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

IJCSMC, Vol. 5, Issue. 1, January 2016, pg.300 – 307

Efficient Technique for Non-Critical Alarm Reduction in IDS: A Review

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 earlier Intrusion Detection System. In this there are number of signature based intrusion detection systems which are widely used in Network Intrusion Detection Systems as defending for different types of attack. However large numbers of alarms are generated during the detection process, which are the Non-Critical alarms. Because of this the effectiveness of the system greatly decreases and also increases difficulty in analysis work for IDS. The reason behind that the detection capability of signature based IDS is only depends on its signature and todays IDS signatures are lack of contextual information related to actual system. Also the traditional signature matching technique is a factor which limits the IDS, in this the burden of processing is at least linear to the size of an input string. To further facilitate the analysis of attack ways, which is essential to many security applications like portable computer, network forensics and incident handling, this paper presents different techniques to deal with different types of attacks.

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

I. INTRODUCTION

Now days, intrusions are being a big challenge to network security environment. To avoid this problem, an intrusion detection system (IDSs) has been widely developed in various network environments. Intrusion detection is the technique of analysing different computer system events or network events in order to find out the measures of possible events that violates the pre-defined security policies of system (such as malware, unauthorized access to systems and users' misbehaviours).Intrusion detection system (IDS) is a tool that has ability to protect our network systems from being attacked.

Traditionally, there are two most important types of intrusion detection system: a) signature based IDS [4, 9] and b) anomaly-based IDS [10, 12]. The signature-based IDS) which identifies an attack by comparing current system or network events with its signatures. The detection ability of signature- based IDS is depending on its signature capability s (i.e., the number of signatures), therefore, this type of detection systems can only detect known attacks. And the anomaly-based IDS detect an attack by finding great deviations between current events and its normal event profiles. Depending on the detection technique, the advantage of anomaly-based IDS is that identifying unknown attacks. In real time, the signature-based approach is mostly used than the anomaly-based approach.

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

II.BYTE-LEVEL NETWORK INTRUSION DETECTION SIGNATURES WITH CONTEXT

Usually in a traditional signature format contains a sequence of bytes that represents a specific attack. If this sequence is found in the packet payload, this is an indicator of a possible attack. Because of, the matcher is a central part of any signature-based NIDS. However many NIDSs only allow fixed strings to search patterns, and we argue for the utility of using regular expressions. Regular expressions have many different merits: they are most flexible than fixed strings. And their expressiveness has made them a well-known tool in many applications, and their power arises in part from providing additional syntactic context with which to sharpen textual searches.

A. Signature Language

Snort's signatures are comprehensive, free and often updated. Therefore, we tend to are notably fascinated by changing them into our signature language.

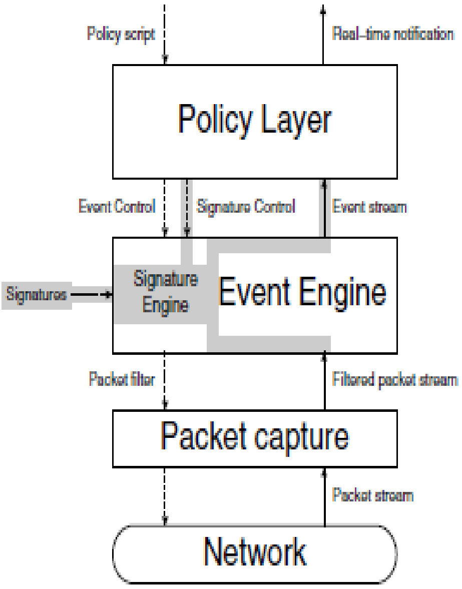
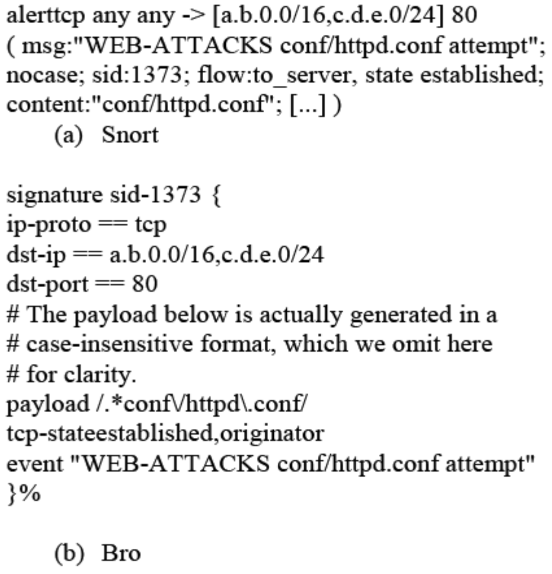


Fig 1. Integrating the signature engine



```
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

It seems to be rather tough to implement an entire computer programme for Snort’s language. As way as we’ve got been ready to confirm, its syntax and linguistics aren't absolutely documented, and in truth typically solely outlined by the ASCII text file. Additionally, as a result of completely different internals of Bro and Snort, it\'s generally insufferable to stay the precise linguistics of the signatures. As the example in Figure two shows, our signatures area unit outlined by means that of associate symbol and a collection of attributes. There are a unit 2 main sorts of attributes: (i) conditions and (ii) actions. The conditions outline once the signature matches, whereas the actions declare what to try to within the case of a match. Additionally conditions will be divided into four sections: header, content, dependency, and context.

B. The Power of Bro Signatures

First, we demonstrate how to define more “tight” signatures by using regular expressions. Then, we show how to identify failed attack attempts by considering the set of software a particular server is running (we call this its vulnerability profile and incorporate some ideas from [22] here) as well as the response of the server. Using Regular Expressions, Vulnerability Profiles, Request/Reply Signatures Attacks with Multiple Steps, Exploit Scanning

III. CONSTRUCTION OF CONTEXTUAL SIGNATURES USING HASH FUNCTION IN INTRUSION DETECTION

Intrusion detection system (IDS) [8] is one type of tool that has capacity to protect our network and systems from being attacked. The main function of this tool is to inform security officers when a network or computer system is violating and to record related events simultaneously.

The network-based intrusion detection systems again have two types as misuse detection and anomaly detection based. The detection ability of signature- based IDS(misuse detection) is depending on its signature capability s (i.e., the number of signatures), therefore, this type of detection systems can only detect known attacks. And the anomaly-based IDS detect an attack by finding great deviations between current events and its normal event profiles. Depending on the detection technique, the advantage of anomaly-based IDS is that identifying unknown attacks. In real time, the signature-based approach is mostly used than the anomaly-based approach. The accuracy of signature-based detection is heavily depends on its signatures, which are extracted from the detected attack traffic.

Problem: A big suffering problem of IDS(both type) is that a large number of alarms are generated during the detection processes, in which mostly are non-critical alarms. While it is hard for IDS to analyse the situation and the state of an attack attempt [4], it has to forward all detected attack events to the security officer so as to solve security risk. This is the biggest challenge and a heavy work burden for an administrative officer to investigate the real attacks after discarding all non-critical alarms [2].

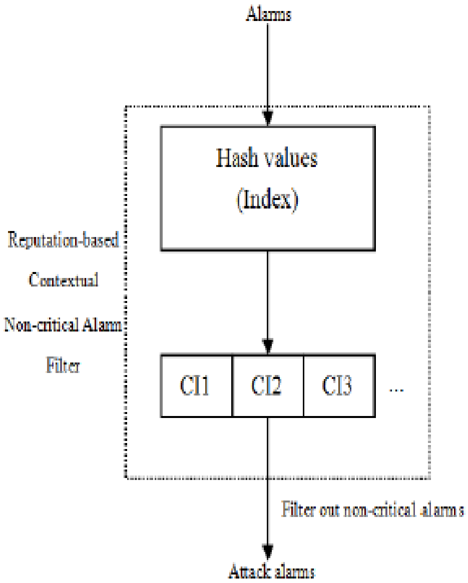


Fig3. The construction of reputation-based contextual non-critical alarm filters by adding hash function.

In this technique, they propose a generic scheme for the construction of hash based contextual signatures to minimize the number of non-critical alarms. However, the use of hash function has two advantages: one is reducing the time for matching the strings and improving the quality of signature representation; another merit is using hash index value to extend the utilization of contextual signatures to form a hashed-based contextual non-critical alarm filter. Our scheme summarizes a representation of traditional contextual signatures at high level and it is compatible to different other representations of ID signatures. In evaluation, we implement a concrete contextual signature depending on our generic scheme. The experimental results show that this scheme is effective and efficient in the network environment.

IV. ALERT VERIFICATION FINDING THE SUCCESS OF INTRUSION ATTEMPTS

Intrusion detection systems monitor the network events and attempt to detect malicious activity. When an attack is identified, an alert is produced. A perfect intrusion detection system has capacity to identify all types of attacks without raising any false alarms. In addition, an analysis work would be executed only when an attack is successful attack. Unfortunately false alarms occur in intrusion detection systems, and therefore events are treated as malicious. Also in addition, non-relevant alerts are also produces. These are alerts related with attacks that were not successful attacks. Such alerts should be tagged accurately so that they have lower priority. This paper states the different issues involved in alert verification process and presents a tool performing real-time verification of attacks detected by IDS. The experimental evaluation of such a tool shows that verification of attack can greatly reduce both false and non-relevant alarms.

A. Alert Verification

Alert verification is the process of verifying the successful attacks. That is, given an attack (and its corresponding alert produced by an intrusion detection system), it is the task of verification process to determine whether detected attack has succeeded or not. There are number of techniques that can be used to perform this verification. One method is to compare the configuration of the attacked machine (e.g., OS, running services, service version) to the requirements for a successful attack. When a machine is not vulnerable for an attack (if the configuration does not match all the requirements) then the alert can be said as failed.

Alert verification is outlined because the method of verification of the success of attacks. That is, given associate degree attack it\'s the task of the alert verification method to see whether or not this attack has succeeded or not. One chance is to match the configuration of the victim machine (e.g., OS, running services, service version) to the necessities of a winning attack. Once the victim is not liable to a specific attack (because the configuration doesn't satisfy the attack requirements), then the alert are often labelled as unsuccessful. Another chance is to model and analyse the expected "outcome" of attacks.

1) Passive

The advantage of passive mechanisms is that the incontrovertible fact that they are doing not interfere with the traditional operation of the network. Additionally, it\'s not necessary to perform extra tests that delay the notification of directors or the beginning of active countermeasures. A drawback of passive mechanisms square measure potential variations between the states hold on within the content and therefore the actual security standing of the network. New services may need been put in or the firewall rules may need been modified while not change the content. This can result in attacks that square measure labelled as non-relevant, albeit a vulnerable target exists. Another disadvantage is that the limitation of the sort of knowledge which will be gathered prior to. As the signature of detected attack is matched against a packet sent to a vulnerable machine, the attack may still fail for variety of different reasons.

2) Active

Active alert verification mechanisms don\'t suppose a priori gathered data. Instead, the verification method actively initiates the knowledge gathering method once Associate in nursing alert is received. A network association permits scanning of the attack target and permits one to assess whether or not a target service continues to be responding or whether or not it\'s become unresponsive. 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 4.

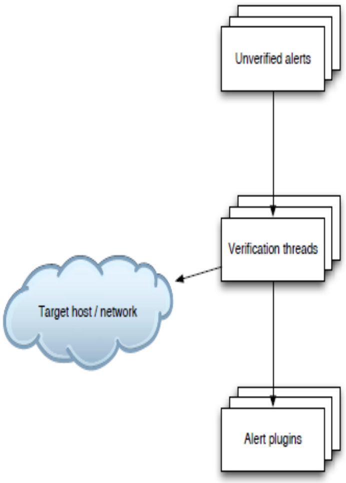


Fig4: Snort Alert Verification Architecture

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 employment of multiple Snort sensors would imply that multiple verification modules area unit running. This could be no downside; as a result of the performance impact of the verification tool is low.

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.

This work represents AN early add constructing a non-critical alarm filter by means that of discourse signatures and hash functions. 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.