Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 5, May 2016, pg.567 – 575

Efficient Technique for Non-Critical Alarm Reduction in IDS

Miss. Priyanka Bhokre[1], Mr. S.G.Vaidya[2]
M.E Student, Assistant Professor
Department of Computer Engineering
SYCET College, Aurangabad, Maharashtra India
1. priyankabhokre6@gmail.com, 2. Swapnil.vaidya@sycet.org

Abstract: In this review paper we are discussing the Network Intrusion Detection System. In this there are number of signature -based network intrusion detection systems as defending for different types of attack. However during the detection process large numbers of alarms are generated, which are the Non-Critical alarms. Hence therefore the effectiveness of the system greatly reduces and also increases difficulty in analysis work-done of IDS. The reason behind that is the detection process of a signature based IDS is only depends on its signature and todays IDS signatures are lack of contextual information related to actual system. Moreover the existing traditional signature matching technique is a key factor which limits the IDS; in this the burden of processing is at least linear to the input string size. In this paper, we propose a novel scheme for the hash based contextual signatures which combine original intrusion detection signatures with contextual information and hash function to identify and filter out non-critical alarms. By using this we can construct adaptive hash based non critical alarm filter which improves he performance of previous system. Moreover, we indicate that our scheme is compatible with the different types of intrusion detection signatures. In the evaluation, we develop our novel scheme to a specific implementation and explore its performance in experimental settings.

Keywords-Intrusion detection; Network security; Non-critical alarm filter; Context-based system, function; contextual signature

---

I. INTRODUCTION

Today, intrusions are being a challenge to the network security environment. To solve this problem, an intrusion detection system (IDSs) has been widely deployed in various types of network environments. Intrusion detection is the method of analysing different computer system events or network events in order to find out the signs of possible events that violates and harms the pre-defined security policies of system (such as malware, unauthorized

access to the systems and users' misbehaviour in the system). Intrusion detection system (IDS) is a tool that has capability to protect our network systems from being attacked by the intruders.

Traditionally, there are two main important types of intrusion detection system: 1) signature based IDS [4, 9] and 2) anomaly-based IDS [10, 12]. The signature-based IDS) which recognizes an attack by comparing current system or/and network events with its signatures. The detection ability of signature- based IDS is depending on its signature ability (i.e., the no. of signatures), therefore, this type of detection systems can only use to detect known attacks. Next the anomaly-based IDS find out an attack by analysing great deviations between current events with its normal event profiles. Based on the detection technique, the benefit of anomaly-based IDS is that identifying unknown attacks also. However in real time environment, the signature-based approach is mostly used as compared to the anomaly-based approach.

Problem: However an intrusion detection system is detecting different types of attacks, but a big suffering problem is that , in the both the signature based and anomaly based IDS , there are large number of alarms are getting, specifically non-critical alarms are generated during their detection procedure.

Definition of non-critical alarms: A non-critical alarm is either not relevant to a malicious activity or not related to a successful true attack. That means, a non-critical alarm is either a false positive or a non-relevant positive.

II. BACKGROUND AND RELATED WORK

Now days IDS area unit wide used for numerous pc networks about to find all types of attacks. However, the main drawback is that variety of non-critical alarms area unit get generated throughout detection, although the attack isn't happen really. This will increase analysis work and decreases effectiveness of systems. Non-critical alarms area unit happened as a result of lack of discourse data thus to get rid of this disadvantage I even have propose design of context supported crucial alarm filter [2]. My alarm filter contains categorization element to link input alarms to corresponding discourse data. There are a unit 2 engines one is Analysis Engine that helps to separate non-critical alarms in step with discourse data another i.e. Monitor engine to update index values. This module implements functions to manage the cluster, merge and correlate alerts. The cluster and merging Functions recognize alerts that correspond to the same occurrence of the attack create a new alert that contains various those various alerts. Practical shows that these functions help us to reduce number of alerts I have also observed that alerts are elementary and can be manage by Security Administrator. Thus, the function can create global synthetic alerts

A. BYTE-LEVEL NETWORK INTRUSION DETECTION SIGNATURES WITH CONTEXT

In the traditional signature format was contains a sequence of bytes that was specifically represents a specific attack. If this sequence is found in the packet payload, then this is the indicator of a possible attack. Since, the matcher is a central part of any signature-based NIDS. However there are many NIDSs which allow only fixed strings to search patterns, and utilize it using regular expressions. Regular expressions have many different advantages: a. they are more flexible than the fixed strings. b. And their representation has made them a well-known tool in many applications, and their power arises in part by providing additional syntactic context with which to sharpen textual searches.

1) Signature Language

Snort's signatures are pervasive, free and automatically updated. Hence therefore, we tend to are notably fascinated by changing them into our signature language. It seems to be rather difficult to implement an entire computer programme for Snort's language. As we've got ready to confirm, its syntax and linguistics aren't absolutely

documented, and in real typically solely outlined by the ASCII text file. More-ever, as a result of completely different structure of Bro and Snort, it is generally unbearable to stay the precise representations of the signatures. As the example in Figure two shows, the signatures area unit outlined by associate symbol and a collection of attributes. There are unit 2 main types of attributes: a. conditions and b. actions. The conditions perform once the signature matches, whereas the actions declare what to do after the case of a match. Additionally conditions will be divided into four sections: a. header, b. content, c. dependency, and d. context.

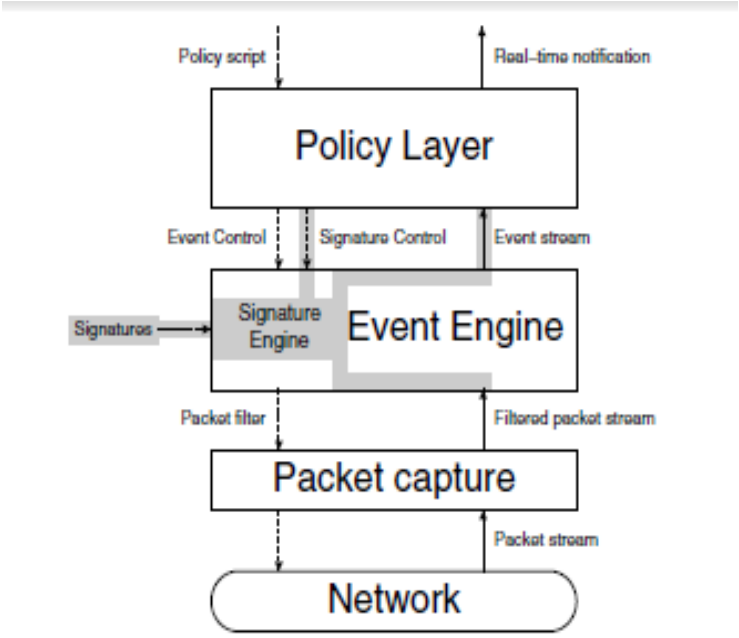

Fig 1. Integrating the signature engine

```
alerttcp any any -> [a.b.0.0/16,c.d.e.0/24] 80
( msg:"WEB-ATTACKS conf/httpd.conf attempt";
nocase; sid:1373; flow:to_server, state established;
content:"conf/httpd.conf"; [...] )
```

(a) Snort

```
signature sid-1373 {
ip-proto == tcp
dst-ip == a.b.0.0/16,c.d.e.0/24
dst-port == 80
# The payload below is actually generated in a
# case-insensitive format, which we omit here
# for clarity.
payload /.*conf\/httpd\.conf/
tcp-stateestablished,originator
event "WEB-ATTACKS conf/httpd.conf attempt"
}%
```

(b) Bro

Fig2. Example of signature conversion

2) The Power of Bro Signatures

First, we demonstrate how to define more "robust" signatures by using regular expressions. Then, we show how to identify false attack attempts by analysing the application a particular server is running (this is known its vulnerability profile) as well as the response of the server. Using regular expressions and vulnerability Profiles, Request/Reply Signatures Attacks with Multiple Steps, exploit scanning.

B. ALERT VERIFICATION FINDING THE SUCCESS OF INTRUSION ATTEMPTS

Intrusion detection systems analyse the network events and identifies malicious activity in the system. When an attack is identified, an alert is generated. A perfect intrusion detection system has capability to identify all types of attacks without raising any false alarms or non-relevant alarms. In addition, an analysis work would be executed only when an attack is successful. Unfortunately number of false alarms occur in intrusion detection systems, and therefore these events are treated as malicious and harmful to the system. These are alerts related with attacks that were not successful attacks or false attacks. Such alerts should be labeled accurately so that they have lowest priority. This paper presents the different issues involved in alert verification process and presents a method performing the real-time verification of attacks identified by IDS. The experimental evaluation of such a method shows that verification process can greatly reduce both false and non-relevant alarms.

Alert Verification

Alert verification is the process of verification of successful attacks. That is, given an attack (and its corresponding alert produced by an IDS), it is the process of verification as determine whether detected attack has successful or not. There are number of techniques that can be used to perform this verification. One technique is used to compare the configuration of the attacked system(e.g., OS, running applications, service version) to the requirements for a successful attack or a true attack.

1) Passive

The benefit of passive mechanisms is that the incontrovertible fact that they are doing not interfere with the traditional operation of the network. A demerit of passive mechanism is potential differences between the state stored in the knowledge base and the actual network security status. New services may be installed or the firewall rules may be changed without updating the knowledge base.This can be lead to attack that are labeled as non-relevant, even though a vulnerable target exists.

2) Active

Active alert verification mechanisms don't depends on a priori gathered information. Instead, the verification method actively initiates the knowledge gathering method once when an alert is received. Active alert verification has the benefit, that the knowledge is current. This enables one to assess the standing of the target host and therefore the attacked service Associate to acknowledge changes at the victim host that function a sign of an attack.

3) Implementation

In implementation comprises Associate in nursing addition to Snort's alert process pipeline that intercepts alerts to be passed to enable alert plug-ins. These alerts area unit queued for verification by a pool of verification threads. This enables Snort to continue process events whereas alert verification takes place within the background. A summary of the design of this implementation is delineated in Figure 3.

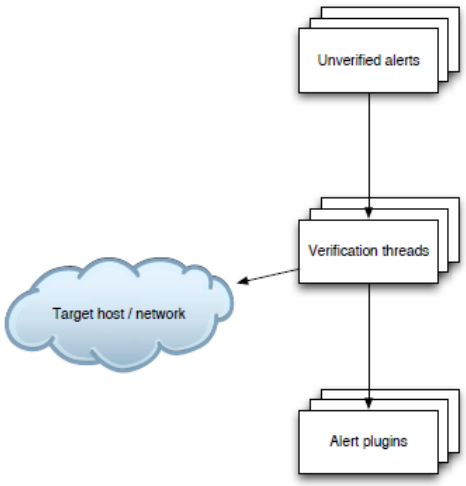

Fig3: Snort Alert Verification Architecture

Additionally, the Snort rule language was extended to incorporate new keywords to perform rhetorical checks at the target hosts. This can be not a demand of active verification, however, and it might even be potential to own one verification system that receives alerts via the network from multiple sensors. During this case, the alert verification tool may be integrated into the alert assortment framework. As a result of we have a tendency to wish to permit the complete use of Snort with the verification enhancements, the deployment of multiple Snort sensors would implement that many verification modules area unit are running. This could be no downside; as a result of the performance impact of the verification tool is low.

C. ALERT CORRELATION IN A COOPERATIVE INTRUSION DETECTION FRAMEWORK

In this paper there is a function to manage, cluster, merge and correlate alerts. The two functions clustering and merging analyse alerts that correspond to the same occurrence of an attack and create a new alert that merge data contained in these various alerts. The aim of the correlation function is thus to generate global and synthetic alerts. This paper presents the approach we suggest to design this function.

This paper developed a cooperation module called CRIM module, which consist of three functions:

1) Alert Clustering:

The clustering function can then have an access to this database and generates alert clusters. When an attack found, the IDS connected to CRIM may generate many alerts for this attack. The clustering function attempts to identify the alerts that actually correspond to the same occurrence of an attack. These alerts are then put into a cluster.

2) Alert Merging:

In this each cluster is then sent to the alert merging function. This function creates a new alert for each cluster that is representative of the information contained in the various alerts belonging to the same cluster.

3) Alert Correlation:

This function used to analyse the cluster alerts provided as outputs by the alert merging function. Here have been observed that the merging function generally provides many elementary alerts. The motto of the correlation function is thus to correlate alerts to provide the security administrator with synthetic information.

III. PROPOSED SYSTEM

Using hash function, our proposed hash-based contextual signature is next used to construct adaptive reputation based contextual and hashed non-critical alarm filter. The proposed system can be represented as by 3 tuple {CI, Sig, H}. Where CI represents contextual information in which consist of networking features, target configuration and non-relevant alarm content. Sig represents IDS signatures such as Snort signature format. And H represents hash function (MD5). The construction of this filter is shown in fig 4. below.

Based on the following construction, here in this filter consist of three parts namely,

1). IP-based Index Hash Table: Used to classify incoming alarms into different comparison table.

2). Matched Contextual Information Table: Contains hashed contextual information which is used to match information in the detection and indexed by IP addresses.

3). All Stored Contextual Information Table: Contains all available and also hashed contextual information used in the process of matching the contextual information.

The application of a hash function is as below:

1) Hashing IP sources as index parameters can speed up the procedure of alarm classification.

2) Hashing the contextual information of both Matched Contextual Information Table and All Stored Contextual Information Table into fixed length strings can greatly enhance the matching ability like speed between incoming alarms and contextual information [18].

All originated IDS alarms will firstly arrive at IP-based Index Hash Table and this table may perform two actions:

1) IP is hashing beneath conditions: if the hashing price is no longer than the length of IPs like thirty-two bits, then hashing the IP supply address by exploitation the chosen hash function . Other else don't hash the IPs and use the first IPs for compartmentalization.

2) The incoming IDS alarms to totally different comparison tables consistent with IP-based index.

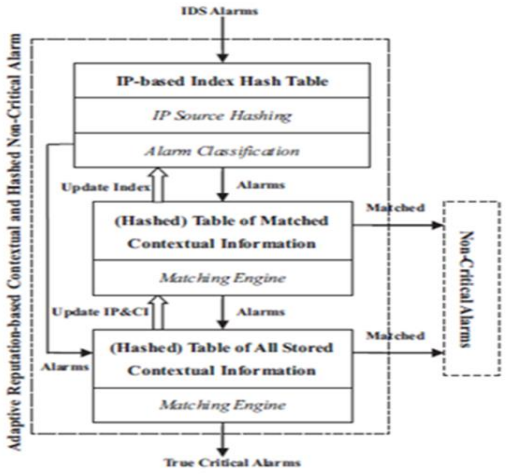

Fig4: Adaptive hash-based contextual non-critical alarm filter

Based on the index info, all incoming IDS alarms are classified as below [22,18]:

1) For the Associate in Nursing IDS alarm, if its supply IP address is within the Table of Matched discourse info, then this alarm are sent to the current table

2) For the Associate in Nursing IDS alarm, if its supply IP address is not within the Table of Matched discourse information, then this alarm are sent to the Table of All keep discourse info.

IV. EXPECTED OUTCOME

To explore the practical performance of our adaptive contextual hash based non-critical alarm filter ,we further implemented the alarm filter in the network environment for non-critical alarm filtration. Moreover, we evaluate our method by using simulated real network traffic. The filtration performance of our method is shown in Fig 5. below. The results show that our method (Filter 1) can greatly reduce the total number of alarms by filtering out non-critical alarms and demonstrate that our method is effective as compared to other method (Filter 2 & Filter 3) in our deployed environment. The Performance of Filter 2 (Hash-based contextual non-critical alarm filter) and Filter 3(Contextual non-critical alarm filter) are the existing systems are compared with the performance of Filter1 (Adaptive Hash-based contextual non-critical alarms filter).

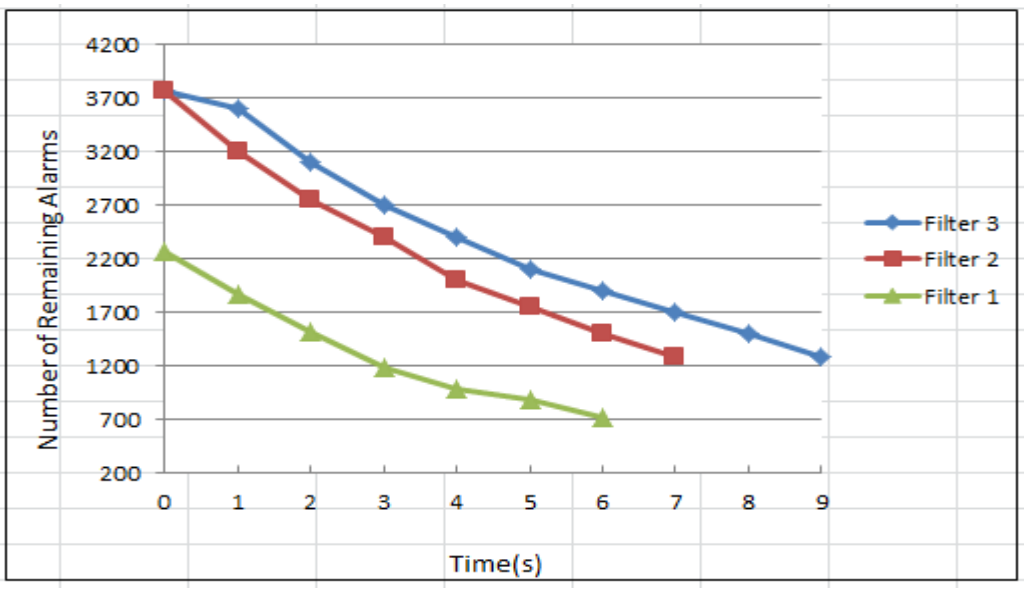

Fig5: Performance Evaluation

V. CONCLUSION

Non-critical alarms square measures enormous challenges for intrusion detection systems, which may greatly minimizes the effectiveness of the system and heavily increase the analysis burden on analysing the generated IDS alarms. To mitigate this downside, we tend to advocate that combining original intrusion detection signatures with discourse data may be a promising approach. On the opposite hand, the standard signature matching may be a key limiting issue for IDSs during which the process burden is a minimum of linear to the dimensions of an input string.

During this paper, we tend to plan a completely unique theme of hash-based discourse signatures that mixes the first intrusion detection signatures with not solely discourse data however additionally hash functions. We tend to summarize the generic discourse signatures as 2- tuple, whereas our planned theme may be portrayed employing a 3-tuple. By employing a hash perform, our theme will any change the illustration and cut back the matching burden compared with the standard discourse signatures. Moreover, our theme may be any accustomed construct an accommodative reputation-based discourse and hashed non-critical alarm filter which will improve the performance (e.g., speed) of existing discourse signatures in filtering out non-critical alarms. The discourse data may be extracted from vulnerability databases and from knowledgeable data.

Future work might embrace exploring alternative hash functions and conducting a study to analyse the result of collision on the performance. additionally, future work might additionally embrace investigation alternative matching approaches in our theme and exploring the performance of mixing alternative matching ways like longest prefix match and regular expression. We tend to additionally decide to develop a wide offered benchmark for examination and evaluating totally different approaches in this area.

REFERENCES

[1] Y. Meng, L.F. Kwok, A generic scheme for the construction of contextual signatures with hash functions in intrusion detection, in: Proceedings of International Conference on Computational Intelligence and Security (CIS), 2011, pp. 978–982.

[2] V. Paxson, Bro: a system for detecting network intruders in real-time, Computer Networks 31 (23–24) (1999) 2435–2463.

[3] K. Scarfone, P. Mell, Guide to Intrusion Detection and Prevention Systems (IDPS), NIST Special Publication 800-94, 2007.

[4] P.A. Porras, R.A. Kemmerer, Penetration state transition analysis: a rule-based intrusion detection approach, in: Proceedings of the 8th Annual Computer Security Applications Conference (ACSAC), pp. 220–229, 1992.

[5] J. McHugh, Testing intrusion detection systems: a critique of the 1998 and1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory, ACM Transactions on Information System Security (2000) 262–294.

[6] P. Ning, D. Xu, Learning attack strategies from intrusion alert, in: Proceedings of the 2003 ACM Conference on Computer and Communications, Security, 2003, pp. 200–209.

[7] S. Axelsson, The base-rate fallacy and the difficulty of intrusion detection, ACM Transactions on Information and System Security 3 (3) (2000) 186–205.

[8] 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, M.A. Zissman, Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation, in: Proceedings of the DARPA Information Survivability Conference and Exposition 2000, pp. 12–26.

[9] M. Roesch, Snort: lightweight intrusion detection for networks, in: Proceedings of the Usenix Lisa Conference, 1999, pp. 229–238.

[10] A.K. Ghosh, J. Wanken, F. Charron, Detecting anomalous and unknown intrusions against programs, in: Proceedings of Annual Computer Security Applications Conference (ACSAC), 1998, pp. 259–267.

[11] Snort, The open source network intrusion detection system. Homepage:<http://www.snort.org/>.

[12] D. Wagner, P. Soto, Mimicry attacks on host-based intrusion detection systems, in: Proceedings of ACM conference on Computer and communications security (CCS), 2002, pp. 255–264.

[13] F. Gagnon, F. Massicotte, B. Esfandiari, Using contextual information for IDS alarm classification, in: Proceedings of the 6th Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), 2009, pp.147–156.

[14] R. Sommer, V. Paxson, Outside the closed world: on using machine learning fornet work intrusion detection, in: Proceedings of IEEE Symposium on Security and Privacy, 2010, pp. 305–316.

[15] T.H. Ptacek, T.N. Newsham, Insertion, evation, and denial of service: eludingne twork intrusion detection, Technical Report, Secure Networks, January 1998.

[16] Bro: The powerful network analysis framework.

[17] Packet generator: Colasoft packet builder. Homepage.

[18] R. Lippmann, S. Webster, D. Stetson, The effect of identifying vulnerabilities and patching software on the utility of network intrusion detection, in: Proceedings of Recent Advances in Intrusion Detection (RAID), 2002, pp. 307–326.

[19] R. Sommer, V. Paxson, Enhancing byte-level network intrusion detection signatures with context, in: Proceedings of ACM Conference on Computer and Communications, Security (CCS), 2003, pp. 262–271.

[20] M. Cost, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang, P. Barham, Vigilante: end-to-end containment of internet worms, in: Proceedings of ACM Symposium on Operating System Principles (SOSP), 2005, pp. 133–147.

[21] D. Brumley, J. Newsome, D. Song, H. Wang, S. Jha, Towards automatic generation of vulnerability based signatures, in: Proceedings of IEEE Symposium on Security and Privacy, 2006, pp. 2–16.

[22] F. Massicotte, Y. Labiche, L.C. Briand, Toward automatic generation of intrusion detection verification rules, in: Proceedings of Annual Computer Security Applications Conference (ACSAC), 2008, pp. 279–288.

[23] H. De bar, A. Wespi, Aggregation and correlation of intrusion-detection alerts In: Proceedings of Recent Advances in Intrusion Detection (RAID), 2001, pp.85–103.

[24] C. Kruegel, W. Robertson, Alert verification: Determining the success of intrusion attempts, in: Proceedings of International Conference on Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), 2004, pp.25–38.

[25] F. Cuppens, A. Miege, Alert correlation in a cooperative intrusion detection framework, in: Proceedings of IEEE Symposium on Security and Privacy, 2002,pp. 202–215.

