Alert Management for Snort IDS Using Pattern Matching

Conference Paper · May 2016

CITATION
1

READS
945

3 authors:

El Mostapha Chakir
HENCEFORTH
16 PUBLICATIONS **58** CITATIONS
SEE PROFILE

Youness Idrissi Khamlichi
Sidi Mohamed Ben Abdellah University, National School of Applied Sciences, Fez, …
35 PUBLICATIONS **247** CITATIONS
SEE PROFILE

Mohamed Moughit
Université Hassan 1er
18 PUBLICATIONS **56** CITATIONS
SEE PROFILE

Alert Management for Snort IDS Using Pattern Matching

El Mostapha CHAKIR
Laboratory of Computer Networks, Mobility and Modeling IR2M
Faculty of Science and Technology
University Hassan First Settat, Morocco
Email: e.chakir@uhp.ac.ma

Youness IDRISSI KHAMLICHI
Laboratory of Computer Networks, Mobility and Modeling IR2M
National School of Applied Sciences
University Sidi Mohamed Ben Abdellah
FES, Morocco
Email: youness.khamlichi@usmba.ac.m

Mohamed MOUGHIT
Laboratory of Computer Networks, Mobility and Modeling IR2M
National School of Applied Sciences
University Hassan First Settat, Morocco
Email: mohamed.moughit@uhp.ac.ma

Abstract*—Intrusion Detection Systems (IDSs) are used to monitor computer systems for signs of security violations. Having detected such signs, IDSs trigger alerts to report them. These alerts are presented to an analyst process, who evaluates them and initiates an adequate response. In practice, IDSs have been observed to trigger thousands of alerts per day, most of which are mistakenly triggered by benign events. This makes it extremely difficult for the analyst process to correctly identify alerts related to attacks. This weakness has led to the emergence of many methods in which to deal with these alerts. Thus, In this paper a new alert clustering algorithm for IDS Alert Management System is proposed and it is based on the Pattern Matching. The proposed algorithm reduces the number of alerts and detects false positive alerts and low importance events. Experimental results on DARPA KDD cup 99 proofs the efficiency of our algorithm when clustering and classifying alerts while leading to a considerable decrease of false positive alerts.

Keywords— Intrusion Detection System, false positive, Snort, alert processing, Pattern Matching.

I. INTRODUCTION

The explosive increase in the number of networked machines and the widespread use of the Internet in organizations have led to an increase in the number of unauthorized activities, not only by external attackers but also by internal sources, such as fraudulent employees or people abusing their privileges for personal gain or revenge. As a result, intrusion detection systems (IDSs), as originally introduced by Anderson [20] and later formalized by Denning [21], have received increasing attention in recent years.

IDSs is a complementary approach to security. They are used to detect intrusions in an efficient manner and when it observes any suspicious event representing a threat or abnormal behaviour, which may lead to infrastructure damage it produces alerts [3].

However, most of popular IDSs suffer from generating a huge number of false alerts which could be of two types. The first one is called false positive and is generated mistakenly by the IDS as an evidence of malicious behaviour of the system, but in reality, it is not such behaviour. The second type of false alerts is called false negative and is generated by the IDS as an evidence of non-malicious event, but in reality, it should be an indication of malicious activity in the system [10]. Previous research on this area reports that this value could be as high as several hundred thousand a day but around 99% of them are false alerts [4, 5]. Generally, network security administrator needs to analyse each IDS alerts manually, whether it is a false or a true alert leading to time consuming. The present work is taken up to save the time and the effort of the network security administrator, by proposing a new approach that minimizes false alerts and distinguish real attacks from false positives and low importance events.

This paper is organized as follows: Section II explains the false positive reduction techniques, Section III presents Related Works, Section IV explains the Proposed Model and Section V establishes the conclusion and highlights future works.

II. ALERT MANAGEMENT CONCEPTS

An IDS produces an alert or alarm immediately after detecting a malicious activity. Usually, an alert contains several features including: Sensor Id; Alert Id; Date and Time the alert occurred; Rule generating the alert; Source IP Address; Destination IP Address; Source Port; Destination Port; Protocol; Priority; Xref and Classification [11].

IDSs are popularly known to indiscriminately generate voluminous number of different alerts with a slight deviation from normal event or behavior [9]. Often, the raw alerts are a mixture of non-redundant, similar, unrelated, relevant, non-relevant, true alerts, false alerts, frequent, non-frequent, interesting, non-interesting, severe and non-severe alerts. According to Robertson [22], when a sensor outputs an alert, there are three possibilities: The sensor has correctly identified a successful attack. This alert is most likely relevant (true alert or

interesting alert); the sensor has correctly identified an attack, but the attack failed to meet its objectives (non-relevant true alert or non-interesting alert); and the sensor has incorrectly identified an event as an attack giving a false alert (non- interesting alert). An IDS produces an alert after detecting a malicious activity and then alerts are forwarded to the analysts for analysis [9]. Usually, analysts use their knowledge to distinguish true alerts from false alerts, a task that could be frustrating and time consuming when dealing with huge volumes of alerts. After identifying the true alerts, the analysts may investigate the intrusion, identify and fix the problem causing the alert, and or modify IDS signatures to filter out the non-interesting alerts.

III. RELATED WORKS

Alert Classification is one of the practical techniques studied for many years to manage alerts in order to handle the alert load efficiently. Generally, the classification methods label alerts as true or false alerts.
Clifton and Gengo [6] used data mining techniques to identify sequences of alarms that result from normal behavior, enabling construction of filters to eliminate those alarms. They have investigated the detection of frequent alert sequences, in order to use this knowledge for creating IDS alert filters.

Julisch and Dacier [7] proposed a conceptual clustering technique to show that intrusion detection alarms can be handled efficiently. During their experiments, they reduced the number of alerts by an average of 75% [7]. This work was later extended by Julisch who reported the reduction of alerts by 87% [2].

Pietraszek [8] proposed an Adaptive Learner for Alert Classification (ALAC) as a new system for reducing false positives. ALAC is an adaptive alert classifier based on the feedback of an intrusion detection analyst and machine learning techniques. With this system, the number of false alerts was reduced by more than 30%. This system has a disadvantage: during a system's lifetime the size of the training set grows infinitely. Later, Pietraszek extended his work and presented two complementary approaches for false positives reduction: CLARAty which is based on alert post processing by data mining, root-cause analysis and ALAC which is based on machine learning. CLARAty is an alert-clustering approach using data mining with a modified version of attribute-oriented induction [8]. Using this system, the number of alerts handled has been reduced by more than 50%.

Al-Mamory et al. have provided a survey on alert processing techniques [9], later they proposed a data mining alert clustering technique that groups alarms whose root causes are generally similar and finds generalized alarms which helps the human analyst to design and write filters [9,10]. During their experiments, the averaged reduction ratio was about 82% [9], 93% [10] and 74% [10] of the total alarms. Their method can be considered as a variation of Julisch's work; however, they have designed a new data mining technique, which is different in clustering methods.

Tian et al. [12] have used pattern mining method to develop an adaptive alert classifier that classifies alerts in true positives and false positives classes and learns knowledge adaptively by the feedback of the operators.

Sabri et al. [13] used data mining to extract the useful information from large databases. They have used the KDD CUP 99 dataset to evaluate their method. The results show that the data mining technique reduces the false alarms rate and increase the accuracy of the system.

IV. PROPOSED MODEL

In our proposed model we focus on alert classification using pattern matching to distinguish real attacks from false positives and low importance events. Figure 2 shows the proposed model. In our system we use binary traffics files of a network KDD 99 dataset [6] instead of real network traffics. The KDD 99 dataset is the most used for evaluating IDSs according to [18]. Snort tool [8] is used to produce alerts of KDD 99 dataset network traffics. Snort is an open source signature based IDS which gets KDD 99 online traffics and then generates alert log files; these files are inputs of our system.

A Statefull Pattern Matching algorithm is used to filter alerts as normal or attacks. Pattern matching looks for a fixed sequence of bytes within a single packet; its deployment is Simple. To filter traffic inspection, the pattern is also usually associated with a particular service and source or destination port. An example of pattern matching is firing an alarm if the packet is Internet Protocol version 4 (IPv4) and User Datagram Protocol (UDP), it has destination port 1458, and it contains the string "madison" in the payload. Statefull pattern matching adds to pattern-matching by searching for unique sequences that might be distributed across several packets within a stream. Statefull pattern matching could improve on the preceding example by firing an alarm if the string "mad" is detected in one packet and "ison" is detected in a subsequent packet.

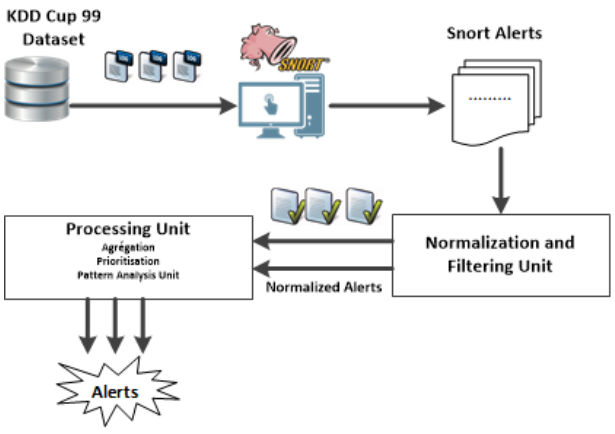

Figure 1. Proposed Alert Management System

To process data alerts, we first need a consistent method for gathering data. The support of several output formats [19] makes Snort more flexible in the formatting and presentation of output to its users. The output modules are run when the alert or logging subsystems of Snort are called, after the preprocessors and

detection engines. Our system supports the Snort's outputs modules listed below:

- **alert_syslog**: sends alerts to the syslog facility.
- **alert_fast**: prints Snort alerts in a quick one-line format to a specified output file.
- **alert_full**: prints Snort alert messages with full packet headers.
- **CSV plugin:** allows data alert to be written in a format easily importable to a database

Normalization and Filtering Unit

Labeling unit gets generated alert from Snort. A labeled alert is an alert with its own attack type. The Snort list files contain information about all packets in DARPA 99 dataset.

Processing Unit

In this phase, accepted attack types are entered to the processing unit and only alerts that are selected in class of predefined attack types [11, 12 and 14]. For filtering alerts, this unit uses eight attributes which are: Signature ID, Signature Rev, Priority, Source IP, Destination IP, Source Port, Destination Port, and Protocol.

The program does the following:

- Initializes the program
- Processes the configuration and data files
- Begins the main loop and reads the first alert data file
- Extracts and records details of each alert, this details are: Signature ID, Signature Rev, Priority, Source IP, Destination IP, Source Port, Destination Port, and Protocol .
- Classify alerts into 5 types :

  - ✓ Distribution of alerts according to the attacks
  - ✓ Classifying alerts by Number of events from same Source IP to same target IP with the same attack
  - ✓ Classifying alerts by Percentage of events from a source IP address to a destination IP address.
  - ✓ Classifying alerts by Percentage of events from one IP source to any destination IP address with the same attack.
  - ✓ Classifying Alerts by Percentage of events to one specific target.

In our experimental we used KDD cup 99 with snort to generate alerts from a different IP source address to be monitored on our result. The tables below present experimental results with 206 Alerts that contain 4 different attempted attacks (Table 1).

Attack Name	Number of alerts	Rate compared to total alerts
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	101	49.03 %
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	100	48.54 %
Consecutive TCP small segments exceeding threshold	4	1.94 %
ET SCAN Grim's Ping ftp scanning tool	1	0.49 %

Table 1: Distribution of alerts according to the attacks & Rate of each attack compared to total alerts

Attack Name	To	Rate compared to total alerts
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	172.16.183.2	49.03 %
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	205.188.248.25	18.93 %
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	205.188.248.57	16.02 %
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	205.188.248.89	13.59 %
Consecutive TCP small segments exceeding threshold	62.57.114.246	1.94 %
ET SCAN Grim's Ping ftp scanning tool	172.16.183.2	0.49%

Table 2: Classifying alerts by Percentage and number of events to one certain host

Attack Name	From	Rate compared to total alerts
(http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE	172.16.183.2	48.54 %
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	205.188.248.25	18.93 %
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	205.188.248.57	16.02 %
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	205.188.248.89	13.59 %
Consecutive TCP small segments exceeding threshold	172.16.183.2	1.94 %
ET SCAN Grim's Ping ftp scanning tool	80.14.184.201	0.49 %
(http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE	205.188.248.6	0.49 %

Table 3: Classifying alerts by Percentage and number of events from one soirce IP to any with same Atack signature.

The experimental results show that the proposed algorithm is able to treat thousands of alerts in real time, classify theme into different categories. Although our preliminary results are promising, we found that when using a large log files, the execution time rises by a few seconds compared to other proposed algorithms, this problem is due to code optimization of the algorithm. So in the next step we will focus this matter to improve the algorithm performances.

V. CONCLUSION AND FUTURE WORKS

In This paper we have proposed a New Alert Management System for Snort IDS based on Pattern Matching. The management system is able to classify alerts by their importance and reduces number of false alerts considerably; also the system has the capability to identify with more accuracy the attack types of the alerts. The implementation of such management systems will contribute to the increase of the network resilience against security attacks while leading to the optimisation of the human resources usage. In the next step we will focus on the risk assessment for the IDS alerts, so this idea is the future work of this paper.

REFERENCES

[1] Tu Hoang Nguyen "Improving the management of IDS alerts" International Journal of Security and Its Applications Vol.8, No.3 (2014), pp. 393-406 http://dx.doi.org/10.14257/ijsia.2014.8.3.38

[2] K. Julisch, "Clustering intrusion detection alarms to support root cause analysis", ACM Trans. Inf. Syst. Secur. 6, 2003 K.

[3] Lorenzo De Carli et.al "Beyond Pattern Matching: A Concurrency Model for Stateful Deep Packet Inspection", CCS '14 Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security Pages 1378-1390

[4] C. Xiang, P.C. Yong, L.S. Meng, "Design of multiple-level hybrid classifier for intrusion detection system using bayesian clustering and decision trees", Pattern Recognition Letters 29, 2008.

[5] S. Lee, G. Kim, S. Kim, "Self-adaptive and dynamic clustering for online anomaly detection", Expert Systems with Applications 38, 2011

[6] C. Clifton, G. Gengo, "Developing custom intrusion detection filters using data mining", MILCOM 2000. 21st Century Military Communications Conference Proceedings, 2000

[7] K. Julisch, M. Dacier, "Mining intrusion detection alarms for actionable knowledge", in: The 8th ACM International Conference on Knowledge Discovery and Data Mining, 2002

[8] T. Pietraszek. "Using adaptive alert classification to reduce false positives in intrusion detection," in Proc. of RAID Symposium, 2004.

[9] S.O. Al-Mamory, H. Zhang, "A survey on IDS alerts processing techniques", 6th WSEAS International Conference on Information Security and Privacy, Tenerife, Spain, 2007

[10] S.O. Al-Mamory, H. Zhang, "IDS alarm reducetion using data mining ", IEEE International Conference on Neural Networks, 2008.

[11] R. Vaarandi, "Real-time classification of IDS alerts with data mining techniques", in Proc. of MILCOM Conference, 2009.

[12] Z. Tian, W. Zhang, J. Ye, X. Yu, H. Zhang, "Reduction of false positives in intrusion detection via adaptive alert classifier", IEEE International Conference on Information and Automation, 2008

[13] F.N. Sabri, N.M. Norwawi, K. Seman, "Identifying false alarm rates for intrusion detection system with Data Mining", IJCSNS International Journal of Computer Science and Network Security, VOL.11, 2011

[14] Curry, D., & Debar, H. (2007, March). "Intrusion Detection Message Exchange Format." Retrieved July 7, 2008, from IETF: http://www.rfc-editor.org/rfc/rfc4765.txt

[15] Zhai, Y., Ning, P., & Xu, J. (2005). "Integrating IDS alert correlation and OS-level dependency tracking". North Carolina: North Carolina State University.

[16] Christopher Kruegel "Intrusion Detection and Correlation, Challenges and Solutions". Book on Advances in Information Security (Volume14). ISBN:0-387-23398-9. Springer.

[17] K. Timm, "Strategies to reduce false positives and false negatives in NIDS", Security Focus Article, available Online at: http://www.securityfocus.com/infocus/1463, (2009).

[18] V. Engen, J. Vincent, and K. Phalp, "Exploring Discrepancies in findings obtained with the KDD Cup 99 data set", Intelligent Data Analysis, Vol.15, pp. 251-276, April 2011.

[19] James P. Anderson. "Computer security threat monitoring and surveillance". Technical report, James P. Anderson Co, 1980.

[20] Dorothy E. "Denning. An intrusion detection model". IEEE Transactions on Software Engineering, SE-13(2):222–232, 1987.

[1] W. Robertson and W. K. Robertson, "Alert verification determining the success of intrusion attempts", The Proceedings of the Detection of Intrusions and Malware and Vulnerability Assessment, Dortmund, Germany, (2004) July 6-7, pp. 25-38.