Computers and Electrical Engineering 35 (2009) 517–526

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

# A hybrid intrusion detection system design for computer network security

M. Ali Aydın *, A. Halim Zaim, K. Gökhan Ceylan

*Department of Computer Engineering, Faculty of Engineering, Istanbul University, 34320 Avcilar, Istanbul, Turkey*

ARTICLE INFO

*Article history:*
Received 14 March 2007
Received in revised form 15 July 2008
Accepted 30 December 2008
Available online 8 February 2009

*Keywords:*
Computer networks
Computer network security
Intrusion detection systems
Hybrid intrusion detection system

ABSTRACT

Intrusions detection systems (IDSs) are systems that try to detect attacks as they occur or after the attacks took place. 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 propose a hybrid IDS by combining the two approaches in one system. The hybrid IDS is obtained by combining packet header anomaly detection (PHAD) and network traffic anomaly detection (NETAD) which are anomaly-based IDSs with the misuse-based IDS Snort which is an open-source project.

The hybrid IDS obtained is evaluated using the MIT Lincoln Laboratories network traffic data (IDEVAL) as a testbed. 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.

© 2009 Elsevier Ltd. All rights reserved.

## 1. Introduction

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 [1].

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 [2,3].

In "2003CSI/FBI Computer Crime and Security Survey" it has been stated that the IDS usage in 1999 had been 42% and this ratio has become 73% in year 2003. This great improvement shows that IDSs are very important as security technologies. This paper is organized as follows: intrusion detection systems are described in Section 2, IDS types are explained in Section 3: Snort is the chosen system as misuse-based IDS; PHAD and NETAD are chosen as anomaly-based IDSs. Section 4 gives a brief

* Corresponding author. Tel.: +90 2124737070x17544; fax: +90 2124737044.
*E-mail addresses:* aydinali@istanbul.edu.tr (M.A. Aydın), ahzaim@istanbul.edu.tr (A.H. Zaim), kgceylan@istanbul.edu.tr (K.G. Ceylan).

0045-7906/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compeleceng.2008.12.005

description of the hybrid IDS we propose in this paper. The newly obtained hybrid IDS is evaluated in Section 5 and finally Section 6 includes conclusion.

## 2. Intrusion detection systems

Intrusion detection systems are hardware and software systems that monitor events occurred on computers and computer networks in order to analyze security problems. The number and severity of these attacks has been increasing continuously. Consequently IDSs have become an integral part of the security infrastructure of organizations.

Intrusions to computer networks are called as "attacks" and these attacks threaten the security of networks by violating privacy, integrity and accessibility mechanisms. Attacks can be originated from users who login to the computer using the Internet trying to gain *superuser* or *administrator* rights and other users who misuse the rights they have. IDSs automate monitoring and analyzing the attacks [1,2,4].

## 3. IDS types

There are two approaches to analyzing of events using IDSs. These are misuse-based and anomaly-based approaches. Misuse-based IDSs aim to distinguish events that violate system policy. Anomaly-based IDSs try analyzing abnormal activities and flag these activities as attacks. Both approaches have advantages and disadvantages when compared to each other [1,2,5].

Snort is the most commonly used signature-based intrusion detection system. Snort is a network intrusion detection system that runs over IP networks analyzing real-time traffic for detection of misuses [6]. Snort depends on a template-matching scheme and makes content analysis. It has the ability to flag alerts depending on pre-defined misuse rules and saves packets in tcpdump files or in plain text files. Snort is preferred to be used in academic research projects as it is an open-source tool and for this reason we have also chosen Snort as the signature-based intrusion detection system in our work.

Anomaly detection based intrusion detection systems are separated into many sub-categories in the literature including statistical methodologies [7–10], data mining [11,12], artificial neural networks [13], genetic algorithms [14] and immune systems [15,16]. Among these sub-categories, statistical methods are the most commonly used ones in order to detect intrusions by analyzing abnormal activities occurring in the network. PHAD [17] and NETAD [18] statistical methods are chosen as the anomaly-based intrusion detection systems in this paper. We have implemented a hybrid IDS by mounting anomaly-based IDSs PHAD and NETAD to Snort as a preprocessor. PHAD is different than the other conventional network-based anomaly detection systems for two reasons. First, it models protocols rather than user behaviors. Second, it uses a time-based model depending on the rapid change of network statistics in short term. PHAD flags only the first anomaly it detected as an alert even if there is a series of the same anomaly recurring. This feature of PHAD helps reducing the number of false alerts. NETAD, models single packets like PHAD, uses dynamic-conditioned rules like ALAD [19], and rule verification like LERAD [20]. Its greatest contribution is modeling values that are not new.

### 3.1. Misuse-based IDSs

Misuse detectors analyze system activities and try to find a match between these activities and known attacks having definitions or signatures introduced to the system beforehand [1,2,21].

*Advantages:*

- Misuse detectors are very efficient in detecting attacks without signaling false alarms (FA).
- Misuse detectors can quickly detect specially designed intrusion tools and techniques.
- Misuse detectors provide systems administrators an easy to use tool to monitor their systems even if they are not security experts.

*Disadvantages:*

- Misuse detectors can only detect attacks known beforehand. For this reason the systems must be updated with newly discovered attack signatures.
- Misuse detectors are designed to detect attacks that have signatures introduced to the system only. When a well-known attack is changed slightly and a variant of that attack is obtained, the detector is unable to detect this variant of the same attack.

Misuse-based IDS used in our hybrid IDS is the open-source project Snort.

#### 3.1.1. Snort

Martin Roesch, a software engineer working on the computer security topics, has developed Snort in 1990 in order to detect attacks targeting his home network. Snort is a fast, signature-based and open-source IDS. It produces alarms using misuse rules defined previously. It uses binary tcpdump-formatted files or plain text files to capture network packets. Tcpdump is a software program that captures network packets from computer networks and stores them in tcpdump-formatted files. Snort is rule-based and it has a language to define new rules. Snort is an open-source project and it has an architecture making it possible to integrate new functionalities at the time of compilation [6,22–24].

Snort consists of the following four components [24,25]:

(1) *Packet capture/decode engine:* Snort's packet-capturing engine uses the *libpcap* packet-capturing library written in Lawrence Berkeley National Laboratories. Captured packets are processed by decoding engine and decoded packets become compliant with the network-level protocols. Resultant packets are re-decoded for upper-level protocols that are TCP and UDP.
(2) *Preprocessor plug-ins:* Packets are passed through a number of preprocessors. This step aims investigating and processing packets before they are passed to the detection engine. Every other preprocessor examine the packets for a different attribute and make a decision to pass the packet to the detection engine without making any modification, modifying and then passing it to detection engine or not passing and generating an alert for the packet.
(3) *Detection engine:* Detection engine tests packets for a number of attributes stated in Snort rules definition file. Detection plug-ins provide extra detection functions.
(4) *Output plug-ins:* These plug-ins accept alarms generated from detection engine, preprocessors or decoding engine.

An illustration of Snort's packet processing is given in Fig. 1 [5].

Snort is a rule-based network intrusion detection system (N-IDS). It has a flexible rule defining language that lets anyone to change existing rules or adding new rules to the IDS. Every rule consists of two logical parts: the rule header and rule options. Rule header has five sections; rule actions (action to be taken when an intrusion is detected), end-to end source and destination information (source and destination IP addresses and port numbers depending on the protocol), and direction of traffic and protocol type (TCP, UDP, or ICMP).

Rule options consist of various conditions that help deciding whether the mentioned misuse operation has occurred or not. A sample Snort rule is shown in Fig. 2. The first field of every rule is the action field. This field can have the following values: log, alert, pass, activate, or dynamic. When the input value matches the criteria, these actions are taken as a response. The selected action in Fig. 2 is "alert". This states that, if an entry matches with the mentioned criteria, an alert will be created. The next field holds the protocol information. Valid values of this field can be TCP, UDP, or ICMP. The protocol in our example is TCP. The third and the fourth fields hold source addresses; the first part stands for IP address and the second part is the port number. If this field has the value "any any", it means that the packets may be originating from any IP address and any TCP port. In the case where protocol value is ICMP, no port value is used as this field is meaningful for only TCP and UDP.

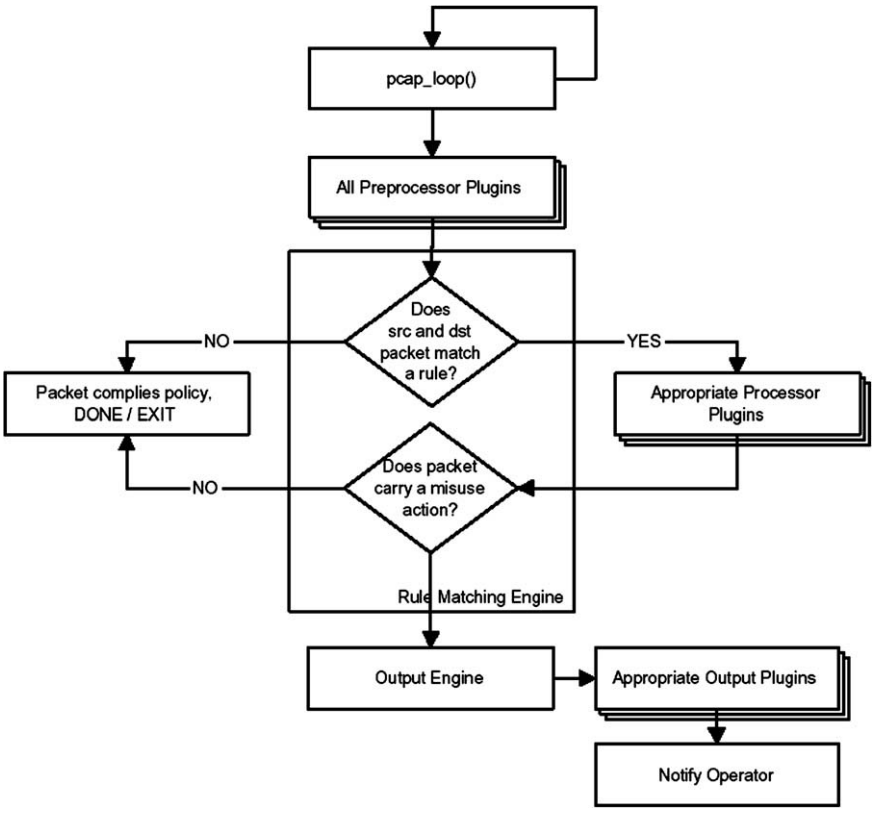


**Fig. 1.** Packet flow through Snort.

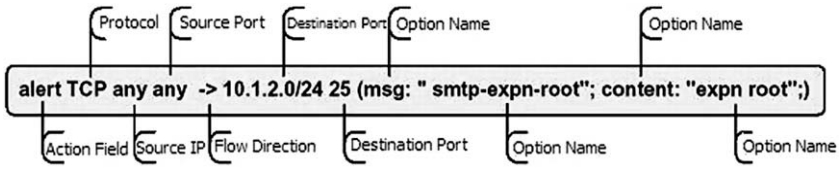


**Fig. 2.** Snort rule structure.

The fifth field shows the flow direction of the information. The sixth and the seventh fields hold destination addresses. The example destination IP address is given as 10.1.2.0/24, which matches all the IP addresses in a C class network. In this example, TCP destination port is set as 25. Port 25 is used for simple mail transfer protocol (SMTP).

Following the destination address, there is an options list written in parenthesis. Every option consists of an option name, an option value if exists and a semicolon indicating the end of that option. The first option shown in Fig. 2 is "msg" and it is used to state the action message. "Content" is the second option and states a template-matching criterion. In the sample shown in Fig. 2, "expn root" string is searched for in the input data. If a match is found in TCP data segment, the condition is met.

All of the below criteria must be met in order to make the sample rule shown in Fig. 2 produce an alert:

- The entry must be a TCP packet.
- The entry may be originating from any IP address and any TCP port.
- The entry must destine the 10.1.2.0/24 network and port number 25 of the computer located in this network.
- The entry must include "expn root" string.

### 3.2. Anomaly-based intrusion detection systems

Anomaly detectors detect behaviors on a computer or computer network that are not normal. According to this approach, behaviors deviating from behaviors assumed as "normal" are thought to be attacks and anomaly detectors compute the deviation in order to detect these attacks. Anomaly detectors construct profiles of users, servers and network connections using their normal behaviors. These profiles are produced using the data that is accepted as normal. After the profile construction, detectors monitor new event data, compare the new data with obtained profile and try to detect deviations. These deviations from normal behaviors are flagged as attacks. Pros and cons of anomaly-based approach are as follows [1,21,26]:

*Advantages:*

- Anomaly-based IDSs, superior to signature-based ones, are able to detect attacks even when detailed information of the attack does not exist.
- Anomaly-based detectors can be used to obtain signature information used by misuse-based IDS.

*Disadvantages:*

- Anomaly-based IDSs generally flag many false alarms (FA) just because user and network behavior are not always known beforehand.
- Anomaly-based approach requires a large set of training data that consist of system event log in order to construct normal behavior profile.

Anomaly-based IDSs used in our hybrid IDS are PHAD and NETAD.

#### 3.2.1. Packet header anomaly detector (PHAD)

Packet header anomaly detector (PHAD) is the first anomaly-based approach added to Snort as a preprocessor in this study. PHAD is different from other network-based anomaly detection systems for two reasons. First, it models protocols rather than the user behavior because the majority of the attacks exploit protocol implementation bugs and can only be understood by detecting unusual input and output. Second, 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. To apply time-based modeling to anomaly detection with explicit training and test periods, an anomaly score is calculated using $tn/r$, 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 example, suppose we are given the following training and test sequences: training (time 0–20): 00000000000000000111122 and test (time 21–27): 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, $n$, 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 $tn/r = (25 - 19)21/3 = 42$. 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 $\sum tn/r$, where the summation is over the anomalous attributes [17,27].

*3.2.1.1. The attributes used by PHAD for anomaly detection.* PHAD calculates anomaly scores for every packet and makes no distinction between incoming and outgoing traffic. It models 33 attributes which correspond to packet header fields with 1–4 bytes. Fields smaller than one byte (such as TCP flags) are combined into one byte. Fields larger then four bytes (such as six byte Ethernet addresses) are split. The attributes are as follows: Ethernet header, IP header, TCP header, UDP header, and ICMP header [27].

*3.2.2. Network traffic anomaly detector (NETAD)*

Network traffic anomaly detector (NETAD) is the second anomaly-based approach added to Snort as a preprocessor in this study. NETAD also models packets as PHAD does and it operates in two phases: the first phase is the filtering of incoming client sessions to distinguish beginning of sessions. The second phase is the modeling phase. Filtering phase eliminates the traffic up to 98–99%. This elimination simplifies the traffic for the modeling phase. Only the traffic data, which evidence of attacks are included in, is passed to the modeling phase. The second phase models nine types of packets. Totally $9 \times 48 = 432$ rules are available.

The nine mostly used protocols chosen are as follows [18,27]:

- All IP packets.
- All TCP packets (if protocol = TCP(6)).
- TCP SYN (if TCP and flags = SYN(2)).
- TCP data (if TCP and flags = ACK(16)).
- TCP data for port numbers between 0 and 255 (if TCP and ACK and DP1 (high order bit of destination port) = 0).
- Telnet (if TCP and ACK and DP1 = 0 and DP0 = 21).
- FTP (if TCP and ACK and DP1 = 0 and DP0 = 23).
- SMTP (if TCP and ACK and DP1 = 0 and DP0 = 25).
- HTTP (if TCP and ACK and DP1 = 0 and DP0 = 80).

Anomaly score designated to every packet is the sum of the anomaly scores reported by 432 rules. Anomaly score is calculated using $tn_a(1 - r/256)/r + t_i n\ (n_i + r/W)$ formula where $n_a$ is the number of normal packets from where last training anomaly seen to the end of training period. This approach gives more weight to the rules that produce fewer false alarms. Constant coefficient 256 makes it possible to give less weight to the rules producing available byte values so that repeating distributions are discarded quickly. $t_i$ is the time when $i$ is last seen and $n_i$ is the number of times $i$ is seen during the training period. $W = 256$ is a constant coefficient obtained experimentally [27].

## 4. Designing the hybrid IDS

### 4.1. Combining PHAD and NETAD to signature-based IDS Snort

Snort's preprocessor architecture has been used to combine PHAD and NETAD with Snort. Preprocessors are engines which have the ability to give alerts, ignore or edit packages before they reach at the Snort's main detection engine. PHAD was built into Snort as a preprocessor implementing the following steps:

- Preprocessor's source code file "spp_phad.cpp" was copied to the directory where "snort.c" lies in.
- The header file "spp_phad.h" defining PHAD was inserted into "plugbase.h" which is used for applying preprocessors 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.
- "SetupPhad()" function required for initializing PHAD must be called from "InitPreprocessors()" function placed in "plugbase.c".
- As a last step, the project was recompiled in order to obtain Snort with PHAD preprocessor.

NETAD was built into Snort as a preprocessor implementing the following steps:

- Preprocessor's source code file "spp_netad.cpp" was copied to the directory where "snort.c" lies in.
- The header file "spp_netad.h" defining NETAD was inserted into "plugbase.h" which is used for applying preprocessors working order with the statement "#define PP_NETAD 262144". 262144 is the value for 218 and tells the compiler that Snort will be processed in the 19th place just after PHAD.

- "SetupNetad()" function required for initializing NETAD must be called from "InitPreprocessors()" function placed in "plugbase.c".
- As a last step the project was recompiled in order to obtain Snort with NETAD preprocessor.

## 5. Evaluation of the hybrid IDS

Scientific advances rely on reproducibility of results so that they can be independently validated and compared. Much of the evaluation in intrusion detection has been based on proprietary data and results are generally not reproducible. One of the main problems of releasing data stems from privacy concerns. To reduce this problem, Lincoln Laboratory (LL), under sponsorship of DARPA, created the IDEVAL datasets that serves as an evaluation benchmark [28].

The goal of the 1998 DARPA intrusion detection system evaluation was to collect and distribute the first standard corpus for evaluation of intrusion detection systems. 1998 DARPA intrusion detection system evaluation was generated and recorded on a network which simulated an operational network connected to the Internet. Automatically generated traffic used more than 20 network services, including dns, finger, ftp, http, ident, ping, pop, smtp, snmp, telnet, time, and X. A Windows NT host is added to three UNIX based target hosts and twelve new Windows NT attacks were added in 1999 along with stealthy versions of many 1998 attacks, new inside console-based attacks and six new UNIX attacks. Fifty-six different attack types were used in the evaluation [29]. Block diagram of the network from which intrusion data is captured is illustrated in Fig. 3 [30].

There are four main "victim" machines, running SunOS, Solaris, Linux, and Windows NT. Traffic generators simulate hundreds of other hosts and users running various applications and Internet. The evaluation data set is collected from the four victim machines and from two network sniffers, an "inside" sniffer between the router and the victims, and an "outside" sniffer between the router and the Internet. The 1999 evaluation had two phases separated by about three months. During the first phase, participants were provided with three weeks of data. The first and third weeks contained no attacks, and could be used to train anomaly detection systems. During the second phase, participants were provided with two weeks of test data (weeks 4 and 5) containing 201 unlabeled instances of 58 attacks, 40 of which were not in the training data. Attacks are classified by category (probe, DOS, R2L, U2R), the type of data examined (inside sniffer, outside sniffer, BSM, audit logs, file system dumps, or directory listings), victim operating system (SunOS, Solaris, Linux, or NT), and whether the attack is new [31–33].

### 5.1. Performance of Snort on IDEVAL data

Snort is tested on IDEVAL dataset (fourth and fifth weeks including attack) and the detected attacks are listed day by day. IDEVAL dataset files used for the test and the days each file belongs to are shown in Table 1. These files have been downloaded from [33].

Attacks detected on a daily bases are shown in Fig. 4. Snort has detected 27 attacks out of 201 attacks available in IDEVAL data.

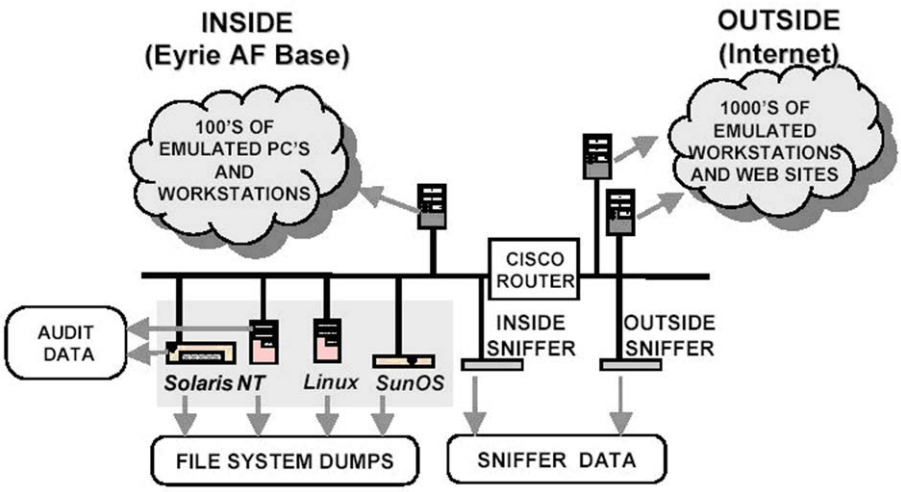


**Fig. 3.** Block diagram of IDEVAL, 1999 testbed.

**Table 1**
Fourth and fifth weeks tcpdump files from IDEVAL dataset and the days they belong to.

| File name | Day |
|---|---|
| in41 | 29 March |
| in43 | 31 March |
| in44 | 1 April |
| in45 | 2 April |
| tn51 | 5 April |
| in52 | 6 April |
| in53 | 7 April |
| in54 | 8 April |
| in55 | 9 April |

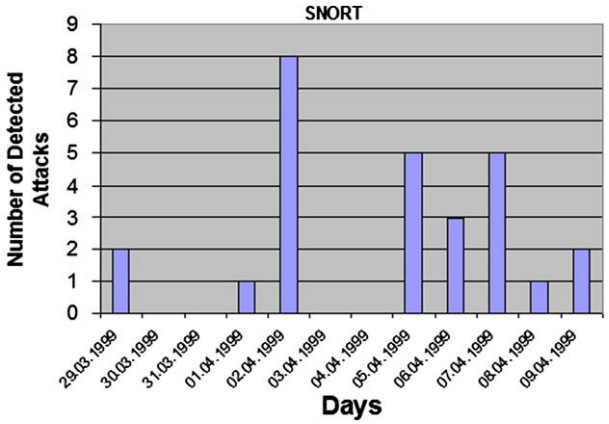


**Fig. 4.** Attacks detected by Snort on a daily bases.

### 5.2. Performance of Snort + PHAD on IDEVAL data

As the third week data included in IDEVAL dataset is used for training anomaly-based IDSs, files belonging to this week are used to train PHAD. Third week files and the days they belong to are listed in Table 2 [33].

Attacks detected by Snort and PHAD on their own and by the obtained hybrid system (Snort + PHAD) are shown in Fig. 5. It is obvious from the graphic that integrating PHAD into Snort as a preprocessor increased the number of attacks detected. This shows the contribution of newly added preprocessor PHAD to Snort IDS. Number of attacks detected by Snort increases from 27 to 51 in Snort + PHAD version of the IDS. The reason for this increase is PHAD making anomaly detection on packet headers and detecting attacks that Snort is unable to detect using rule definition files.

### 5.3. Performance of the hybrid system (Snort + PHAD + NETAD) on IDEVAL data

Attacks detected by Snort, PHAD and NETAD on their own and by the obtained hybrid system (Snort + PHAD + NETAD) are shown in Fig. 6.

It is clear from Fig. 6 that after NETAD is added as a preprocessor Snort becomes a more powerful IDS. This shows the contribution of the newly added preprocessor NETAD to Snort IDS. The number of attacks detected by Snort + PHAD increases from 51 to 146 in Snort + PHAD + NETAD version of the IDS (hybrid IDS). The reason for this increase is NETAD making anomaly detection using dynamic rules and detecting attacks that Snort + PHAD is unable to detect.

**Table 2**
Third week tcpdump files from IDEVAL dataset and the days they belong to.

| File name | Day |
|---|---|
| in31 | 15 March |
| in32 | 16 March |
| in33 | 17 March |
| in34 | 18 March |
| in35 | 19 March |
| in36 | 22 March |
| in37 | 23 March |

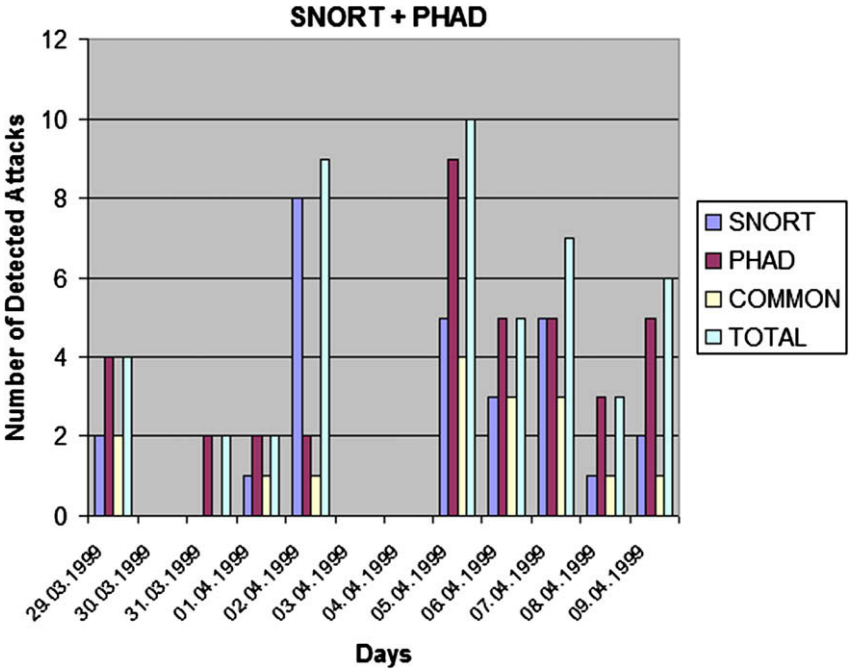


**Fig. 5.** Attacks detected by Snort+PHAD on a daily bases.

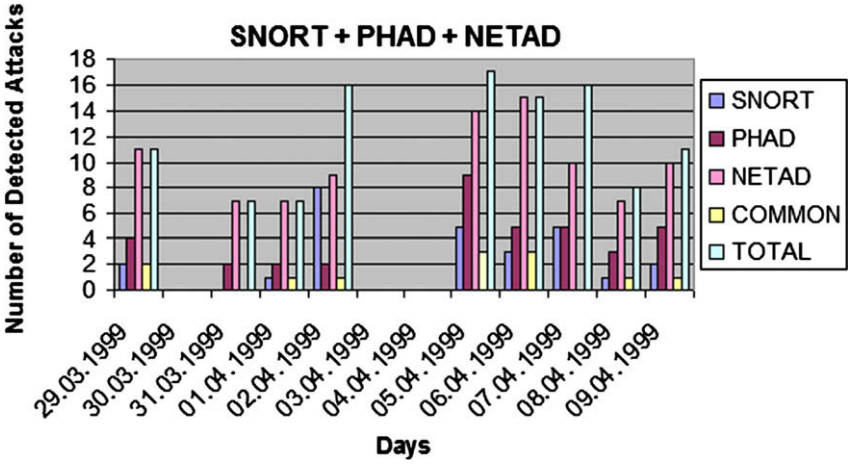


**Fig. 6.** Attacks detected by Snort+PHAD+NETAD on a daily bases.

## 6. 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 rule files. PHAD and NETAD are added one by one to signature-based IDS namely Snort as a preprocessor in this study. IDEVAL testbed 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 preprocessor and this new version of Snort (Snort + PHAD) is tested on the same data. There is an increase in the number of attacks detected in this case. Finally another anomaly detection system, NETAD, is added to PHAD version of Snort as a second preprocessor. This final system is called the hybrid IDS (Snort + PHAD + NETAD) and it is also tested on IDEVAL data. It is observed that number of attacks detected increases much more with the hybrid IDS. As seen from Fig. 7, Snort, on its own, is able to detect 27 attacks. After PHAD is added as a preprocessor, this number increases to 51 and finally after NETAD is added as a preprocessor the number of attacks detected increases up to 146.

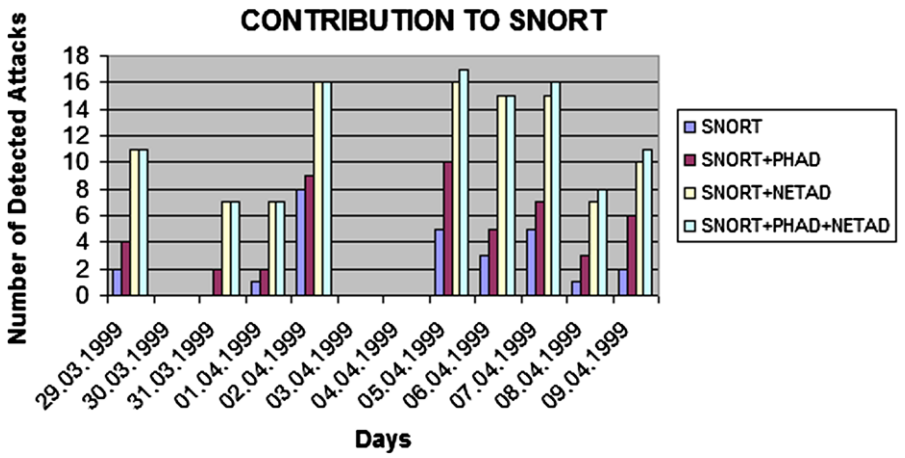


**Fig. 7.** Attacks detected by the hybrid IDS on a daily bases.

As a result it can be mentioned that combining PHAD and NETAD as a preprocessor which are anomaly-based systems with the signature-based IDS Snort, contributes to intrusion detection positively. The hybrid IDS is said to be more powerful than the signature-based on its own because it uses the advantages of anomaly-based approach for detecting unknown attacks.

## Acknowledgement

This work is supported by the Research Fund of Istanbul University, Turkey. Project Number: 407/13092005.

## References

[1] Bace R. Intrusion detection. Indianapolis, USA: Macmillan Technical Publishing; 2000.
[2] Scarfone K, Mell P. Guide to intrusion detection and prevention systems (IDPS). NIST Special Publication 800-94; 2007.
[3] Bace R. An introduction to intrusion detection and assessment for system and network security management. ICSA Intrusion Detection Systems Consortium Technical Report; 1999.
[4] Bace R, Mell P. Intrusion detection systems. NIST Special Publication on Intrusion Detection Systems; 2001, SP 800-31.
[5] Dayioğlu B. Use of passive network mapping to enhance network intrusion detection. Thesis (Master), The Graduate School of Natural and Applied Sciences of the Middle East Technical University; 2001.
[6] Roesch M. Snort – lightweight intrusion detection for networks. In Proceedings of the 13th LISA conference of USENIX association; 1999.
[7] Denning DE. An intrusion-detection model. IEEE Trans Software Eng 1997;13(2):222–32.
[8] Javitz HS, Valdes A. The SRI IDES statistical anomaly detector. In Proceedings IEEE symposium on security and privacy, Oakland, CA; May 1991. p. 316–26.
[9] Neumann PG, Porras PA. Experience with EMERALD to date. In First USENIX workshop on intrusion detection and network monitoring, Santa Clara, CA; 11–12 April 1999. p. 73–80.
[10] Lankewicz L, Benard M. Real time anomaly detection using a nonparametric pattern recognition approach. In proceedings of the seventh annual computer security applications conference, San Antonio, TX; 2–6 December 1991. p. 80–9.
[11] Noel S, Wijesekera D, Youman C. Modern intrusion detection, data mining, and degrees of attack guilt, in applications of data mining in computer security. Kluwer Academic Publisher; 2002.
[12] Lee W, Stolfo S. Data mining approaches for intrusion detection. In Proceedings of the seventh USENIX security symposium (SECURITY'98), San Antonio, TX; 26–29 January 1998.
[13] Debar H, Becker M, Siboni, D. A neural network component for an intrusion detection systems. In Proceedings of the 1992 IEEE symposium on security and privacy, Oakland, CA; 4–6 May 1992. p. 240–50.
[14] Ludovic M. GASSATA: a genetic algorithm as an alternative tool for security audit trails analysis. In First international workshop on the recent advances in intrusion detection, Louvain-la-Neuve, Belgium; 14–16 September 1998.
[15] Kim J, Bentley P. The artificial immune model for network intrusion detection. In Seventh European congress on intelligent techniques and soft computing (EUFIT'99), Aachen, Germany; 13–19 September 1999.
[16] Warrender C, Forrest S, Pearlmutter B. Detecting intrusions using systems call: alternative data models. In Proceedings of the 25th IEEE symposium on security and privacy, Oakland, CA; 9–12 May 1999. p. 133–45.
[17] Mahoney MV, Chan PK. PHAD: packet header anomaly detection for identifying hostile network traffic. Florida Institute of Technology Technical Report, CS-2001-04.
[18] Mahoney MV. Network traffic anomaly detection based on packet bytes. In Proceedings of ACM-SAC; 2003.
[19] Mahoney MV, Chan PK. Learning nonstationary models of normal network traffic for detecting novel attacks. In Proceedings of eighth international conference on knowledge discovery and data mining; 2000. p. 376–85.
[20] Mahoney MV, Chan PK. Learning models of network traffic for detecting novel attacks, Florida Institute of Technology Technical Report, CS-2002-08; 2003.
[21] Mukherjee B, Heberlein LT, Levitt KN. Network intrusion detection. IEEE Network 1994;8(3):26–41.
[22] Russell R. Snort intrusion detection 2.0. Rockland, MA: Syngress Publishing, Inc.; 2003.
[23] JacobsonV, Mccanne S. Tcpdump: a tool for network monitoring and data acquisition. Berkeley, CA: Network Research Group, Lawrence Berkeley National Laboratory. <ftp://ftp.ee.lbl.gov/tcpdump.tar.z>.
[24] Snort Users Manual 2.6.1; 3 December 2006. <www.snort.org/docs/snort_manual/2.6.1/snort_manual.pdf>.

[25] Rehman RU. Intrusion detection systems with snort. Upper Saddle River, New Jersey: Publishing as Prentice Hall PTR; 2003.
[26] Zhang J, Zulkernine M. Anomaly based network intrusion detection with unsupervised outlier detection. In Symposium on network security and information assurance – proceedings of the IEEE international conference on communications (ICC), Istanbul, Turkey; June 2006.
[27] Mahoney MV. A machine learning approach to detecting attacks by identifying anomalies in network traffic. Thesis (PhD), Florida Institute of Technology; 2003.
[28] Mahoney MV, Chan PK. An Analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection. Recent advances in intrusion detection. In Sixth international symposium, Raid 2003, Pittsburgh, PA, USA; 8–10 September 2003. p. 220–39.
[29] Kendall K. A database of computer attacks for the evaluation of intrusion detection systems. Master Thesis, Massachusetts Institute of Technology, Lexington, MA; 1999.
[30] Lippman R, Haines JW, Fried DJ, Korba J, Das K. The 1999 DARPA off-line intrusion detection evaluation. Comput Networks 2000;34(4):579–95.
[31] Lippman R, Haines JW, Fried DJ, Korba J, Das K. Analysis and results of the 1999 DARPA off-line intrusion detection evaluation. In Proceedings of the third international workshop on recent advances in intrusion detection, Toulouse, France; 2–4 October 2000. p. 162–82.
[32] Haines JW, Lippman R, Fried DJ, Zissman MA, Tran E, Boswell SB. 1999 DARPA intrusion detection evaluation: design and procedures. MIT Lincoln Laboratory Technical Report, TR-1062, Massachusetts, USA; 2001.
[33] Data set, DARPA intrusion detection evaluation data set; 1999. <http://www.ll.mit.edu/IST/ideval/data/1999/1999_data_index.html>.