Hybrid Intrusion Detection System Using Packet Header Anomaly Detection Algorithm

Sandeep Bhorde[1], Geetanjali Khabale[2], Priyanka Shinde[3] , Prof. Minal Zope[4]
BE Scholar, Department of Computer Engineering, AISSMS IOIT, Pune University, India
BE Scholar, Department of Computer Engineering, AISSMS IOIT, Pune University, India
BE Scholar, Department of Computer Engineering, AISSMS IOIT, Pune University, India
Assistant Professor, Department of Computer Engineering, AISSMS IOIT, Pune University, India

[1]bhordesandeep@gmail.com
[2]gitanjalikhabale@gmail.com
[3] priyankashinde1110@gmail.com
[4]minalzope@gmail.com

Abstract-**With the development of new technology of networking, expansion and online procedures requesting a secure channel, it has become an inevitable requirement to provide the network security. There are various threat sources including software bugs mostly as the operating systems and software used becomes more functional and larger in size. Intruders who do not have rights to access these data can steal valuable and private information belonging to network users. Firewalls are hardware or software systems placed in between two or more computer networks to stop the committed attacks, by isolating these networks using the rules and policies determined for them.**

It is very clear that firewalls are not enough to secure a network completely because the attacks committed from outside of the network are stopped whereas inside attacks are not. This is the situation where intrusions detection systems (IDSs) are in charge. IDSs are used in order to stop attacks, recover from them with the minimum loss or analyze the security problems so that they are not repeated. This paper proposes a solution for obtaining Hybrid IDS based on packet header anomaly detection (PHAD) which are anomaly-based IDSs with the misuse-based IDS Snort which is an open-source project. The experimental results show that the proposed HIDS are able to detect more attacks whose signatures are not included in rule files.

Keywords- **Packet header anomaly detection,
Intrusion detection system,
Hybrid Intrusion detection system.**

I. Introduction

Organizations like bank, companies and other use internet and its applications for their day to day work. Nowadays with the spreading of the Internet and online procedures requesting a secure channel, it has become an inevitable requirement to provide the network security. There are various threat sources including software bugs mostly as the operating systems and software used becomes more functional and larger in size. Intruders who do not have rights to access these data can steal valuable and private information belonging to network users.

Firewalls are hardware or software systems placed in between two or more computer networks to stop the committed attacks, by isolating these networks using the rules and policies determined for them. It is very clear that firewalls are not enough to secure a network completely because the attacks committed from outside of the network are stopped whereas inside attacks are not. This is the situation where intrusions detection systems (IDSs) are in charge. IDSs are used in order to stop attacks, recover from them with the minimum loss or analyze the security problems so that they are not repeated. The hybrid IDS is obtained by combining packet header anomaly detection (PHAD) which are anomaly-based IDSs with the misuse-based IDS Snort which is an open-source project

Intrusions detection systems (IDSs) are systems that try to detect attacks as they occur or after the attacks took place[1]. IDSs collect network traffic information from some point on the network or computer system and then use this information to secure the network. Intrusion detection systems can be misuse-detection or anomaly detection based. Misuse-detection based IDSs can only detect known attacks whereas anomaly detection based IDSs can also detect new attacks by using heuristic methods[1]. In this paper we propose a hybrid IDS by combining the two approaches in one system. IDSs collect information from a computer or a computer network in order to detect attacks and misuses of the system. Many IDSs only analyze the attacks and some of them try stopping the attack at the time of the intrusion. Three types of data are used by IDSs. These are network traffic data, system level test data and system status files.

Most network intrusion detection systems (IDS) that use anomaly detection look for anomalous or unusual port number and IP addresses, where "unusual" means any value not observed in training on normal traffic[2,3]. They use the firewall paradigm; a packet addressed to a nonexistent host or service must be hostile, so we reject it. The problem with the firewall model is that attacks addressed to legitimate services will still get through, even though the packets may differ from normal traffic in ways that we could detect.

The hybrid IDS is obtained by combining packet header anomaly detection (PHAD) which are anomaly-based IDSs with the misuse-based IDS Snort which is an open-source project[4].

The hybrid IDS obtained is evaluated using the MIT Lincoln Laboratories network traffic data (IDEVAL) as a test bed[6]. Evaluation compares the number of attacks detected by misuse- based IDS on its own, with the hybrid IDS obtained combining anomaly-based and misuse- based IDSs and shows that the hybrid IDS is a more powerful system.

The main goal of the proposed work is to investigate how PHAD can be improved the performance of the snort IDS to find the unknown attacks i.e. the attacks having no signature in the database of snort.

II. Related works

[1]Network intrusion detection systems like *snort* (2001) or *Bro* (Paxson, 1998) typically use signature detection, matching patterns in network traffic to the patterns of known attacks. This works well, but has the obvious disadvantage of being vulnerable to novel attacks. An alternative approach is anomaly detection, which models normal traffic and signals any deviation from this model as suspicious. The idea is based on work by Forrest et al. (1996), who found that most UNIX processes make (mostly) highly predictable sequences of system calls in normal use. When a server or *suid root* program is compromised (by a buffer overflow, for example), it executes code supplied by the attacker, and deviates from its normal calling sequence.

[2]It is not possible to observe every possible legitimate pattern in training, so an anomaly detector requires some type of machine learning algorithm in order to generalize from the training set. For rest uses an ngram model, allowing any sequence as long as all subsequence's of length n (about 3 to 6) have been previously observed. Sekar et al. (2000) uses a state machine model, where a state is defined as the value of the program counter when the system call is made, and allows any sequence as long as all of the state transitions have been observed in training. Ghosh et al. (1999) use a neural network trained to accept observed n-grams and reject randomly generated training sequences.

[3]Network anomaly detectors look for unusual traffic rather than unusual system calls. ADAM (Audit Data and Mining) (Barbará, Wu, and Jajodia, 2001) is an anomaly detector trained on both attack-free traffic and traffic with labelled attacks. It monitors port numbers, IP addresses and subnets, and TCP state. The system learns rules such as "if the first 3 bytes of the source IP address is X, then the 3 destination port is Y with probability *p*". It also aggregates packets over a time window. ADAM uses a naive Bayes classifier, which means that the probability that a packet belongs to some class (normal, known attack, or unknown) depends on the *a-priori* probability of the class, and the combined probabilities of a large collection of rules under the assumption that they are independent. ADAM has separate training modes and detection modes.

[4]NIDES (Anderson et. al. 1995), like ADAM, monitors ports and addresses. Instead of using explicit training data, it builds a model of long term behaviour over a period of hours or days, which is assumed to contain few or no attacks. If short term behaviour (seconds, or a single packets) differs significantly, then an alarm is raised. NIDES does not model known attacks; instead it is used as a component of EMERALD (Neumann and Porras, 1998), which includes host and network based signature detection for known attacks.

[5]Spade is a *snort* (2001) plug-in that detects anomalies in network traffic. Like NIDES and ADAM, it is based on port numbers and IP addresses. It uses several user selectable statistical models, including a Bayes classifier, and no explicit training period. It is supplemented by *snort* rules that use signature detection for known attacks. *Snort* rules are more powerful, in that they can test any part of the packet including string matching in the application payload. To allow examination of the application layer, *snort includes* plug-ins that reassembles IP fragments and TCP streams.

III. Proposed Model

In the previous section, we have reviewed the most recent works. IDSs collect network traffic information from some point on the network or computer system and then use this information to secure the network. Intrusion detection systems can be misuse-detection or anomaly detection based. Misuse-detection based IDSs can only detect known attacks whereas anomaly detection based IDSs can also detect new attacks by using heuristic methods. In this paper we pro- pose a hybrid IDS by combining the two approaches in one system. The hybrid IDS is obtained by combining packet header anomaly detection (PHAD) which are anomaly-based IDSs with the misuse-based IDS Snort which is an open-source project.

Proposed System is divided into two modules as follows. Module 1 is the Packet header anomaly detection, whereas Module 2 is Designing the hybrid IDS (the process of combining the signature base IDS with anomaly base IDS).

A. Packet Header Anomaly Detection:

Packet header anomaly detector (PHAD) is the anomaly-based approach added to Snort as a pre-processor in this study. PHAD is different from other network-based anomaly detection systems for two reasons:

(1) It models protocols rather than the user behaviour because the majority of the attacks exploit protocol implementation bugs and can only be understood by detecting unusual input and output[1].

(2) It uses a time-based model, assuming a quick change in a short time in the network statistics.

PHAD reduces false alarm rate by flagging only the first anomaly as an alarm[3]. To apply time based modelling to anomaly detection with explicit training and test periods, an anomaly score is calculated using (t*n)/r, where n (number of packets including the related attribute field where an abnormal value is searched) and r (number of values accepted as normal) are counted during the training period, and where t is the time since the last anomaly. In this model normal values are the values seen at the time of training. Deviations from these values are detected at the test phase.

For ex-ample, suppose we are given the following training and test sequences: training (time 0-20): 000000000000000111122 and test (time 2127): 0122334. During training, the set of normal values 0, 1, 2 is recorded. The size of this set, r is 3 and the number of observations that is n are 21. If observations are made at unit time intervals starting at 0, then the last anomaly in training is 2 at time 19. The values 3, 3, and 4 at times 25, 26, and 27 in testing are anomalies because they are not in the training set. The anomaly score of the first 3 is (t*n)/r = (25-19) 21/3 = 42. The anomaly scores of the second 3 is (26-25) 21/3 = 7. The anomaly score of the 4 is (27-26) 21/3 = 7. The anomaly scores of 0, 1, and 2 are 0 because the values occur at least once in training.

The anomaly score of an instance with more than one anomalous attribute is P*t*n/r, where the summation is over the anomalous attributes. The attributes used by PHAD for anomaly detection. PHAD calculates anomaly scores for every packet and makes no distinction between incoming and outgoing traffic[2].

It models 33 attributes which correspond to packet header fields with 14 bytes. Fields smaller than one byte (such as TCP ags) are combined into one byte. Fields larger than four bytes (such as six byte Ethernet addresses) are split[3]. The attributes are as follows:

1. Ethernet header
2. IP header
3. TCP header
4. UDP header
5. ICMP header.

B. *Designing Hybrid IDS:*

Snorts pre-processor architecture has been used to combine PHAD with Snort. Pre-processors are engines which have the ability to give alerts, ignore or edit packages before they reach at the Snorts main detection engine[4,5]. PHAD was built into Snort as a pre-processor implementing the following steps:

1. Preprocessors source code file spp_phad.cpp was copied to the directory where snort.c lies in.

2. The header file spp_phad.h defining PHAD was inserted into plugbase.h which is used for applying pre-processors working order with the statement define PP_PHAD 131072. 131072 is the value for 217 and tells the compiler that Snort will be processed in the 18th place.

3. SetupPhad() function re-quired for initializing PHAD must be called from InitPreprocessors() function placed in
plugbase.c

4. As a last step, the project was recompiled in order to obtain Snort with PHAD pre-processor.

IV. (Expected) Analyses and Results

We have tested this HIDS system for some case studies; following section includes experimental review of these.

To solve this problem using the proposed approach, we consider the following the tables as sample initial data.

To test the performance of PHAD on Snort, we first ran Snort only with its own rule set (i.e. in Misuse mode), then only with the PHAD pre-processor (i.e. in Anomaly mode) and then with the rules and PHAD (i.e. HYBRID mode).

Here are the comparative results.

TABLE 1

Result Table

Snort Performance			
Day	**Without PHAD**	**Only PHAD**	**Rules+ PHAD**
Monday	637	20	660
Tuesday	552	8	562
Wednesday	699	34	737
Thursday	589	11	678
Friday	518	24	546

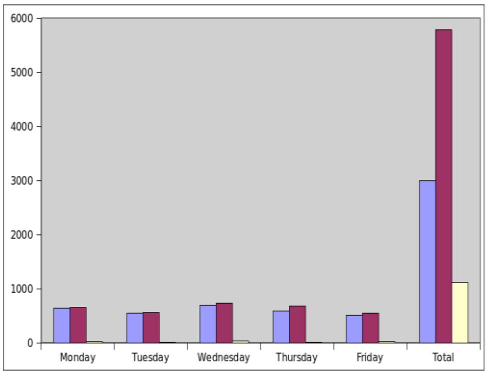

Figure 1: Quantitative Comparison

V. Conclusion

Signature-based systems can only detect attacks that are known before whereas anomaly-based systems are able to detect unknown attacks. Anomaly-based IDSs make it possible to detect attacks whose signatures are not included in rules. PHAD is added one by one to signature-based IDS namely Snort as a pre-processor in this study.

IDEVAL test bed which was created in MIT Lincoln Laboratories is used to evaluate the performance of new constructed hybrid IDS. Firstly, Snort is tested on IDEVAL data and the number of attacks it detects is found. Secondly, anomaly detection system, PHAD, is added to Snort as a pre-processor and this new version of Snort that is Hybrid IDS (Snort + PHAD) is tested on the same data. There is an increase in the number of attacks detected in this case. It is observed that number of attacks detected increases much more with the hybrid IDS. As seen from result, Snort, on its own, is able to detect 2995 attacks. After PHAD is added as a pre-processor, this number increases to 5791.

References

[1] K. Gkhan Ceylan M. Ali Aydn, A. Halim Zaim. “A hybrid intrusion detection system design for computer network security” Mar 14 2007. http://www.elsevier.com/locate/compeleceng

[2] Roesch M. Snort – lightweight intrusion detection for networks. In Proceedings of the 13th LISA conference of USENIX association; 1999, 2009.

[3] Bace R. Intrusion detection. Indianapolis, USA: Macmillan Technical Publishing; 2000.

[4]Snort Users Manual 2.6.1; 3 December 2006. http://www.snort.org/docs/snort_manual/2.6.1/snort_manual.pdf

[5] TMartin Roesch. Snort Users Manual. May 23, 2012. www.snort.org

[6] Data set, DARPA intrusion detection evaluation data set; 1999.