Payload Content based Network Anomaly Detection

Sandeep A. Thorat Amit K. Khandelwal Bezawada Bruhadeshwar K. Kishore
Centre for Security Theory and Algorithmic Research (CSTAR)
IIIT-Hyderabad, India
{sandeep_thorat,khandelwal}@students.iiit.ac.in, {bezawada,kkishore}@iiit.ac.in

Abstract

We present Payload Content based Network Anomaly Detection, we call as PCNAD. PCNAD is an improvement to PAYL system which is considered one of the complete systems for payload based anomaly detection. PAYL takes into consideration the entire payload for profile calculation and effectively for anomaly detection. Payload length is very high on port numbers like 21 and 80. Hence it is difficult to apply PAYL on high speed, high bandwidth networks. We use CPP (Content based Payload Partitioning) technique which divides the payload into different partitions depending on content of payload. PCNAD does payload based anomaly detection using a few CPP partitions. We demonstrate usefulness of the PCNAD on the 1999 DARPA IDS data set. We observed 97.06% accuracy on port 80 using only 62.64% packet payload length with small false positive rate. This is a significant improvement over PAYL approach which uses 100% of the packet payload for anomaly detection.

1 Introduction

In recent era of information security systems all major network intrusion detection systems are still using signature based approaches for attack detection. Snort [1] and Bro [13] are popular example of signature based intrusion detection systems. Such systems use attack detection mechanisms based on signatures of already known attacks or vulnerabilities. This technique works well if the signature database is up to date. However, it has been observed that signature based detection mechanism fails for zero day attacks or mutation of known attacks due to lack of signature availability when a zero day attack is launched. An alternative choice for this problem is anomaly detection systems [8]. In anomaly detection mechanism deviation from normal behavior is detected to signal possible novel attack. Network based anomaly detection can be applied at different levels: protocol headers, packet payload. In PCNAD we focus on packet payload based network anomaly detection. Such system is useful for detecting payload based attacks like R2L, U2R, and worm infections. We found PCNAD system useful for detecting suspicious packets arriving on the network which contain any payload based attack.

We propose a system which analyzes normal payloads for a particular service on a host and makes a set of payload profiles that are expected for that service. The payload's profiles are specific to the host and the communication behavior of the service; hence same profiles are not applicable across the different network environments. The system calculates byte frequency distribution using 1-gram based approach to make payload profile. The packet payload length has strong impact on the byte frequency distribution, so multiple profiles are created for different payload lengths. Due to this, number of profiles for particular service becomes very large. To minimize the complexity of profile comparisons, profiles are clustered together. The system is trained in unsupervised way for profiles creations. In the testing phase system captures incoming payloads and compares the payload with stored normal profiles. If the new payload profile does not match with any stored profile for same service, then an alert is generated indicating suspicious packet. The system can be deployed at network entry point level or at an end host.

A similar approach is used by PAYL [18] system and proved to be very useful for attack detection on port 21 and 80. But payload length observed on the port 21 and 80 is generally very high. And byte frequency computation for longer length payload becomes difficult on high speed networks. Also if proper care is not taken in payload profiling such a system is vulnerable to mimicry attacks. Taking into consideration these facts, our system makes payload profiles depending on content of the payload rather than using the same approach for all payloads. PCNAD uses Content based Payload Partitioning (CPP) which was introduced and used by [11] and [17] for making partitions in the files according to the file content. CPP partitions the packet payload into a number of partitions and then packet profile is computed on these partitions. We tested our system on the 1999 DARPA IDS data set [14] which is a commonly used data set by intrusion detection research community. In the 1999 DARPA IDS data set entire packet payloads are available which is useful of our system. We found 97.06% correctness in attack detection with 8.82% false positive rate in the results by profiling port 80 packets on 11 CPP partitions in payload. The 11 CPP partitions use 62.64% of payload length on average which is much less than payload length used by anomaly detection system PAYL [18] (which is 100%). This reduction in payload scanning length makes PCNAD system operable in high speed networks. While partitioning CPP takes into consideration payload content, hence the system is robust against the mimicry attacks.

The rest of the paper is organized as follows. Section 2 discusses related work in network anomaly detection. In Section 3 we describe our system working in detail. Section 4 presents the experimental results and evaluations of the method applied to the 1999 DARPA IDS data set. In Section 5 we conclude the paper.

2 Related Work

Presently in industry rule based network intrusion detection systems such as Snort [1] and Bro [13] are most popular. These systems use signatures or fingerprints to identify known attacks. But signature based systems are clueless in case of novel attacks where attack pattern is not matching with stored signatures. In such scenarios anomaly detection systems differentiate between a 'normal' network activity and something other than 'normal'. These systems give better attack detection at the cost of high false positive rate. Network Anomaly detection systems such as PAYL [18], ALAD [9], PHAD [10], SPADE [3], NIDES [4] and NATE [16] compute statistical models for normal network traffic and generate alarms when incoming traffic shows a large deviation from the normal model. These systems can be further classified depending on the features used to compute the normal models. Some of these systems viz. SPADE, NIDES and PHAD systems work with protocol headers. These systems use different features extracted from Ethernet, IP, ICMP, TCP, UDP packet headers for anomaly detection. These systems have shown better results for detecting protocol level attacks like scanning, probing etc. As these systems ignore the payload contents, they show very poor results for detecting application level attacks. Few network intrusion detection systems use connection and state level information for modeling profiles.

Some systems use few payload features for anomaly detection. NATE [16] uses first 48 bytes as a statistical feature starting from the IP header which includes at most the first 8 bytes of the payload of each network packet. But 48 bytes of payload is too less to find out any payload based attack. ALAD [9] uses features depending on the first word of each input line from the first 1000 application payloads in addition to packet header features. ALAD system tries to analyze the payload without any pre-knowledge just like us, but it restricts profiling to only first 1000 lines.

Web based anomaly detection systems like [7] and [8] are based on only packet payload. As these approaches focus only on HTTP traffic, it takes advantage of known protocol format for constructing profile model. But these systems are not useful for detecting attacks on other protocols. Our system shares many characteristics with PAYL [18] which uses an approach based on payload byte distribution for profile computation. PAYL uses the entire payload for profile computation. Due to this system is not well suited for high speed network with huge data travelling across the network. Our system aims to remove this weakness of the PAYL system by profiling payloads depending on CPP (Content Based Payload Partitioning).

3 Payload Content based Network Anomaly Detection

Anomaly detection systems run in two phases. In the training phase these systems profile normal behavior of network activity and store these profiles. In the testing phase such system compares current network activity profile with the stored profiles and report alerts when anything other than normal profile is seen on network. For such a system following are major design goals [18]:

1. Generality of the system: The system should be applicable for broad range of applications or protocols. Due to this anomaly detection systems which are protocol and service independent are always preferred.

2. Incremental Profiling: The incremental profile updates computed profiles to accommodate changing communication patterns. The network activities keep on changing; the attack detection mechanism should proactive to accommodate these changes.

3. Low false positive rate: False positives are a major area of concern in anomalous detection systems. As these systems reports alert for any thing other than stored normal profiles; accuracy in detecting truly anomalous events is very important.

4. Resistance to Mimicry attack: In mimicry attacks, the attacker has access to the same information as the attack detection mechanism. Here the attacker attempts to know what is "normal" for detection mechanism. Once this information is available, attacker crafts an attack to replicate normal behavior. Such an attack may be considered as normal packet payload by anomaly detection system which is a serious vulnerability. The system should be resistant to mimicry attacks.

5. Efficiency to operate in the high bandwidth environments: As high speed, high bandwidth networks becoming very common, anomaly detection mechanism need to be enough efficient to scan a huge amount of traffic.

6. Unsupervised Learning: Anomalous detection system requires availability of labeled data in the training phase. Due to human errors in labeling a huge volume of data unsupervised learning is always preferred in any anomalous detection system for normal profile creations. Unsupervised learning requires very little or no human intervention in the training phase.

In past few years it has been observed that, the above design goals are difficult to achieve together. Hence different systems attempt to balance these competing criteria's for payload anomaly detection.

3.1 PCNAD System Architecture

The PCNAD system architecture is shown in Figure 1. The system has two major components Packet Profiler and PCNAD Anomaly Detector. The Packet profiler component deals with normal packet profiling and creates the reduced profile model. In PCNAD Anomaly Detector module incoming packet is compared with stored profiles and alerts are generated if incoming packet is significantly different from stored profiles.

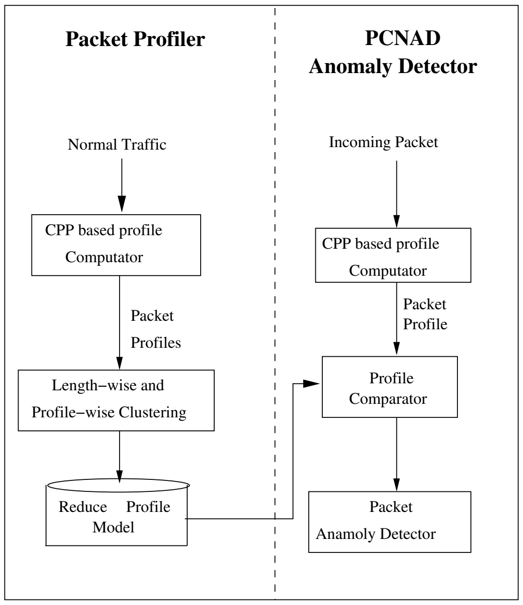

Figure 1. PCNAD System Architecture.

In PCNAD, we model the payload using an n-gram analysis. An n-gram is a sequence of n adjacent bytes in a payload unit. A sliding window with width n is passed over the whole payload and the occurrence of each n-gram is counted. Recently many security systems have started using n-gram analysis [2] for anomaly detection systems.

Our system profiles the payload using 1-gram approach in unsupervised way. A 1-gram model is considered as the simplest model for payload profiling. This simplicity is very useful while processing high voluminous network data. A 1-gram model requires a linear time scanning of the payload data and an update of a small 256-element histogram. This histogram stores byte frequencies observed for ASCII characters 0-255. PAYL has shown that the 1-gram model is sufficient and accurate for payload based attack detection.

As packet payloads show very wide variation in content, it is necessary to cluster packet payloads according to various criterions and then apply profiling on these clusters of packets. Our system does the payload clustering depending on the destination port number, length of the payload, direction of traffic (inbound or outbound) and byte frequency profile of payload. This results in a much reduced profile model which is useful in the testing phase. Each network application has its own protocol and has its own payload type. This payload is site specific and varies with time so incremental profiling is required. Our system divides the packet payload into variable length content blocks using CPP(Content Based Partitioning). Before understanding the training phase for packet profiling lets discuss CPP.

3.2 Understanding Content Based Payload Partitioning

Content Based Payload Partitioning was introduced in the file system domain in LBFS [11]. Autograph [6] and Earlybird [15] uses CPP for partitioning the payload at real time. CPP scheme determines the boundaries of each payload partition based on payload content. It generates variable length content block from the payload. The generated partitions changes a little under byte insertion or deletion from payload. This makes PCNAD system robust against the mimicry attacks.

CPP uses Rabin fingerprinting [12] to partition packet payload into content blocks. It computes a series of Rabin fingerprints r_i over a sliding n-byte window on the packet payload. It starts with first n bytes in the payload and slides one byte at a time toward the end of the payload. It is efficient to find a Rabin fingerprint over a sliding window due to the linear complexity of computations. As CPP slides its window along the payload, it ends a content block when r_i matches a predetermined stopping criteria S. The average content block size produced by CPP is depending on user defined stopping criteria which is configured by the user. We can choose stopping criteria in different ways, it does not affect end results as long as we apply same criteria in the training and testing phase.

Rabin fingerprint function gives an integer value (r_i) over a window size payload. If r_i mod 1000 is in the range of 550 to 600 (this is stopping criteria S we have chosen) we declare end of the current partition and move to next character to get another partition. Else we proceed to the next character after adding the present character to the current partition.

As CPP decides content block boundaries probabilistically, CPP may generate very short content blocks. Very short content blocks do not represent byte frequency distribution of the payload properly causing payload profiling is most likely to be incorrect. Due to this in PCNAD, we impose minimum content block size limit and take care that this condition is satisfied by CPP generated partitions.

3.3 Training PCNAD for profile creations

In the training phase PCNAD calculates the payload profile specific for a destination port. Since payload length arriving at a particular port varies a lot, this causes variation in payload profile. The different length payloads have different types of content; larger payloads are more likely to have non-printable characters. Thus, we compute a payload model for all different lengths for each port depending on direction of flow (inbound or outbound). For a real time traffic monitoring system it is necessary to keep profile model simple. We use the frequency of each ASCII character 0-255 which we call as 'byte frequency' for profile calculation. But some stable character frequencies and some variant character frequencies can result in the same average frequency. Hence we compute the variance and standard deviation of each frequency as another characterizing feature. The average and standard deviation values of 256 characters from one profile are used to compare that profile with another profile. In the training phase we compute profile for each specific length l of payload targeted to a destination port d. It computes $P_{d,l}$ model on packet payloads which is byte frequency of payload with length l targeted to port d . This generates huge number of profiles as for every small length variations we are computing profiles differently. To reduce information stored in the profile model we apply different clustering techniques on these profiles. These clustering techniques are length-wise clustering and profile-wise clustering. PCNAD initially applies length wise clustering, which combines profiles of length l_i and l_j together where difference between the lengths l_i and l_j is less than threshold l_t. The resultant profile has byte frequency distribution which is average of two profiles combined together. The resultant profile represents payload's having a length equal to the average payload length of combined profiles. The value of l_t is kept user configurable. Once length-wise clustering is done profile-wise clustering is applied. The profile-wise clustering pays attention on sparse profiles. In a sparse profile very few number of packets participate in the calculation of the profile. In profile-wise clustering, sparse profiles are merged with their nearest matching profile. The nearest matching profile is found by taking into consideration Manhattan distance m_d of the profiles from each other. If the value m_d is less than threshold m_t, then two profiles are combined together using the same technique used in length-wise clustering. The value of m_t is user configurable. The graph in Figure 2 shows normalized average byte frequencies observed on port 80 for characters 0-255 before applying CPP.

Up to this point we use similar techniques for payload profiling with PAYL. But PAYL uses entire payload for profile computations, which is difficult to do in high speed, high bandwidth networks. Our system computes profile depending on only initial N partitions we got after applying CPP on payload. The value of N is user configurable. As described in the above section, CPP does payload partitioning depending on the content of the payload in protocol independent way. CPP gives an offset in payload which is last position of N^{th} partition. We compute the payload profile by using payload length up to this offset. We kept number of partitions to be used i.e. N configurable in system and tested system for correctness of the results. Figure 3 shows the byte frequency graph by taking into consideration 11 partitions (where we got most correct results) of payload on port 80. As we can see graphs in Figure 2 and Figure 3 similar nature and we are not loosing any profile relevant information after using 11 CPP partitions.

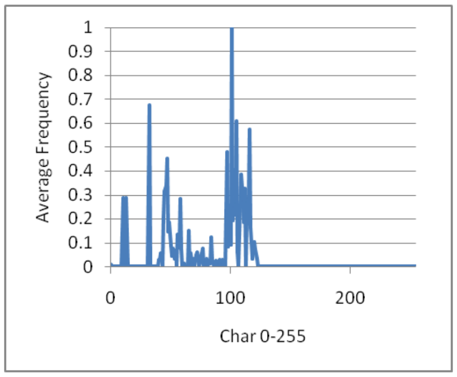

Figure 2. Normalized average byte frequency distribution on port 80 using entire payload.

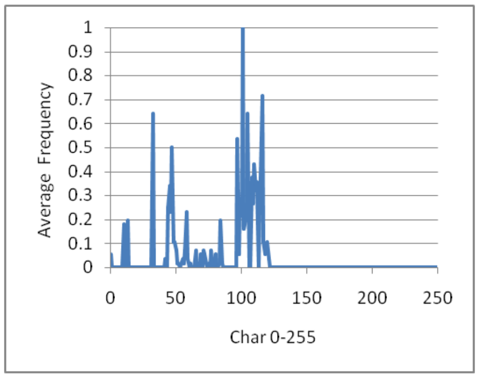

Figure 3. Normalized average byte frequency distribution after applying CPP on port 80 using 11 partitions in CPP.

3.4 Testing PCNAD system for attack detections

In the testing phase PCNAD compares the incoming packet payload profile with stored profile $P_{d,l}$ associated with same destination port and having similar payload length. This comparison is done by finding out the simplified Mahalanobis distance between the stored profile and the new profile. If incoming payload is not matching with any of the stored profile, then an alert is generated notifying suspicious packet arrival. In the testing phase as well we use CPP to partition the incoming payload into N partitions. PCNAD takes into consideration payload length up to N^{th} partition to compute profile of the payload. Here the value of N is equal to the one used in the training phase. To find out the distance between stored profile and incoming packet payload profile, we use simplified Mahalanobis distance which is given by PAYL system [18]. Each profile is considered as a feature vector of 256(ASCII values) elements. The advantage of Mahalanobis distance is that, it takes into account not only the average values but also variance and the covariance of the variables. In computing the Mahalanobis distance high price is paid to compute multiplications and square roots after summing the differences across the byte value frequencies. After noting down these things PAYL has given simplified Mahalanobis distance formula as: $d(B, \overline{A}) = \sum_{i=0}^{255}(|B_i - \overline{A_i}|/(\sigma_i + \alpha))$ where B and $\overline{A}$ are two feature vectors, vector B is the feature vector of the arriving payload, and $\overline{A}$ is the feature vector computed in the training phase. σ_i is standard deviation on i^{th} ASCII characters while α is smoothing factor added to remove possibility of distance becoming infinity when σ_i is zero.

In the testing phase PCNAD does incremental profiling as well. If arriving payload was found to be normal with comparison to some stored profile, then that stored profile is combined with arriving payload profile using technique used in the length-wise clustering.

4 Experimental Results

We tested PCNAD on the 1999 DARPA IDS data set [14], which is considered as a standard data set to evaluate intrusion detection systems. The data consists of three weeks of training data and two weeks of test data. Although there are problems due to the nature of the simulation environment that created the data, it still remains a useful set of data to compare techniques [5].

In our experiment setup we initially implemented payload based anomaly detection system which profiles the packet taking into consideration entire payload. We conduct experiments using each packet as the data unit which were inserted in mySQL database. We examined the only inbound TCP traffic to the ports 0-1023 of the hosts 172.16.xxx.xxx.

We trained the system on the DARPA dataset using data of week 1 and week 3 and then evaluated the detector on data of weeks 4 and 5. We found 61 attack instances out of 62 on port 21 and port 80. Thus using full payload 98.39% of attacks were detected. Afterwards we added CPP module in the profiling module for selecting initial N partitions of the packet payload and we computed payload profile on these partitions (rather than using whole payload). We kept number of CPP partitions to be taken into consideration for payload profiling configurable. We got best result at 0.001 smoothing factor. The rest of configurations like l_t,m_t, smoothing factor etc. kept exactly similar as that of previous system (where whole payload is taken into account for profiling) implementation and observed correctness of result for various choices of N.

As we increase number of partitions, the average payload size taken into consideration for profile computation goes on increasing. Also we observed an increase in the correctness of results with increase in number of CPP partitions taken into consideration for payload profiling. Figure 4 shows the relevant results.

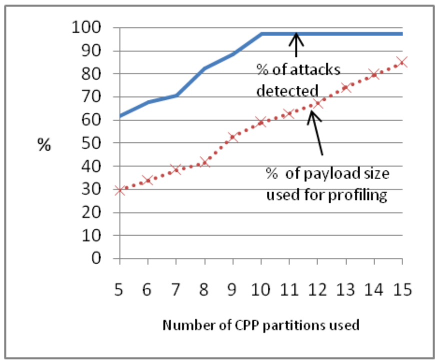

Figure 4. Percentage of attacks detected and % of payload size used by varying CPP partitions on port 80.

For 11 partitions, we found best results with 97.06% of attacks detected correctly and using an average 62.64% of packet payload. False positive rate observed for these configurations is 0.17%. This indicates that we could save 37.36% of packet processing on port 80 which is a significant improvement over PAYL. We got similar success on port 21, where 96.43% of attack detected using average 60.12% of packet payload used and a false positive rate of 0.1114% when 11 partitions are used. The results are shown in Table 1.

5 Conclusion

In this paper we proposed a system which can detect anomalous packet payloads without taking into consideration entire payload length. This is improvement to systems like PAYL which considers whole payload for anomaly detection. The experimental results prove our method good

	Payload profiling using full payload	Payload profiling using CPP
Attacks detected	61	60
Average % of payload considered	100	61.13
Average % of false positive rate	0.0623	0.1407
Time required to process 100 packets in the training phase	102.592 ms	101.269 ms
Time required to process 1000 packets in the testing phase	5.171 ms	5.014 ms

Table 1. Result Summary at port 21 and 80

in attack detection on port 21 and port 80. As PCNAD uses Content based Payload Partitioning (CPP), the system is safe against the mimicry attacks. Presently our system is showing poor results at other ports like 22, 23, 25. Also false positive rate is more after using CPP partitions for profiling which may be due to random nature of data at these ports. If we combine our technique with header level, session, and connection based information then false positive rate can be brought down. As a future work we plan to revise the partitioning criteria in order to apply this technique successfully on other ports like 22,23, and 25.

References

[1] Snort: The open-source network intrusion detection system.

[2] M. Damashek. Gauging similarity with n-grams: language independent categorization of text. In *Science, 267(5199)*, pages 843–848.

[3] J. Hoagland. Spade. In *Silican Defense, http://www.silicondefense.com/software/spice*, 2000.

[4] H. S. Javits and A. Valdes. The nides statistical component: Description and justification. In *Technical report, SRI International, Computer Science Laboratory*, 1993.

[5] Salvatore J. Stolfo Ke Wang, Gabriela Cretu. Anomalous payload-based worm detection and signature generation. In *Proceedings of the Eighth International Symposium on Recent Advances in Intrusion Detection RAID 2005*, pages 227–246, 2005.

[6] Hyang-Ah Kim and Karp B. Autograph: Toward automated, distributed worm signature detection. In *Proceedings of the 13th Usenix Security Symposium*, pages 19–19, 2004.

[7] Christopher Krgel, Thomas Toth, and Engin Kirda. Service specific anomaly detection for network intrusion detection. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 201–208, 2002.

[8] C. Kruegel and G. Vigna. Anomaly detection of web-based attacks. In *Proceedings of 10th ACM Conference on Computer and communications Security CCS'03*, pages 251–261, October 2003.

[9] M. Mahoney. Network traffic anomaly detection based on packet bytes. In *Proc.ACM-SAC, Melbourne FL.*, pages 346–350, 2003.

[10] M. Mahoney and P. Chan. Learning nonstationary models of normal network traffic for detecting novel attacks. In *Proc. SIGKDD 2002*, pages 376–385, 2002.

[11] Athicha Muthitacharoen, Benjie Chen, and David Mazieres. A low-bandwidth network file system. In *Symposium on Operating Systems Principles*, pages 174–187, 2001.

[12] Rabin M. O. Fingerprinting by random polynomials. In *Tech. Rep. TR-15-81, Center for Research in Computing Technology, Harvard University*, 1981.

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

[14] et al. R. Lippmann. The 1999 darpa off-line intrusion detection evaluation. In *Computer Networks,34(4)*, pages 579–595, 2000.

[15] Singh S., Estan C., Varghese G., and Savage S. The early-bird system for real-time detection of unknown worms. In *Tech. Rep.CS2003-0761, UCSD*, 2003.

[16] C. Taylor and J. Alves-Foss. An empirical analysis of nate: Network analysis of anomalous traffic events. In *10th New Security Paradigms Workshop*, 2002.

[17] U.Manber. Finding similar files in a large file system. In *In Proceedings of the USENIX Winter Technical Conference*, pages 2–2, 1994.

[18] K. Wang and S. Stolfo. Anomalous payload-based network intrusion detection. In *Recent Advances in Intrusion Detection, RAID 2004*, pages 203–222, September 2004.