be manually constructed by security experts using knowledge such as topology and vulnerabilities of the protected network. But this approach is very time-consuming and error prone. Consequently, a number of alert correlation techniques have been introduced in order to help security analysts to learn strategies and patterns of the attackers. However, all these approaches have their own limitations. These techniques either cannot reveal the causal relationship among the alerts or require a larger number of predefined rules in order to correlate alerts and generate attack graphs. In this paper, we propose a complementary alert correlation method that targets the automated construction of attack graphs from a large volume of raw alerts. The proposed correlation method is based on multi-layer perceptron (MLP). We use the probability output of MLP to connect correlated alerts in a way that they represent the attack scenarios.

Overall,the task of such technique is to determine:

- Whether two alerts should be or not to be correlated.
- If yes, the approximate probability with which they are correlated.

MLP can fulfill theseabove requirements, conditionally given that appropriately labeled training data is provided. This probability is requiredbecause it is useful forpractical situation; it also reflects the uncertainty of correlation.

3.1. Alert correlation matrix (ACM)

Table 1: Alert Correlation Matrix

ALERTS	A1	A2	A3	A4
A1	$W_c(A1,A1)$	$W_c(A2,A1)$	$W_c(A3,A1)$	$W_c(A4,A1)$
A2	$W_c(A1,A2)$	$W_c(A2,A2)$	$W_c(A3,A2)$	$W_c(A4,A2)$
A3	$W_c(A1,A3)$	$W_c(A2,A3)$	$W_c(A3,A3)$	$W_c(A4,A3)$
A4	$W_c(A1,A4)$	$W_c(A2,A4)$	$W_c(A3,A4)$	$W_c(A4,A4)$

The correlation strength between two types of alert play an important role in attack pattern analysis [9]. It reveals the causal relationship of the two alerts.An Alert Correlation Matrix (ACM) for n alerts A1, A2, A3…is a matrix with n * n cells, each of which contains a correlation weight of two typesof alert.

Fig. gives an ACM example of five alerts A1; A2; A3; A4, andA5. We use C(Ai; Aj) to denote a cell in ACM for alert Ai; Aj. Note that ACM is not symmetric. It depicts the temporal relation between two alerts. As shown in Figure, C(A1; A2) and C (A2; A1) represent two different temporal relationships. C(A1; A2) suggests that alert A2 arrives after A1, while C(A2; A1) indicates that alert A1 arrives after A2. By distinguishing these two situations, one can gain better understanding on the relationship of the types of attacks.

Each cell in ACM holds a correlation weight. It is computed as follows:

$$W_C\,(A_i, A_j) = \sum_{k=1}^{N} P_{i,j}\,(k) \qquad (2)$$

N is the number of times these two types of alerts have been directly correlated, and P(k) is the probability of the k^{th}correlation, which is given by SVM(support vector machine) correlation engine. These correlation weights are incrementally updated during the training process.Certain knowledge can be inferred based onthe structure of ACM and correlation weights in ACM.

Since ACM encodes the temporal relationship between two types of alerts. Accordingly, two types of correlation strength are defined, namely, Backward Correlation Strength(πb) and Forward Correlation Strength (πf). The reason of having two different types of correlation strength is that, for alert correlation, we are more interested in finding which previous alerts must be correlated with the current one. And for intrusion prediction and attack strategy recognition, we concern about what alert is most likely to happen next. The calculation of πbis basically the normalization of the correlation weight of vertical elements in ACM, while normalizing the correlation weights of horizontal elements gives the πf.

- **Temporal Relationship:** Given two alerts A1 and A2 in the cell C(A1; A2) of ACM, we interpret it as A2 comes after A1. This temporal relationship is important for inferring the causal relationship [10]. For example, for two alerts, the corresponding correlation weight with respect to different temporal relation is given by W_c(A1; A2) and W_c(A2; A1). If these two values turn out be to very close after training, then it is more likely that they do not have any causal relationship. On the contrary, causal relationship can be observed if, for example, one of W_c(A1; A2) and W_c(A2; A1) is much greater than the other one. This suggests that a2 might be the consequence of a1.
- **Causal Relationship**: The causal relationship can be inferred from temporal relationship given that training data is good representative of the real work data. Identifying the causal relationship might help administrator to recognize the strategies or the intention of an attacker [11]. The ultimate goal of alert analysis is to understand the security landscape of the protected network, preventing further attack and therefore minimizing the damage. All these require fully or at least partial understanding of attackstrategies by identifying the logical connection among the alerts.

For example,

1. Anadversary always installs DDOS software before actuallylaunching DDOS attacks.
2. E-mail spams coming in an alarming number is also an indication of identity theft attack or denial of service attack.

If we are able to capture thesecausal relationships, it may help us build stepwise attackscenarios and reveal the adversary's attack strategy.

- **Feature selection[12]:**When an intrusion detection system raises an alert, it also provides information associated with that alert, such as timestamp, source IP address, destination IP address, source port, destination port, and type of the attack. All of these can be used to construct the features for alert correlation technique. The approach presented in this paper is similar to the probabilistic alert correlation technique in a sense that they all involve selecting set features, compute the probability based on these features and decide if two alerts should be correlated. For the proposed correlation technique, the following five features are selected.

Table 2: similarity features

S. No	Feature
F1	The similarity between two sources IP addresses of two alerts.
F2	The similarity between targets IP addresses of alerts by different source IP address i.e. probing port on the same target IP within short time interval.
F3	If the source IP address of the current alert matches the target IP address of a previous alert.
F4	The backward correlation strength (πb) between these two types of alert.
F5	A feature indicating the frequency those two alerts are correlated.

3.2. Alert correlation using multi-layer perceptron

With the growing number of attacks on network infrastructures, the need for techniques to detect and prevent attacks is becoming urgent.Artificial neural networks, commonly referred to as neural networks, have been widely used in many areas for pattern recognition, classification and time series prediction [3].

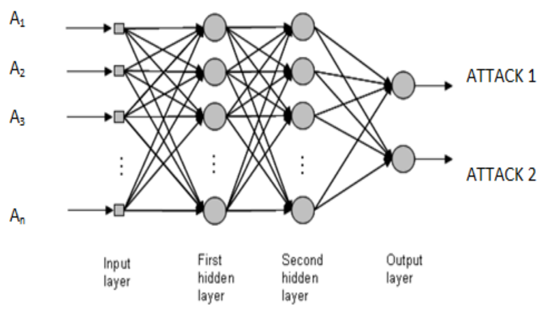

Fig. 2: Multilayer Perceptron

Multi-layer perceptron is an important class of neural networks. Asshown in Figure, multilayer perceptron contains one input layer, one or more hidden layers, andone outputlayer. The neurons areinterconnected in afeed-forward way. The neurons in one layeraredirectly connected to the neurons of thesubsequent layer,and each of them includes a nonlinear activation function. What is important for alert analysis is not only whether two alerts should be correlated, but also the probabilistic with which they should be correlated.

The weight matrix of MLP encodes the knowledge learned from training examples,so the task of the learning process is actually adjusting the weight matrix in order to minimize the error in output. MLP can be applied to solve difficult problems by training them in a supervised manner with particular algorithm, one highly popular algorithm used by MLP for training is known as error back-propagation algorithm.The input is a six vector of six elements, each of which represent one of the features we define above.

Vector(F_R) = [F1; F2; F3; F4; F5; F6]

The output of this MLP is a value between 0 and 1, indicating the probability that two alerts are correlated. The output of correlation engine is used to determine whether or not two alerts are correlated. The goal of correlation process is to construct a list ofhyper-alerts [3]. A new alert is not always linked to the latest alert in the hyper-alert. Instead, it connects to the alert that is most probable to be correlated with itself. So, the representation is useful to inference the multiple goals of attackers. The intentionof using graph representation for correlated alerts is to give network administrator intrinsic view of attack scenarios. In order to construct the hyper-alert graphs, the threshold is defined called correlation threshold. Since the probabilistic output of MLP is between 0 and 1, intuitively, a value of 0.5 is asuitable threshold to decide if the two alerts can be correlated or not.

4. CONCLUSION

This research work carries out an alert correlation technique based on MLP and ACM. The input vector for both MLP consists of six elements representing six alert features. Each of these features is extracted from the attributes of two alerts to be considered. The probabilistic outputs of MLP are used to determine if two alerts are correlated. The outputs are also used to update the correlation weight in an Alert Correlation Matrix (ACM). The use of the ACM can be divided into two parts. Firstly, the correlation weights in ACM are used to enforce the correlation. Secondly, the attack strategies are also extracted based on the correlation weights in ACM. Also described in this paperis the graph representation for correlated alerts called hyper-alert graphs, as well as a similar graph representation for attack strategies called attack graphs. Both of these representations are defined in a way that they can be easily understood by network administrators.

5. FUTURE WORK

Even though the goals of correlation seem to be well-defined, the correlation approaches proposed so far emphasize different aspects of the correlation process, making it difficult to compare the results of each solution

Some extensions that can be made are:

- **Recognizing the variations of attack strategies:** Attackers often change attack patterns to achieve their goal. For example, they may use different attacks to gather information and compromise the target system to gain root access. Attack graphs provide useful information for analyst to study the vibrations of attack pattern. But automated analysis techniques on top of attack graphcan greatly reduce the analyst's work load and therefore should be further investigated.
- **Identifying more features for correlation:** This paper only uses six features for alert correlation. MLP have the ability to handle high dimensionalinput. So the proposed alert correlation technique can be improved by introducing more features.

REFERENCES

[1] Manu Malek and FotiosHarmantzis," security management of web servlces", Steven Institute of Technology.

[2] Da-pengchen, xiao-song zhang," internet anomaly detection with weighted fuzzy matching over frequent episode rules",IEEE,2008.

[3] PengNing, DingbangXu," Learning Attack Strategies from Intrusion Alerts",ACM,2003.

[4] Ashley Chonka, Wanlei Zhou, Jaipal Singh, Yang Xiang," Detecting and Tracing DDoS attacks by Intelligent Decision Prototype", Sixth Annual IEEE International Conference on Pervasive Computing and Communications,2008.

[5] Dr. BhavaniThuraisingham," Data Mining for Security Applications", The University of Texas at Dallas, Richardson.

[6] Christine Dartigue, Hyun Ik Jang, and WenjunZeng," A New Data-Mining Based Approach for Network Intrusion Detection", Seventh Annual Communications Networks and Services Research Conference IEEE,2009.

[7] Robert Moskovitch, Ido Gus, Shay Pluderman, DimaStopel, ChananGlezer, Yuval Shahar and Yuval Elovici," Detection of Unknown Computer Worms Activity Based on Computer Behavior using Data Mining", IEEE Symposium on Computational Intelligence in Security and Defense Applications,2007.

[8] HuwaidaTagelsir Ibrahim Elshoush," An Innovative Framework for Collaborative Intrusion Alert Correlation", Science and Information Conference IEEE,2014.

[9] Chih-Hung Wang and Ji-Min Yang," Adaptive Feature-Weighted Alert Correlation System Applicable in Cloud Environment", Eighth Asia Joint Conference on Information Security IEEE,2013.

[10] DonghaiTian, Hu Changzhen, Yang Qi," Hierarchical Distributed Alert Correlation Model", Fifth International Conference on Information Assurance and Security IEEE,2009.

[11] Samira Lagzian, FatemehAmiri, AliRezaEnayati, HossienGharaee," Frequent Item set mining-based Alert Correlation for Extracting multi-stage Attack Scenarios", 6'th International Symposium on Telecommunications IEEE,2012.

[12] AleksandarLazarevic, DragoijubPokrajac, JelenaNikolic," Applications of Neural Networks in Network Intrusion Detection", 8th Seminar on Neural Network Applications in Electrical Engineering,2006.