Comparative Evaluation of Header vs. Payload based Network Anomaly Detectors

Faisal M. Cheema, Adeel Akram, Zeshan Iqbal

Abstract— **With the immense growth of services offered by Internet, the requirement of broadband connectivity has increased significantly in past few years. Organizations and individuals are relying heavily on the internet for their daily communication needs. Consequently, networks have become more prone to different types of network attacks. Intrusion Detection Systems (IDS) offer a method to protect networks against many such attacks. Numerous IDS have been proposed in literature, which employ different techniques to identify attack patterns as well as abrupt changes in network traffic flows. Anomaly detection is a type of Intrusion Detection corresponding to a suite of techniques that can be used to identify novel or "zero-day" attacks against computers and network infrastructure. Different Anomaly-based Intrusion Detection Systems (ADS) work on different principles e.g., a few take into account the packet headers only, where as others operate on payload as well as packet headers. In this paper we evaluate six different ADS; three of them work on packet header only, while remaining three works on both header and payload. We aim to provide a conclusive comparison of these ADS (header only or both header and payload) in terms of accuracy, complexity and detection delay to highlight factors that must be considered while designing IDS in future. The comparison is performed using two real-world labeled datasets to enable cross-reference for future research in this field. In the end of this paper we will conclude that anomaly detectors which work on both header and payload perform better than those ADS which consider only header for intrusion detection.**

Index Terms—**Anomaly detection, Comparative evaluation, Worms, Zero-day attacks.**

I. Introduction

In recent era of information technology, organizations are becoming vulnerable to a wide variety of network attacks and try to mitigate these attacks because of there high financial impact. Recent security report by Cisco [1] has shown that network attacks by viruses, worms and backdoors have increased many folds with the growth of internet. Although viruses attach themselves to a program and require humans to propagate them to other locations and computers, however worms also propagate on their own by using network infrastructure. Both viruses and worms infect hosts, and exploit known vulnerabilities in computer operating systems, application software, device drivers and services. A major challenge in networks is to detect new worms and viruses in the early stages of propagation to limit or stop them from spreading. As an example, a single worm W32/SQLSlam-A [2] infected 75,000 machines in 30 minutes [3] and caused disruption of major network services. In response to these threats, there is an immense requirement for effective techniques to detect the presence of such malicious activities in the networks. So network based IDS play an important role in this regards.

Most of these IDS employ signature based technique for detecting network attacks e.g., Snort [4]. This technique works well only for attacks that have a known pattern match in the signature database. Such IDS must be updated frequently in order to keep their signature database up to date for efficient detection of newer threats. With the ever increasing number of new or zero-day attacks, it's not possible to maintain a large database of newly identified and previous attack patterns. Signature-based detection method suffers from their inability to detect new type or zero-day attacks whose signature is not included in the signature database. Furthermore continuous growth in the size of signature database adversely affects the efficiency of the detection algorithms. In contrast ADS e.g., SPADE [4] builds models of normal traffic flow data and then attempt to detect deviations from the normal model in observed data.

Network based ADS use protocol header or both header and payload to build a network traffic profile and detect various types of network intrusions by using this profile. Most of the emerging worms use payload to deliver malicious contents to the remote machine over the network. This indicates that, both protocol header and the payload contain very important information regarding intrusion detection.

Anomaly detection overcomes the limitation of signature based detection by focusing on the run-time deviations observed from normal behavior. Anomaly detection algorithms work on two phases i.e., training phase and the detection phase. In training phase normal traffic behavior is observed from the benign/non-malicious traffic and algorithm make a profile of it. In the detection phase, the live/run-time network traffic is compared with the learned profile and any deviation from the normal behavior is flagged as anomaly.

Several anomaly detection algorithms are being proposed these days [5]-[9], some of which employ data-mining techniques and others work on neural-networks based learning approaches to build the normal traffic behavior but very little effort has been made on the comparison of these anomaly detection algorithms. In this paper we compare and evaluate three header based anomaly detectors with three header and payload based anomaly detectors on two labeled datasets. One dataset is publicly available and other is a real

Manuscript received March 9, 2009. This work was supported in part by the University of Engineering & Technology Taxila, Pakistan.

Faisal Munawar Cheema, Adeel Akram and Zeshan Iqbal are with the University of Engineering & Technology Taxila, Pakistan and corresponding authors e-mail are faisalcheema@uettaxila.edu.pk, adeel@uettaxila.edu.pk and zeshan@uettaxila.edu.pk.

network data collected from our university research centre. The anomaly detectors evaluated in this paper can be categorized in two classes, first class comprises of anomaly detectors which take into account the header only [10], [11] and [12]. However the second class comprises of anomaly detectors that operates on both header and payload [13], [14] and [15]. All the anomaly detectors, evaluated in this work, employ very different theoretical frameworks for building profiles.

The first objective of this study is to propose a framework which would be most effective for detecting emerging network threats and for designing efficient anomaly detectors in future. Second objective is to identify the best algorithm under varying rate of attacks and complexity. These conclusions should be taken into account, while developing new ADS, thus, improving the performance of proposed detectors for detecting novel threats.

These anomaly detectors' performance is compared based on the accuracy, complexity and detection delay. Accuracy of an anomaly detector can be termed as the ratio of the detection rate and false alarm rates. However, complexity refers to the inherent working of algorithm which requires computation for building normal or run-time profile. Furthermore, detection delay is the time gap between the start of anomaly and alarm raised by the anomaly detector. Later this paper shows that the ADS works on both header and payload provide better results in term of detection rate but have high complexity and detection delay.

Rest of this paper is organized as follows: Section II gives an overview of related work. Section III, describes the evaluation datasets used. Section IV, gives a brief overview of the algorithms used in this research work. Section V, explains the comparative evaluation and reasons for better performance of the anomaly detectors. Section VI concludes this paper.

II. Related Work

In this section we focus on work previously done in the field of anomaly detections and their comparisons. Network anomaly detection has been actively researched since late 1990s. Researchers approach this work by using various techniques such as data mining, artificial intelligence, machine learning and information theory etc.

With the DARPA dataset available publicly the research of anomaly detection increased rapidly but comparative study of these anomaly detectors is rarely performed. We only found four such studies but none of them compared the anomaly detectors on the basis of header and payload.

Wong, et al in [16] presented an empirical analysis of different rate limiting schemes and evaluated their performance using error rate. However in another study [17], seven header based web anomaly detectors are compared on the basis of their accuracy and outline the serious limitations of several anomaly detectors.

Another comparative study of anomaly detection [18] was performed using standard as well as specific metrics that are especially suitable in detecting intrusions involving multiple network connections and compares several data mining based anomaly detection techniques on DARPA 1998 dataset. Finally, comparative evaluation of bio-inspired anomaly detection [19] was performed where accuracy of the port scan detection is enhanced using special artificial intelligence features.

Our work compares performance in terms of accuracy, complexity and detection delay of anomaly detectors on the basis of header and payload.

III. Evaluation Dataset

An accurate evaluation of anomaly detectors can only be performed on real and labeled dataset collected through a well designed experiment. For the purpose of comparative evaluation of anomaly detectors we used two datasets; one is collected by Lincoln Laboratory MIT, known as DARPA 1999 IDS evaluation dataset which is publicly available and commonly used by the research community. The second dataset is collected from our university's Network Administration and Research Center.

A. The 1999 DARPA Dataset

The 1999 DARPA intrusion detection evaluation dataset [20] was collected by Information Systems Technology group of Lincoln Laboratory MIT under sponsorship of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL/SNHS) for the evaluation of intrusion detection systems. What was this dataset designed to find the strength and weaknesses of existing approaches and lead to large performance improvements and valid assessments of intrusion detection systems. In their experiments they collected the actual network statistics and then using traffic generators they synthesized normal and attack traffic on a private network. On that network they generated non-sensitive traffic using public domain contents (e.g. email, files) and randomly generated n-gram word sequences.

TABLE 1: Shows the different instances of header and payload based attacks in 1999 DARPA dataset

Header Attacks	Attacks Instances	Header+Payload Attacks	Attacks Instances
Ipsweep	7	Is_domain	2
Queso	4	Satan	2
Arppoison	5	Mscan	1
Warez	4	Crashiis	7
Smurf	5	Mailbomb	3
Sshtrojan	3	Guess passwd	10
Imap	2	Ncftp	5
Netcat	4	Sendmail	2

Traffic generation was automatic and used the same software tools as were used in the Air Force Research Lab. They incorporated the attack traffic using many old and new attacks and then performed careful labeling of those attacks in datasets.

The 1999 DARPA dataset contained 201 instances of 58 different kinds of attacks. Some of these attacks work on header only and some of these works on both header and payload. There are total 107 instances of 33 types of payload based attacks including denial of service (DOS) attacks,

remote system to a local user attacks (R2L), attacks that transit from a local user to root (L2R), and surveillance/ probing attacks. TABLE 1 shows some instances of attacks based on only header vs. header and payload.

B. Endpoint Dataset

There is no labeled dataset of payload based attacks which is publicly available except for MIT Labs'. Therefore, we developed a second real traffic dataset with header and payload based attacks. We selected two payload based attacks: 1) Cross Site Scripting, 2) SQL Injection. In cross site scripting an attacker injects malicious scripts in the web application on the client side. This malicious script runs in the security context of the victim's browser and can access the credentials of the victim. The attack is usually possible as most of the web applications allow user input to be entered in the web application and is unable to filter malicious contents. SQL injection is a subset of unauthorized user input vulnerability, and the idea is to convince the application to run SQL code that was not intended. If the application begins to create SQL strings naively on the fly and start running them, it is straightforward to create some real surprises. In addition to above attacks, the malicious server infected with Nimda-D [21], Mydoom, [22] and CodeRed [23] worms to generate attack traffic. The simulation of these worms using parameters has been discussed in literature [24]. Our experimental setup comprises of a very simple design in controlled environment as shown in Fig. 1.

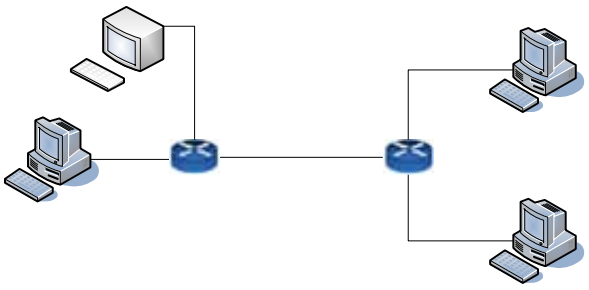

Fig. 1: Network Architecture for Endpoint Dataset Collection

Sniffer provides a copy of the ongoing traffic between user, malicious and normal servers. We generated benign traffic between user and normal server using FTP sessions, mails and web services. During this communication we initiated sessions between user and malicious server at regular intervals, with different signatures of attack web pages and it comprises 241,532 sessions during five months. Scans were identified by flagging those packets having malicious server IP. These are then labeled according to attack type and signature.

IV. Anomaly Detection Algorithms

We evaluated six anomaly detectors; three of them work on header only, however remaining three works on both header and payload. All the anomaly detectors compared in this work are well known in the research community. Brief introduction of them is given below. Readers are advised to refer to their respective publications for more details.

A. Virus Throttle (VRT)

Virus Throttle [10] detects and prevents anomalous traffic and worms like Nimda-D [21] and CodeRed [23] Worms. This algorithm is based on the fact that suspicious traffic can be detected by limiting the number of outgoing connections. An infected host tries to make many outgoing connections in a small period of time. Virus Throttle restricts only one connection per second to maximum of five new destinations and further connection attempts are added in a queue of certain threshold and processes one connection per second. An anomaly is detected when this queue exceeds a threshold value.

B. Threshold Random Walk (TRW)

Threshold Random Walk [11] is a fast anomaly detector. This algorithm states that the successful connection attempt of a *scanner* host should be lower than that for a benign host. This algorithm uses sequential hypothesis testing to determine the behavior of a scanner or benign host. It was specially designed to detect incoming port scan attacks.

C. Max Entropy Estimation (MEE)

Max Entropy Estimation is a behavior-based anomaly detector [12]. This algorithm compares the network traffic distribution against a baseline distribution which is calculated using Maximum Entropy Estimation. The training traffic is divided into 2,348 packet classes and their baseline distribution is calculated using Max Entropy Estimation and then compared it with the run-time traffic distribution using Kullback-Leibler measure for detecting anomalies.

D. Application Layer Anomaly Detection (ALAD)

Application Layer Anomaly Detection [13] is a non stationary model in which the probability of an event depends on the time of occurrence and not on the average frequency. ALAD models incoming TCP connections to the well known server ports and examine first 1000 bytes of the request. To make a good detection model it selects five fix rule forms or models whose attributes are source and destination IP address, destination port, TCP flags and application layer keyword. These five models are selected because they give better performance for detecting novel attacks and are used for anomaly detection. To increase the performance, ALAD also work with PHAD [25] because both algorithms work on different parameters.

E. Learning Rules for Anomaly Detection (LERAD)

Learning Rules for Anomaly Detection [14] uses a machine learning approach to model TCP connections by selecting large number of attributes. It works the same as ALAD but monitors more attributes and parses the application payload to first 8 words. In training phase, a detection model is build using more attributes through a rule generating algorithm and these rules assign a fix score to each TCP connection and anomaly is detected when there is any deviation from the score assigned in training phase.

F. Network Traffic Anomaly Detector (NETAD)

Network Traffic Anomaly Detector [15] models packets

for detecting anomalies. First of all the traffic is filtered to get required data. This greatly increases processing speed. It models first 48 bytes of the packet as an attribute starting from the IP header which contains 8 bytes of the application payload as well. It make 9 models corresponds to most useful protocols like TCP, IP etc. Then anomaly score is calculated by considering both the frequency of events and time.

V. Performance Evaluation

In this section we evaluate the accuracy, complexity and detection delay of selected anomaly detectors on two real-world datasets which are described in Evaluation Datasets section. Performance of anomaly detectors is evaluated by using Receiver Operating Characteristics (ROC) curves. These curves are produced by joining different points of performance obtained at different thresholds.

A. Accuracy Comparison

Using 1999 DARPA IDS evaluation dataset, the average ROC analysis of six anomaly detection algorithms is shown in Fig. 2.

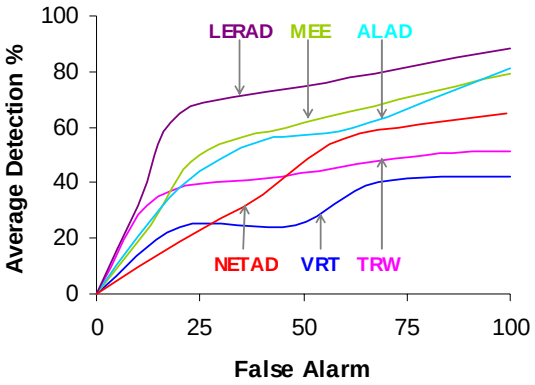

Fig. 2: Performance Evaluation on 1999 DARPA Dataset

It was observed that LERAD provides highest accuracy by achieving 88% detection rate at 100 alarms because it operated on more attributes as compared to the other algorithms. This dataset had much deviation in the attributes of attack traffic which are used for attack detection and LERAD uses excellent profile making technique by taking several attributes for easy detection of any malicious activity. MEE and ALED were in second place and achieved approximately 80% detection rate at 100 alarms. It is noteworthy that a header-only based anomaly detector; MEE, having very low complexity can detect most of the attacks. This is due to high number of header attacks in this dataset. The NETAD approach is also quite accurate as it provides up to 64% detection rate at 100 false alarms. However VRT performed poorly because it is designed to prevent large number of outgoing connection attempts. We also performed a comparison of all these anomaly detectors in terms of their accuracy. It was observed that some of these anomaly detectors provide very low accuracy with large number of false alarms.

Using endpoint dataset, the performance of all the anomaly detectors is shown in Fig. 3. LERAD achieve 100% detection rate with only 92 false alarms which also perform the best in 1999 DARPA dataset. Similarly ALAD also outperform in end point data set with 98% detection rate at 100 false alarms even it had small number of attribute for training. TRW is not suitable to scale the endpoint dataset and provide very poor performance up to 41% average detection rate. Similarly the performance of all detectors can be observed and compared using ROC analysis of Figure 3. However it is concluded that the false alarm rate is very high in both datasets.

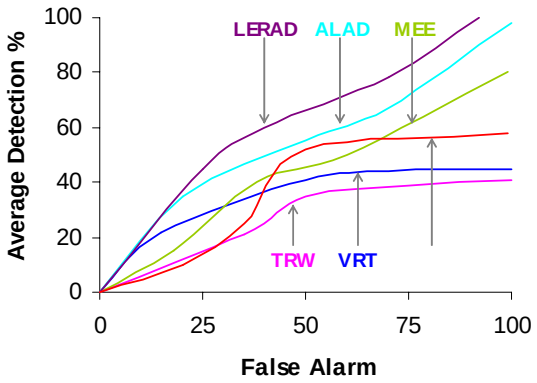

Fig. 3: Performance Evaluation on Endpoint Dataset

Results obtained from the two datasets clearly indicate that LERAD and ALAD provide the best accuracy and excellent performance in detection rates. Both of these anomaly detectors operate on both header and payload. Hence, this should be considered as a basic requirement for developing new anomaly detectors, in terms of accuracy.

B. Complexity Comparison

The run-time complexity analysis [26] of these anomaly detectors is shown in TABLE 2. The complexity of the anomaly detector algorithm has no relationship with the accuracy of an anomaly detector.

TABLE 2: Complexity Analysis of Anomaly Detectors

S. No.	Anomaly Detector	Complexity (sec)
1	VRT	72
2	TRW	81
3	MEE	53
4	ALAD	96
5	LERAD	102
6	NETAD	89

LERAD has highest complexity because it takes into account the highest number of attributes for building run-time profile thus causing more delay as compared to other anomaly detectors. Consequently, deep packet inspection introduces more delay in their performance. Since header and payload based anomaly detectors perform better than those operating on header only, there is a trade-off between complexity and performance. Thus we conclude that a lot of effort is required to reduce the complexity of these

algorithms, allowing future anomaly detectors to improve their performance while keeping the complexity low.

C. Detection Delay Comparison

Detection delay of all the anomaly detectors was calculated by observing the time difference between the start of anomaly and alarm raised by the anomaly detector. Since, few algorithms incur high detection delay and report an anomalous event after its complete execution, thus making systems more prone to network attacks. Hence, this is an important consideration for designing new ADS because an accurate anomaly detector may report an anomaly in very short interval so that mitigation can be possible.

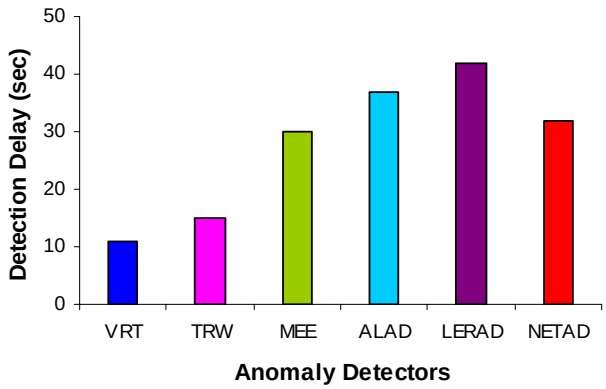

Fig. 4: Average detection delay of Anomaly Detectors using both data sets.

Average Detection delay comparison of anomaly detectors using both dataset is shown in Fig. 4. It can be clearly seen that LERAD has the highest detection delay due to its inherently high complexity. Owing to the fact that, it operates on more attributes as compared to other algorithms, so it suffers with high detection delays. This comparison gives an insight that anomaly detectors operating on both header and payload incur high detection delay due to deep packet inspection for payload. However, on the other hand, anomaly detectors operating on header only incur low detection delay.

VI. Conclusion

In this paper we evaluated six network-based ADS using two real-world labeled datasets. It is observed that both packet header and payload contains important information that must be considered for making profiles of most network attacks. Secondly the detection rate under false alarms is also not satisfactory indicating that there is a strong need to improve the accuracy keeping the complexity as low as possible. It is also concluded that anomaly detectors which work on both header and payload provide better performance only if they use good model for profile creation but their complexity and detection delay is much greater due to deep packet inspection. Hence, there is a strong need of investigating techniques that can provide better accuracy with low complexity and detection delay. Currently, it is a trade-off between accuracy, complexity and detection delay.

References

[1] Cisco 2008 Annual Security Report http://www.cisco.com.
[2] W32/SQLSlam-A. Sophos Anti-Virus. http://www.sophos.com/security/analyses/viruses-and-spyware/w32sqlslama.html
[3] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. The spread of the Sapphire/Slammer worm (2003).
[4] Snort, the Open Source Network Intrusion Detection System, http://www.snort.org/.
[5] Schechter S.E., Jung J., Berger A.W.: Fast detection of scanning worm infections. In: RAID (2004).
[6] Soule A., Salamatian K., Taft N.: Combining Filtering and Statistical methods for anomaly detection. In: ACM/Usenix IMC (2005).
[7] Lakhina A., Crovella M., Diot C.: Characterization of network-wide traffic anomalies in traffic flows. In: ACM Internet Measurement Conference (IMC) (2004).
[8] Next-Generation Intrusion Detection Expert System (NIDES), http://www.csl.sri.com/projects/nides/
[9] K. L. Ingham. Anomaly Detection for HTTP Intrusion Detection: Algorithm Comparisons and the Effect of Generalization on Accuracy. PhD thesis, Department of Computer Science, University of New Mexico, Albuquerque, NM, 87131, (2007).
[10] Twycross J., Williamson M.M.: Implementing and testing a virus throttle. In: Usenix Security (2003).
[11] Jung J., Paxson V., Berger A.W., Balakrishnan H.: Fast portscan detection using sequential hypothesis testing. In: IEEE Symp Sec. and Priv. (2004).
[12] Gu Y., McCullum A., Towsley D.: Detecting anomalies in network traffic using maximum entropy estimation. In: ACM/Usenix IMC (2005).
[13] M. Mahoney, P. K. Chan, "Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks", Edmonton, Alberta: Proc. SIGKDD, 2002, 376-385.
[14] M. Mahoney, P. K. Chan, "Learning Models of Network Traffic for Detecting Novel Attacks", Florida Tech. technical report 2002-08.
[15] M. Mahoney, "Network Traffic Anomaly Detection Based on Packet Bytes", Proc. ACM-SAC, Melbourne FL, 2003.
[16] Wong C., Bielski S., Studer A., Wang C.: Empirical Analysis of Rate Limiting Mechanisms. In: RAID (2005).
[17] Kenneth L.I., Inoue H.: Comparing Anomaly Detection Techniques for HTTP. In: RAID (2007).
[18] Lazarevic A., Ertoz L., Kumar V., Ozgur A., Srivastava J.: A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection. In: SIAM SDM (2003).
[19] Shafiq M.Z., Khayam S.A., Farooq M.: Improving Accuracy of Immune-inspired Malware Detectors by using Intelligent Features. In: ACM GECCO (2008).
[20] DARPA-sponsored IDS Evaluation (1999) by MIT Lincoln Lab, http://www.ll.mit.edu/IST/ideval/data/data_index.html.
[21] W32/Nimda-D. Sophos Anti-Virus. http://www.sophos.com/support/disinfection/nimda.html
[22] Mydoom, Symantec Security Response, http://www.symantec.com/
[23] Code Red, CERT Advisory CA-2001-19. CERT Coordination Center. http://www.cert.org/advisories/CA-2001-19.html.
[24] Shannon, C., Moore, D.: The spread of the Witty worm. In: IEEE Sec & Priv, 2(4), pp. 46-50 (2004).
[25] Mahoney M.V., Chan P.K.: PHAD: Packet Header Anomaly Detection for Identifying Hostile Network Traffic. Florida Tech. technical report CS-2001-4 (2001).
[26] HPROF, Java Heap/CPU Profiler. http://java.sun.com/