International Journal of Advanced Computer and Mathematical Sciences
ISSN 2230-9624. Vol 3, Issue 4, 2012, pp 424-429
http://bipublication.com

ADVANCE TECHNIQUES FOR THE DETECTION OF INTERRUPTION ALERTS

SAIKIRAN B

Dept of IT, Guru Nanak Engineering College, Hyderabad, A.P, India

[Received-11/09/2012, Accepted-02/10/2012]

ABSTRACT:

The important subtask of intrusion detection is Alert aggregation. Protecting our data in the internet is a great risk. Intruders and hackers make an attempt to grab our data. Intrusion detection is an emerging technology to identify unauthorized users and to cluster different alerts produced by low-level intrusion detection systems firewalls. The relevant information whereas the amount of data can be reduced substantially by Meta-alters which can be generated for the clusters. At a certain point in time which has been initiated by an attacker is belonging to a specific hacking attack instance. For communication within a distributed intrusion detection system the meta-alerts may be the basis for reporting to security experts. In this paper, for online alert aggregation we propose a novel technique which is based on a dynamic and probabilistic model of current attack situation. For the estimation of the model parameters, it can be regarded as a data stream version of a maximum likelihood approach. The first alerts belonging to a new attack instance are generated with meta-alerts with a delay of typically only a few seconds. In this paper, to achieve reduction rates while the number of missing meta-alerts is extremely low can be possible with the three benchmark data sets are demonstrated.

Keywords: Hacking attack, Meta-alters, Intrusion detection, Alert aggregation.

INTRODUCTION:

The virtual private networks, authentication mechanisms, and encryption techniques are the other protective measures besides Intrusion detection systems (IDS), and are very important to guarantee information security. Protecting our data in the internet is a great risk. Intruders and hackers make an attempt to grab our data. By detecting the actions of attackers or attack tools in a network or host-based manner with misuse or anomaly detection techniques the IDS helps to defend against the various threats to which networks and hosts are exposed to. At present, by evaluating TCP/IP connections or log files, the suspicious actions are detected by most IDS which are quite reliable. If once IDS find a suspicious action, then it immediately creates an alert. IDS do not distinguish between different attack instances but they usually focus on detecting attack types. Thousands of network packets or log file entries are inspected, if even low rates of false alerts could easily result in a high total number of false alerts.

The occurrence of an attack of a particular type that has been launched by a specific attacker at a certain point in time as the intrusive actions caused by a single attack instance which are often spread over many network connections, and a single attack instance often results in hundreds or even thousands of alerts. The IDS creates many alerts at a low level of abstraction as a consequence. The flood of alerts, and decisions that follow from single alerts might be wrong with a relatively high probability it is extremely difficult for a human security expert to inspect. In this paper, without losing any important information which is necessary to identify ongoing attack instances substantially we has to reduce the amount of alerts is our main goal. At any point in time the IDS should "know" what is going on in its environment regarding attack instances and attackers if perfect IDS should be situation-aware in the sense.

In this paper, by introducing and evaluating a new technique for alert aggregation we make an important step towards the goal. We accept false or redundant meta-alerts to a certain degree in turn if we want to have no missing meta-alerts. Alerts may originate from low-level IDS such as those mentioned above, from firewalls etc. The current solutions are typically based on a quite simple sorting of alerts, as this problem is not new, according to their source, destination, and attack type. Uncertainty with respect to the source of the attack due to spoofed IP addresses or the presence of classification errors of the low-level IDS or wrongly adjusted time in windows are under real conditions, for instance, such an approach fails quite often. The distinct properties of our approach are: Using probabilistic methods the generative modeling approach is used. Using approximate maximum likelihood parameter estimation, the attack instances can be regarded as random processes "producing" alerts, which we aim at modeling these processes. Thus, the completion of attack instances can be detected. Each observed alert is processed only a few times is data stream approach.

RELATED WORK:

To detect attacks with high accuracy the existing IDS are optimized. However, the IDS have been outlined in a number of publications as it has various disadvantages and a lot of work has been done to analyze IDS in order to direct future research. Besides others, the large amount of alerts produced is the one main drawback. The correlation of alerts from IDS focuses on the recent research. All approaches outlined in the following present either online algorithms or as we see it if not stated otherwise, can easily be extended to an online version. The attack thread reconstruction is the one step presented correlation approach, which can be seen as a kind of attack instance recognition. A strict sorting of alerts within a temporal window of fixed length according to the source, destination, and attack classification is used but no clustering algorithm is used. Alerts that share the same quadruple of source and destination address as well as source and destination port is used to eliminate duplicates. In order to provide a more condensed view of the current attack situation, the alerts are aggregated into predefined clusters.

A group of alerts belong to alert clustering or even though the same attack occurrence is

used called clustering and there is no clustering algorithm in a classic sense. A similarity relation which is based on expert rules is used to group similar alerts together as the alerts from one IDS are stored in a relational database. For instance, with imperfect classifiers as two alerts are defined to be similar as these approaches are likely to fail under real-life conditions, with false alerts or wrongly adjusted time windows. A weighted, attribute-wise similarity operator is used to decide whether to fuse two alerts or not when another approach to alert correlation is presented in. However, this approach suffers from the high number of parameters that need to be set. The first, quite simple one groups alerts according to their source IP address only. The different supervised learning techniques are based on the other two approaches. Labeled training data need to be generated which could be quite difficult in case of various attack instances due to the supervised nature. In many other approaches to alert correlation as described so far has the same or quite similar techniques, especially in the field of intrusion scenario detection.

An offline clustering solution based on the CURE algorithm is presented and solution is restricted to numerical and cluster attributes. A repeatedly fuse alerts into more generalized ones is used by a link-based clustering approach. The attribute-oriented induction algorithm is forced “to find large clusters” as the only root causes that are responsible for a majority of alerts are of interest and as the alert load can thus be reduced at most. A small number of alerts are likely to be ignored completely as a result of attack instances. The algorithm can only be used in an offline setting and is intended to analyze historical alert logs as the main difference to our approach. To reduce the amount of created alerts the created cluster structure is used as a filter. A completely different clustering approach is presented. The reconstruction error of an auto associator neural network (AA-NN) is used to distinguish different types of alerts. The same reconstruction error are put into the same cluster as they yield alters. An offline training phase and training data are needed to train the AA-NN as the approach can be applied online, but to manually adjust intervals for the reconstruction error that determine which alerts are clustered together.

ONLINE ALERT AGGREGATION TECHNIQUE:

Online aggregation alert is trivial change, over the offline alert. At each point in time based on a probabilistic model of the current situation our new alert aggregation approach is used. We start with a short sketch of our intrusion framework to outline the preconditions and objectives of alert aggregation. The generation of alerts and the alert format are briefly described. The alerts are clustered systematically and grouped continuously. For offline alert aggregation which is basically a parameter estimation technique for the probabilistic model we continue with a new clustering algorithm. An offline method to an algorithm is extended by online alert aggregation which is applied by the data stream clustering. Finally, we make some remarks on the generation of meta-alerts. Collaborating Intrusion Detection Agents: The network and the host on which the agent resides are provided by the interface of sensor layer. The filter incoming

data, and extract interesting and potentially valuable information which the sensors acquire raw data from both the network and the host, are needed to construct an appropriate event. The alert aggregation module has to combine alerts that are assumed to belong to a specific attack instance at the alert processing layer. Thus, so called meta-alerts are generated. Scenario detection or decentralized alert correlation are used or enhanced by meta-alerts in various ways. An important task of the reaction layer is reporting. In this paper we want to have a minimal number of missing meta-alerts, with the alert aggregation module on which we focus and we accept some false meta alerts and redundant meta-alerts in turn.

PROPOSED SYSTEM:

In this proposed system, by introducing and evaluating a new technique for alert aggregation we make an important step toward the goal. Alerts may originate from low-level IDS such as those mentioned above, from firewalls etc. Without losing any important information which is necessary to identify ongoing attack instances substantially to reduce the amount of alerts is the main goal. To model the current attack situation we use an online approach. A filter is used to reduce the amount of created alerts. Only a few times each observed alert is processed. Under harsh timing constraints it can be applied online. The algorithm can only be used in an offline setting and is intended to analyze historical alert logs is the main difference to our approach. In contrast, we use an online approach to model the current attack situation.

CONCLUSION

In this paper, we demonstrated the applicability of online alert aggregation approach. Component and runtime creation delay are well suited for an online application. In this paper, by using three benchmark data sets the reduction rates are extremely low with the number of missing meta-alerts which are demonstrated here. The amount of data could be reduced substantially in all cases. The instance detection rate is very high although there are situations that described are especially clusters that are wrongly split. Only very few attack instances were missed.

REFERENCES:

[1] S. Axelsson, “Intrusion Detection Systems: A Survey and Taxonomy,” Technical Report 99-15, Dept. of Computer Eng., Chalmers Univ. of Technology, 2000.

[2] M.R. Endsley, “Theoretical Underpinnings of Situation Awareness: A Critical Review,” Situation Awareness Analysis and Measurement, M.R. Endsley and D.J. Garland, eds., chapter 1, pp. 3-32, Lawrence Erlbaum Assoc., 2000.

[3] C.M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[4] M.R. Henzinger, P. Raghavan, and S. Rajagopalan, Computing on Data Streams. Am. Math. Soc., 1999.

[5] A. Allen, “Intrusion Detection Systems: Perspective,” Technical Report DPRO-95367, Gartner, Inc., 2003.

[6] F. Valeur, G. Vigna, C. Kru¨ gel, and R.A. Kemmerer, “A Comprehensive Approach to Intrusion Detection Alert Correlation,” IEEE Trans. Dependable and Secure Computing, vol. 1, no. 3, pp. 146-169, July-Sept. 2004.

[7] H. Debar and A. Wespi, “Aggregation and Correlation of Intrusion-Detection Alerts,” Recent Advances in Intrusion Detection, W. Lee, L. Me, and A. Wespi, eds., pp. 85-103, Springer, 2001.

[8] D. Li, Z. Li, and J. Ma, “Processing Intrusion Detection Alerts in Large-Scale Network,” Proc. Int’l

Symp. Electronic Commerce and Security, pp. 545-548, 2008.
[9] F. Cuppens, "Managing Alerts in a Multi-Intrusion Detection Environment," Proc. 17th Ann. Computer Security Applications Conf. (ACSAC '01), pp. 22-31, 2001.
[10] A. Valdes and K. Skinner, "Probabilistic Alert Correlation," Recent Advances in Intrusion Detection, W. Lee, L. Me, and A. Wespi, eds. pp. 54-68, Springer, 2001.
[11] K. Julisch, "Using Root Cause Analysis to Handle Intrusion Detection Alarms," PhD dissertation, Universita¨ t Dortmund, 2003.
[12] T. Pietraszek, "Alert Classification to Reduce False Positives in Intrusion Detection," PhD dissertation, Universita¨ t Freiburg, 2006.
[13] F. Autrel and F. Cuppens, "Using an Intrusion Detection Alert Similarity Operator to Aggregate and Fuse Alerts," Proc. Fourth Conf. Security and Network Architectures, pp. 312-322, 2005.
[14] G. Giacinto, R. Perdisci, and F. Roli, "Alarm Clustering for Intrusion Detection Systems in Computer Networks," Machine Learning and Data Mining in Pattern Recognition, P. Perner and A. Imiya, eds. pp. 184-193, Springer, 2005.
[15] O. Dain and R. Cunningham, "Fusing a Heterogeneous Alert Stream into Scenarios," Proc. 2001 ACM Workshop Data Mining for Security Applications, pp. 1-13, 2001.
[16] P. Ning, Y. Cui, D.S. Reeves, and D. Xu, "Techniques and Tools for Analyzing Intrusion Alerts," ACM Trans. Information Systems Security, vol. 7, no. 2, pp. 274-318, 2004.
[17] F. Cuppens and R. Ortalo, "LAMBDA: A Language to Model a Database for Detection of Attacks," Recent Advances in Intrusion Detection, H. Debar, L. Me, and S.F. Wu, eds. pp. 197-216, Springer, 2000.
[18] S.T. Eckmann, G. Vigna, and R.A. Kemmerer,"STATL: An Attack Language for State-Based Intrusion Detection," J. Computer Security, vol. 10, nos. 1/2, pp. 71-103, 2002.
[19] A. Hofmann, "Alarmaggregation und Interessantheitsbewertung in einem dezentralisierten Angriffserkennung system," PhD dissertation, Universita¨t Passau, under review.
[20] M.S. Shin, H. Moon, K.H. Ryu, K. Kim, and J. Kim, "Applying Data Mining Techniques to Analyze Alert Data," Web Technologies and Applications, X. Zhou, Y. Zhang, and M.E. Orlowska, eds. pp. 193-200, Springer, 2003.
[21] J. Song, H. Ohba, H. Takakura, Y. Okabe, K. Ohira, and Y. Kwon, "A Comprehensive Approach to Detect Unknown Attacks via Intrusion Detection Alerts," Advances in Computer Science—ASIAN 2007, Computer and Network Security, I. Cervesato, ed., pp. 247-253, Springer, 2008.
[22] R. Smith, N. Japkowicz, M. Dondo, and P. Mason, "Using Unsupervised Learning for Network Alert Correlation," Advances in Artificial Intelligence, R. Goebel, J. Siekmann, and W. Wahlster, eds. pp. 308-319, Springer, 2008.
[23] A. Hofmann, D. Fisch, and B. Sick, "Identifying Attack Instances by Alert Clustering," Proc. IEEE Three-Rivers Workshop Soft Computing in Industrial Applications (SMCia '07), pp. 25-31, 2007.
[24] M. Roesch, "Snort—Lightweight Intrusion Detection for Networks," Proc. 13th USENIX Conf. System Administration (LISA '99), pp. 229-238, 1999.
[25] O. Buchtala, W. Grass, A. Hofmann, and B. Sick, "A Distributed Intrusion Detection Architecture with Organic Behavior," Proc. First CRIS Int'l Workshop Critical Information Infrastructures (CIIW '05), pp. 47-56, 2005.
[26] D. Fisch, A. Hofmann, V. Hornik, I. Dedinski, and B. Sick, "A Framework for Large-Scale Simulation of Collaborative Intrusion Detection," Proc. IEEE Conf. Soft Computing in Industrial Applications (SMCia '08), pp. 125-130, 2008.
[27] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, second ed. Wiley Interscience, 2001.
[28] IANA, "Port Numbers," http://www.iana.org/assignments/ port-numbers, May 2009.
[29] Y. Rekhter, B. Moskowitz, D. Karrenberg, and G. de Groot, "RFC 1597—Address Allocation for PrivateInternets,"http://www.faqs.org/rfcs/rfc1597.html, Mar. 1994.
[30] J. Postel, "RFC 790—Assigned numbers,"http://www.faqs.org/ rfcs/rfc790.html, Sept. 1981.
[31] O. Buchtala, A. Hofmann, and B. Sick, "Fast and Efficient Training of RBF Networks," Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003, O. Kaynak, E. Alpaydin, E. Oja, and L. Xu, eds., pp. 43-51, Springer, 2003.

[32] R.P. Lippmann, D.J. Fried, I. Graf, J.W. Haines,K.R. Kendall, D. McClung, D. Weber, S.E. Webster, D. Wyschogrod, R.K. Cunningham, and M.A. Zissman, "Evaluating Intrusion Detection Systems: The 1998 DARPA Offline Intrusion Detection Evaluation," Proc. DARPA Information Survivability Conf. and Exposition (DISCEX), vol. 2, pp. 12-26, 2000.

[33] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On Clustering Validation Techniques," J. Intelligent Information Systems, vol. 17, nos. 2/3, pp. 107-145, 2001.

[34] J.C. Dunn, "Well Separated Clusters and Optimal Fuzzy Partitions," J. Cybernetics, vol. 4, pp. 95-104,1974.