김일성종합대학학보
(자연과학)
주체103(2014)년 제60권 제6호

JOURNAL OF KIM IL SUNG UNIVERSITY
(NATURAL SCIENCE)
Vol. 60 No. 6 JUCHE103(2014).

A Method of Processing Anomaly Alerts

Kong Hye Ok

As computer users can be seriously harmed by hackers who try to damage computer systems and steal data, it is very important to protect computer systems and data from such intrusions.

A great deal of research has been undertaken into intrusion detection systems.

Intrusion detection is a process of recognizing a malicious behavior directed towards computer systems and their resources and of dealing with them. An Intrusion Detection System (IDS) is a system with which to realize this process.

Network IDS analyzes all communication data on the network in real time usually by using network connection devices. All this time it detects attacks by using a number of methods including pattern comparison, frequency comparison, threshold comparison, statistical anomaly detection, immediately gives an alarm and takes an action. These actions include administrator's alarm, disconnecting, recording, and analyzing attack data.

Network security systems witness increasing attempt to detect accurately malicious intrusions by encouraging human analysts to check all the warnings. Without automated support, however, this manual way of checking has proved difficult due to the great number of alerts that have to deal with. Sensors can generate about 850 000 alerts, among which over 18 000 are estimated to be serious. To help the network security analysts who have to deal with all this enormous flood of warnings, an effective use has been made of data mining methods. [1]

Reference [1] applied a classification method to the problem of intrusion detection in order to classify events into separate attack categories and Reference [2] used it to characterize normal use of a network service [2].

My work aims to improve the performance of the existing network defense system by adding data mining methods instead of replacing the current intrusion detection methods. Therefore, I use network warning data from IDS instead of the direct network data as in Ref. [1, 2], thus reducing the number of false warnings for humans to check by employing data mining methods for protection of existing data.

Reference [3], which deals with a kind of indirect data instead of direct data of sensors, used a classification algorithm called RIPPER to update the rules used by a commercial real-time network monitoring tools. Reference [4] used association rules to reduce false warnings generated by sensors. Reference [5] made a statistical analysis of false alarms.

In my paper, IDS' anomaly alerts are classified additionally into corresponding attacks according to certain criteria, so that alerts for attacks will disappear out of human analyst's sight, reducing the burden of the network security analyst. And I provide a computational model which scores the warnings classified as corresponding events to help the analyst find clues for unusual behaviors.

1. Procedures for Classifying Alerts

Fig. 1 illustrates our use of data mining.

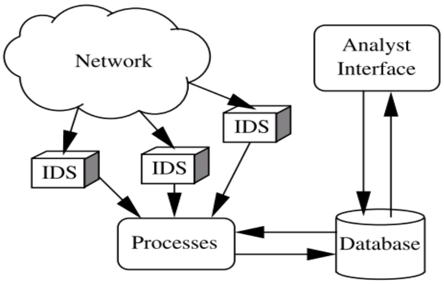

Fig. 1. An architecture of network security system

Network communication is analyzed by a variety of available signature-based intrusion detection sensors. All this sensor data is sent periodically to a central server for processing, and loaded into a relational database. Analysts review incident data and individual alerts through the analyst interface.

The processing includes data loading, aggregation, filtering, classification, and ranking to support data analysis as shown in Fig. 2.

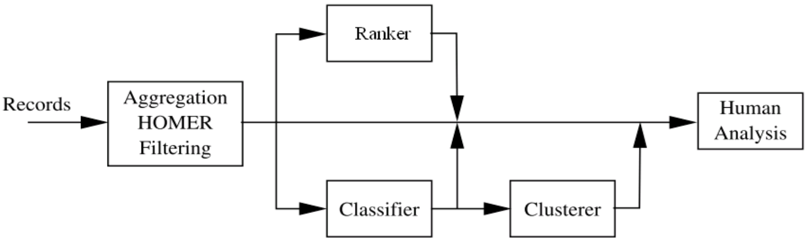

Fig. 2. Data mining processes in intrusion detection system

First, hundreds of thousands of warnings from IDS are aggregated by source IP address into thousands of processor units. A filter classifies these units. If the number of destination IP addresses is greater than a threshold, these corresponding events are classified as attacks and then a ranker ranks them according to the computed severity. Those warnings estimated not to be part of corresponding events are sent to a classifier to filter false alarms. Those records found to be benign by the classifier are sent to a cluster-based anomaly detector for additional testing. The following figure shows each of these processes in more detail. (Fig. 3.)

The aggregation process operates on a batch of alert records from several sensors in the database. It aggregates records generated during a time window according to a simple aggregation criterion, such as the common source IP address.

Aggregated data that passes a filter is tagged by the filter classifier as corresponding events and all associated records are then removed from analyst’s view. Thus, the classifier tags obvious corresponding events in order to reduce analysts’ burden.

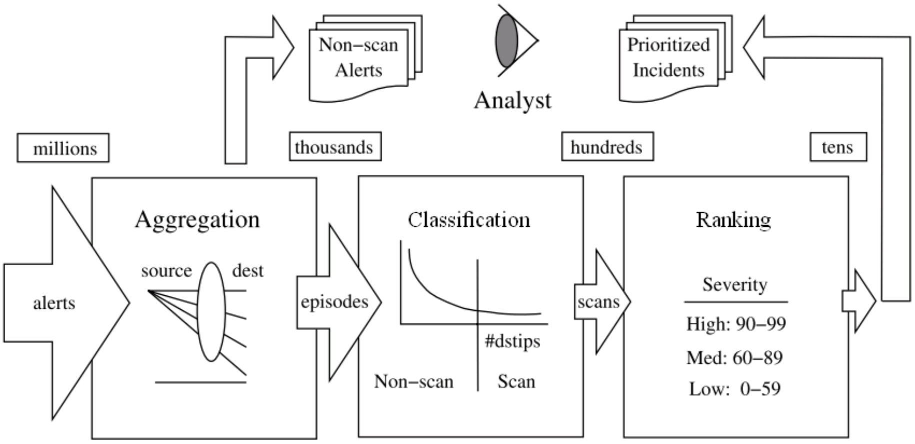

Fig. 3. Aggregation, mapping episode classifier, and tagging review

If the number of destination addresses in the aggregate incident exceeds a threshold, set by domain experts (for example, 99.5%), then it is classified as a *corresponding event.* All alerts associated with an event are summarized so that analysts do not have to check them individually. Thus, if a source IP communicates with an unusually large number of destination machines in a short period of time, the associated records are summarized and removed from analyst`s view.

Ranking is a process that generates a score for aggregated network sensor alerts that have been tagged as corresponding events by the filter classifier. The score indicates the potential that the corresponding event is more than a benign event and can help analysts pay attention to critical alerts.

It looks for clues of unusual behaviors in a collection of alerts that make up a corresponding event.

2. A Computational Model for Alerts Classification

The ranking stage of alerts classification mentioned above generates scores for aggregated network sensor alerts that have been tagged as corresponding events to look for clues of unusual behavior.

In this section you will see a computational model for scoring. A score is defined as numerical values of three metrics for detecting anomalous incidents: coverage, popularity, and uniqueness. The individual alert here that is collected and tagged by the sensor is called a cell event.

Coverage Let us consider a group of destination IPs that each warning in a corresponding event sets as its target. If destination IPs that serve as the target of a corresponding event, the event is simply regarded as a single mapping.

On the other hand, if a particular cell event covers only a small portion of destination IPs, it indicates that vulnerability has been detected and further exploits have been attempted against a host. This also indicates that it's more inclined to attack toward a small proportion of targets.

We compute coverage as $c = 99\left(1-\min\limits_{E}(n_{\text{dstip}})\big/N_{\text{dstip}}\right)$ where E is the set of cell events in the corresponding event, $\min\limits_{E}(n_{\text{dstip}})$ is the minimum value of distinct destination IPs of a certain cell event in the corresponding event, and N_{dstip} is the total number of destination IP addresses in the corresponding event.

The coverage detects cell events whose target is just a small proportion of destination IPs in a corresponding event. Coverage is inversely proportional to the ratio of the destination addresses with the least distribution within the cell event in the corresponding event.

Suppose, for example, that there are 1 000 events labelled "Scan Proxy" with 1 000 individual destination IPs and two cell events tagged as "WEB MISC" with two of the destination IPs among the 1 000. If all these cell events share the same source IP, $\min\limits_{E}(n_{\text{dstip}}) = 2$ and $N_{\text{dstip}} = 1\,000$, so $c = 98.8$. The "WEB MISC" events which fail to cover the IP space that are covered by the Scan Proxy investigation are suspected.

Popularity In a benign corresponding event, the number of warnings will be evenly distributed over the whole of destination IPs. If a particular target is significantly" popular" (targeted with a big proportion of the records in the event), the additional attention toward one host might indicates that it has been attacked.

We can work out the popularity with the following formula $p = \begin{cases} 5\nu - 1, & \nu < 20 \\ 99, & \text{otherwise} \end{cases}$, $\nu = \max\limits_{D}(n_r)\big/\bar{n}_d$ and D is the set of destination IP addresses in the event, $\max\limits_{D}(n_r)$ is the maximum number of records associated with any destination IP in the event, and $\bar{n}_d$ is the average number of records related to the destination IP in the event.

The popularity metric detects corresponding events that target destination IPs with a large number of irregular records. Popularity is proportional to the ratio of the number of maximum cell events per destination address to the average number of events per destination address.

For example, when there are 1 000 events labeled "Scan Proxy" which have 1 000 individual destination IPs and additional 100 cell events of various types to one destination IP in the set, all sharing the same source IP, $\max\limits_{D}(n_r) = 101$ and $\bar{n}_d = 1.1$ and therefore $\nu = 9.18$ and $p = 99$. This means that the additional events directed toward the single IP are suspicious because they target one particular host, showing special interest to the source in that host.

Independence There are a great number of records associated with each cell event in the corresponding event. In a benign corresponding event, it is likely that the number of records for each event can be almost the same. If a particular event has a smaller number of records than the average number of records per cell, it means that this is an independent cell event or maybe a corresponding event which includes a certain additional behavior.

We can work out independence with the following formula

$$u = 99\left(1 - \min_{E}(n_r) \Big/ N_r\right)$$

where E is the set of cell events in a corresponding event, $\min_{E}(n_r)$ is the minimum number of records in the cell events, and N_r is the total number of records in the corresponding event.

The independence metric detects corresponding events when there are few records associated with any given cell event in a corresponding event. Independence is inversely proportional to the percentage of records associated with the least used cell event in the corresponding event.

For example, if there are 1,000 events labeled "Scan Proxy" with 1 000 individual destination IPs and a cell event tagged as "WEB MISC" with one destination IP, $\min_{E}(n_r) = 1$ and $N_r = 1\,001$, so $u = 98.9$. The "WEB MISC" event is suspicious because it only has one waring, whereas cell events in benign corresponding events usually direct most or all of the destination IPs in the corresponding event.

Ranking Let's see the scaled function of coverage, popularity, and uniqueness metrics as $m' = 1 - \alpha \ln(1 + m/100)$ where m is the metric (i.e., *c*, *p*, or *u*), m' is the scaled metric, and α is a tunable weight factor. That is, this scaled function of metrics is an opposition function which increases sharply as the metric decreases.

In this experiment, we weighted popularity and emphasized the coverage with independence as $\alpha \approx 0.1$ and coverage as $\alpha \approx 0.2$.

The ranking combines values of coverage, popularity, and independence metrics into a single score, *s*. The combination could be done in many ways. We can use a simple formula of $s = 100(1 - c'p'u')$ where c', p', u' are scaled functions of the coverage, popularity, and independence metrics. Thus, the bigger the metrics, the greater the value of *s*.

According to the value of *s*, the rank of attacks can be divided into high (90－99), med (60－89) and, low (0－59), and we can take appropriate measures according to this division. Those warnings not filtered by attacks could be sent to a false alarms classifier to be properly processed.

3. Conclusion

This paper provides a procedural model for classifying anomaly alerts into corresponding events as well as a computational model for scoring the network sensor warnings that have been tagged as corresponding events by the filter classifier so as to reduce the number of false alerts made in the existing network sensors and to detect extraordinary warnings corresponding to new attacks. As a result of our experiments of 5 days, 71094 warnings regarded as serious were reduced to 1011.

This method of processing warnings can be applied not only to anomaly detection system but also to signature –based detection and the IDS based on either the network or the host.

References

[1] C. Elkan; Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining 1, Kluwer Academic Publication, 63～64, 2000.

[2] W. Lee et al.; In Proceedings of the 7th USENIX Security Symposium, 79～94, 1998.

[3] W. Lee et al.; In Proceedings of the 1st Workshop on Intrusion Detection and Network Monitoring, 5～14, 1999.

[4] S. Manganaris et al.; Computer Networks, 34, 571, 2000.

[5] Cheng Yuan Ho et al.; IEEE Communications Magazine, 3, 146, 2012.