A Hybrid Snort-Negative Selection Network Intrusion Detection Technique

Tarek M. Mahmoud
Department of Computer Science, Faculty of Science, Minia University, Egypt

Abdelmgeid A. Ali
Department of Computer Science, Faculty of Science, Minia University, Egypt

Hussein M. Elshafie
Department of Computer Science, Faculty of Science, Minia University, Egypt

ABSTRACT
Network Intrusion Detection Systems (NIDSs) are systems that monitor computer networks to detect, identify and prevent the malicious events, which attempt to compromise the integrity, confidentiality or availability of computer networks. The NIDS may be classified according to the detection technique into two types, the "Signature-Based" and "Anomaly-Based" NIDS. In order to increase the efficiency of the NIDS, a hybrid signature-anomaly NIDS based on both snort and negative selection algorithm is proposed. To evaluate the efficacy of the proposed system the 1999 DARPA data set is used. The experimental results show that the performance of the proposed system is more efficient than using snort on its own.

General Terms
Security, Network Intrusion Detection System (NIDS)

Keywords
Signature Based, Anomaly Based, Snort, Negative Selection

1. INTRODUCTION
The wide use and development of networks brings people more and more convenience. However, it also induces many security problems to people's life and work. Security is a big issue for all networks in recent enterprise environment. Hackers and intruders have made many successful attempts to bring down high-profile company networks and Web services. There are many methods to secure the network infrastructure and communication over the Internet, among them is the use of firewalls [1]. 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. 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 [2]. This is the situation where Intrusions Detection Systems (IDSs) are in charge. IDSs are used to stop attacks, recover from them with the minimum loss and analyze the security problems so that they are not repeated [3]. Several IDSs are suggested to protect the networks security. According to the used detection method, IDSs can be classified as signature based (misuse) in which known attacks can be classified easily and anomaly based which can identify newly attacks. Signature-based detection is very effective at detecting known threats but largely ineffective at detecting unknown threats. On the other hand, anomaly based detection is very effective in detecting unknown attacks. The main disadvantage of anomaly based detection is that it produces many false positives alarm. Snort is a signature-based intrusion detection used to audit network packets and compare those packets with the database of known attack signature. The negative selection algorithm (NSA) is anomaly based intrusion detection technique used to create detectors to detect the newly attacks.

In this paper, a hybrid signature anomaly based intrusion detection system is proposed. The proposed detection system uses both Snort and negative selection algorithm. The conducted experimental results shows that using the proposed detection method is more powerful compared with using Snort detection alone.

This paper is organized as follows: related works are mentioned in Section 2, intrusion detection systems are described in Section 3, and Section 4 describes the signature based IDS in some detail. In section 5 the anomaly based IDS are described in some detail. The proposed hybrid NIDS is implemented in Section 6. Evaluation strategy and experimental results will be presented in section 7. Finally, conclusion and future work will be shown in section 8.

2. RELATED WORKS
Several studies were carried out on IDS.

Shen and Wang [4] proposed an artificial immune system based network intrusion detection scheme. An optimized feature selection and parameter quantization algorithms were defined. The complexity issue was addressed in the design of the algorithms.

Jinyin and Dongyong [5] presented several new methods are adopted to improve the performance of NSA, and finally cooperative coevolution detector generation model has been constructed as a novel structure for IDS.

A. Aziz et al. [6] presented an approach for detecting network traffic anomalies using detectors generated by a genetic algorithm with deterministic crowding Niching technique. Particularly, the suggested approach is inspired by the negative selection mechanism of the immune system that can detect foreign patterns in the complement (non-self) space.

Zhou et al. [7] explained how Snort implements the intrusion detection, which includes building the compiling environment and analyzing the work-flow and rule tree.

Kumar and Joshi [8] designed a system with the help of Entropy based technique and integrating with real time system Snort so that it can have advantages of both techniques. Entropy is one of the anomaly detection technique used in intrusion detection.

Peng and HongJie [9] proposed a new design idea that combining the Snort with NTOP, and then introduces the implementation of the system; lastly it is verified by experiment. The result proves that the instruction behavior can be detected effectively by this system. NTOP is a flexible, complete functions tool of solving the problem of local network by monitoring. Through the analysis of network

dataflow, it can determine the existing problems of network, such as the bottleneck effect or slow performance.

Hussein et al. [10] proposed hybrid IDS by integrated signature based (Snort) with anomaly based (Naive Bayes) to enhance system security to detect attacks.

3. INTRUSION DETECTION SYSTEMS

IDS constitutes a primary component for securing computing infrastructures. An IDS monitors activity and seeks to identify evidence of ongoing attacks, intrusion attempts, or violations of the security policies. IDSs have evolved since the first model proposed in the late 1980s[11].

The Defense Advanced Research Projects Agency (DARPA) Common Intrusion Detection Framework (CIDF) splits intrusion detection systems to four logical components as depicted in Figure 1.

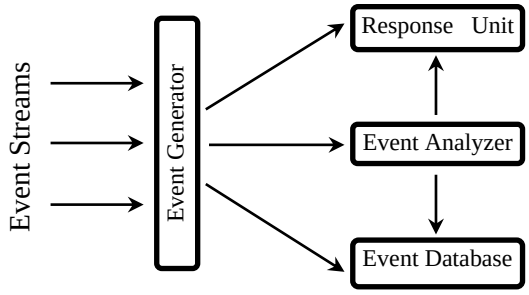

Figure 1: Common Intrusion Detection Framework Model

Event Generators are the sensors of the IDS and their purpose is to collect data from the event stream and to provide it (raw or after some pre-processing) to the other components. Event Databases are the places where events and intermediate information are stored for future analysis. The event analyzer is the core unit of any IDS because it contains the decision algorithm responsible for discrimination between intrusive and normal (non-intrusive) events. Using some kind of model stored in an internal knowledge base, the analyzer distinguishes the former from the latter, and communicates the decision as output. A consequence of this model is that all IDSs are based on the assumption that the input stream contains enough information to distinguish between intrusive behaviors and non-intrusive ones. The distinction between the two classes can be considered as the first step in the intrusion detection process. Response Units use some form of countermeasure that can block the detected attack or modify the environment to prevent similar action to happen again in the future [12].

Depending on the nature of the event stream, one can distinguish network-based from host-based intrusion detection systems. Each has a distinct approach for monitoring, securing data and systems. A Host Intrusion Detection System (HIDS) is a software agent that can be installed in a particular computer in order to monitor and analyze events on that particular host to detect any suspicious behavior [11]. It is reasonably easy for a host IDS to spot when an application crashes, when it tries to open a suspicious file, or it attempts to open a connection through the network. Moreover, these systems can also detect intrusions where a legitimate user abuses his/her privileges, trying to perform some illegal action. In this case the event stream consists most of the time of system call sequences and application logs.

Network-based IDSs (NIDSs) differ from HIDS because they are placed on a network segment (or connected to a monitor port in a switch) where they can monitor the whole traffic directed towards one or more computers. In this case the sensor is basically a sniffer and the event stream is composed of raw network packets. The major advantage is that a single system can be used to monitor the whole network (or part of it), without the need of installing a dedicated software sensor on each host[12]. NIDSs are responsible for protection of the entire environment of the network from the intrusion. This task asks for full knowledge of the system status and monitoring both the components of the network and the transactions between them. NIDSs are capable of accessing the network routers and instructing them to perform tasks. Using this feature, system can ask the router to disconnect a terminal or a subnet that has become a security threat [13].

IDS technologies use many methodologies to detect incidents. The primary classes of detection methodologies are signature-based and anomaly-based. Signature based detection techniques match the signatures of already known attacks that are stored into the database to detect the attacks in the computer system. Anomaly based detection techniques consist of defining, what is the normal (allowed) behavior of the system and then flagging as intrusive any event that falls outside the "normal" boundaries, or that is different enough from a statistical perspective.

4. SIGNATURE-BASED DETECTION

A signature is a pattern that corresponds to a known threat. Signature-based detection is the process of comparing signatures against observed events to identify possible incidents. Signature-based detection is very effective at detecting known threats but largely ineffective at detecting previously unknown threats. Signature-based detection is the simplest detection method because it just compares the current unit of activity, such as a packet or a log entry, to a list of signatures using string comparison operations [13]. One of the common signature-based IDS is the open-source project Snort.

4.1 Snort

Snort is logically divided into multiple components. These components work together to detect particular attacks and to generate output in a required format from the detection system. A Snort IDS consists of the following major components:

- Packet Decoder
- Preprocessors
- Detection Engine
- Logging and Alerting System
- Output Modules

Figure 2 shows how these components are arranged. Any data packet coming from the Internet enters the packet decoder. On its way towards the output modules, it is either dropped, logged or an alert is generated.

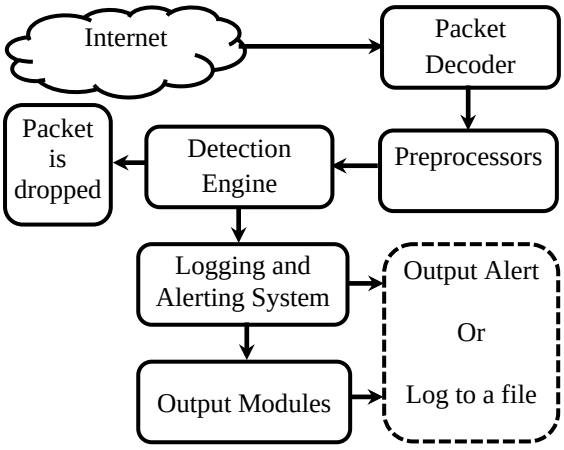

Figure 2 Components of Snort

Packet Decoder
The packet decoder takes packets from different types of network interfaces and prepares the packets to be preprocessed or to be sent to the detection engine.

Preprocessors
Preprocessors are components or plug-ins that can be used to arrange or modify data packets before the detection engine does some operation to find out if the packet is being used by an intruder. Some preprocessors also perform detection by finding anomalies in packet headers and generating alerts.

Preprocessors are also used for packet defragmentation. When a large data chunk is transferred to a host, the packet is usually fragmented. On IDS, before you can apply any rules or try to find a signature, you have to reassemble the packet. For example, half of the signature may be present in one segment and the other half in another segment. To detect the signature correctly you have to combine all packet segments.

The Detection Engine
The detection engine is the most important part of Snort. Its responsibility is to detect if any intrusion activity exists in a packet. The detection engine employs Snort rules for this purpose. The detection engine is the time-critical part of Snort. Depending upon how powerful your machine is and how many rules you have defined, it may take different amounts of time to respond to different packets. If traffic on the network is too high when Snort is working in NIDS mode, some packets may be dropped to obtain a true real-time response.

Logging and Alerting System
Depending upon what the detection engine finds inside a packet, the packet may be used to log the activity or generate an alert. Logs are kept in simple text files.

Output Modules
Output modules or plug-ins can do different operations depending on the wanted method to save output generated by the logging and alerting system of Snort. Basically these modules control the type of output generated by the logging and alerting system. Other tools can also be used to send alerts in other formats such as e-mail messages or viewing alerts using a Web interface [14].

5. ANOMALY-BASED DETECTION

Anomaly-based detection is the process of comparing definitions of what activity is considered normal against observed events to identify significant deviations [15]. An IDS using anomaly-based detection has profiles that represent the normal behavior of such things as users, hosts, network connections, or applications. Throughout this paper, the focus is on network connection. The profiles are developed by monitoring the characteristics of typical activity over a period of time. The major benefit of anomaly-based detection methods is that they can be very effective at detecting previously unknown threats [16]. An initial profile is generated over a period of time (typically days, sometimes weeks) called training period. Profiles for anomaly-based detection can either be static or dynamic. Once generated, a static profile is unchanged unless the IDS is specifically directed to generate a new profile. A dynamic profile is adjusted constantly as additional events are observed. Because network activities are changed over time, the corresponding measures of normal behavior also change; a static profile will eventually become inaccurate, so it needs to be regenerated periodically. Dynamic profiles do not have this problem, but they are susceptible to evasion attempts from attackers [17].

Anomaly-based IDPS products often produce many false positive rates because the users sometimes perform new and different activities, making it very hard to build a model of normal that is broad enough to encompass such actions but not so broad as to also mistakenly count attack patterns as normal [18].

Through the research on intrusion detection system and biological immune system, they look similar. The main goal of intrusion detection and immune system is to identify normal data and eliminate abnormal data. Hence, immune theory has good use for reference of network intrusion system. It is antibody that detects abnormal cells in immune system, so, the key to build the network intrusion system based on immune theory is to negative selection algorithm (NSA) to generate intrusion detectors [19].

5.1 The Negative Selection Algorithm (NSA)

The NSA is used to generate the network intrusion detectors. Forrest et al. (1994) proposed a computational model of self/nonself discrimination, which is called the NSA. This algorithm models the T cell maturation process that occurs in the thymus. Several variations of NSAs have been proposed after the original version was introduced[20].

There are two stages in NSAs as follows: “detector generation” and “abnormal detection.” In the first stage, a set of detectors is generated by completely randomized process that uses a collection of self as the input. Candidate detectors that match any of the self samples are eliminated, whereas unmatched ones are kept and added to detector set as shown in Figure 3.

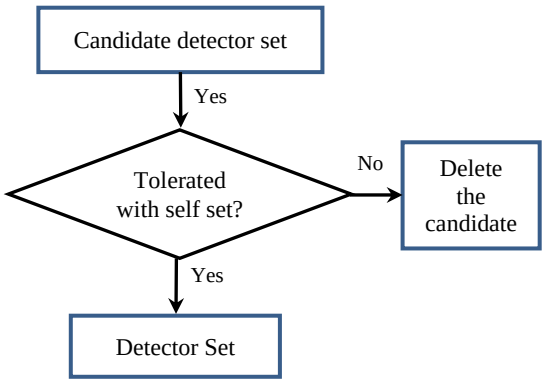

Figure 3: Detector Generation

In the detection stage, the stored detector set (generated in the first stage) are used to check whether new incoming samples correspond to self or nonself instances. If an input sample matches a detector, then it is identified as part of nonself, which in most applications, means that an anomaly/change has occurred as shown in Figure 4.

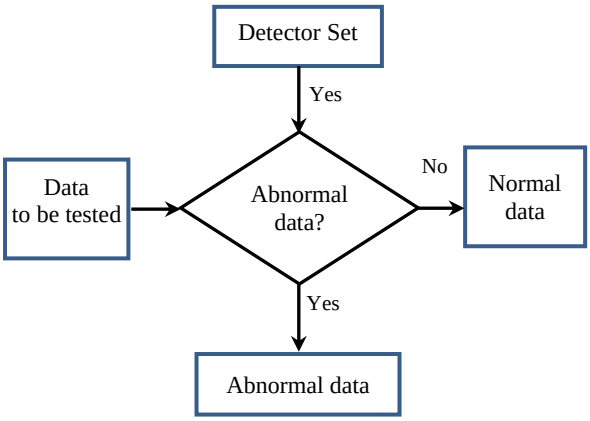

Figure 4: Abnormal Detection

6. THE PROPOSED HYBRID NIDS

The proposed hybrid network intrusion detection system use the open-source project Snort 2.9.7.5, which was issued in July 2015, as NIDS software. Snort running under Ubuntu 15.04 operating system. As shown in Figure 5, to achieve the proposed hybrid intrusion detection system, there are two main steps: detector generation and combination.

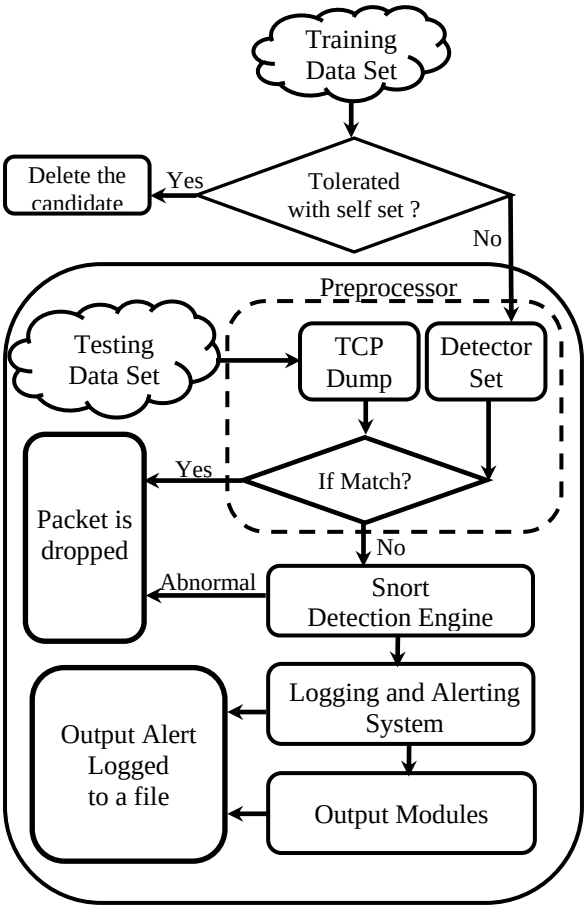

Fig. 5 Hybrid Intrusion Detection System

In the detector generation step, the KDD'99 training data is used to generate the intrusion detection detector using the following NSA detector generator algorithm:

Algorithm: NSA detector generator
Input: L, r, Z ∈ N where $1 \le r \le L$ and $S \subset U$; L = string length, r = matching threshold, Z = detector repertoire size, S is self (normal), U is the unity (self and non-self)

Output: Set $D \subset U$ detectors generated using r-chunk matching rule
begin
  $D := \emptyset$
  while $|D| < Z$ **do**
    Generate randomly bit string $d \in U$
    if d does not match any string in S **then**
      $D := D \cup \{d\}$
end

The used matching scheme is called r-chunk matching. The rule of r-chunk matching is defined as follows:

given a string $x = x_1 x_2 \ldots x_L$ and a detector $d = (d_1 d_2 \ldots d_M)$, with $M \le L$ and $i \le L - M + 1$,

$$\mathbf{d\ matches\ x} \equiv \mathbf{x_j} = \mathbf{d_j\ for\ j = i, \ldots, i + M - 1}$$

Where i represents the position where the r-chunk starts. Preliminary experiments suggest that the r-chunk matching rule can improve the accuracy and performance of the NS algorithm[21].

In the combination step, after detector generation Snort's preprocessor architecture has been used to combine the following NSA abnormal detection algorithm 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.

Algorithm: NSA abnormal detection
Input: Detector repertoire, Testing data
begin
  For (Testing data$_i$ ∈ **Testing data)**
    Testing data $_i$ Class := Self
    For (Detector $_j$ ∈ Detector repertoire)
      If (Matches **Testing data$_i$** , **Detector $_j$**)
        Testing data $_i$ Class := Non Self
        Break
      End If
    End For
  End For
End

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

- Preprocessor's source code file "nsa.cpp" was copied to the directory where "snort.c" lies in.
- The header file "nsa.h" defining NSA.
- "SetupNSA" function required for initializing NSA must be called from initial preprocessors function.
- As a last step, the project was recompiled in order to obtain Snort with NSA preprocessor.

7. EVALUATION STRATEGY AND EXPERIMENTAL RESULTS

In this paper several tools were used, including both hardware and software, to meet the intended objectives. The evaluation was performed on a system using Intel(R) Core(TM) i5 CPU M520 @ 2.40GHz processor with 4GB RAM running Snort 2.9.7.5 under Ubuntu 15.04 operating system. To evaluate the effective of the proposed system the 1999 DARPA data set was used. The DARPA evaluation dataset was used for the purpose of training as well as testing the intrusion detectors.

In the DARPA IDS evaluation dataset, all the network traffic including the entire payload of each packet was recorded in tcpdump format and provided for evaluation. The recorded data was in the form of sniffed network traffic, audit data and file system snapshots and tried to identify the intrusions that had been carried out against a test network during the data collection period. A mix of real and simulated machines formed the test network. Background traffic was artificially generated by the real and simulated machines while the attacks were carried out against the real machines. DARPA 1999 dataset consists of weeks one, two and three of training data and weeks four and five of test data. In training data, the weeks one and three consist of normal traffic and week two consists of labeled attacks [22].

The confusion matrix was used to evaluate the performance of the IDS. A confusion matrix is a specific table layout that allows visualization of the performance of an IDS. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another). In the binary class IDS, the intrusion detection system is mainly discriminate between to classes, "Attack" class (malicious threats or abnormal data) and "Normal" class (normal data). Table 1 shows the confusion matrix.

Table 1 Confusion Matrix

		Predicted		**Total**
		Normal	Attacks	
Actual	Normal	TN	FP	TN+FP
	Attacks	FN	TP	FN+TP
Total		TN+FN	TP+FP	

True Positives (**TP**) : The number of attack classified as attack.

True Negatives (**TN**) : The number normal classified as normal.

False Positives (**FP**) : The number of normal classified as attack.

False Negatives (**FN**) : The number of attack classified as normal

The efficiency of an IDS can be measured by the number of false positives and false negatives it produces. From the confusion matrix there are two important metrics can be calculated, Recall (R) and Precision (P). In the Intrusion detection the Recall (R) or (True Positive Rate – TPR) is used and defined as the Proportion of correctly predicted attack cases to the actual size of the attack class and calculated using the following equation:

$$\boldsymbol{R = \frac{TP}{(TP + FN)}} \qquad (\mathbf{1})$$

The Precision (P) may be defined in the intrusion detection field as the proportion of attack cases that were correctly predicted relative to the predicted size of the attack class, as calculated using the following equation:

$$\boldsymbol{P = \frac{TP}{(TP + FP)}} \qquad (\mathbf{2})$$

Often, there is an inverse relationship between precision and recall, where it is possible to increase one at the cost of reducing the other. So, the F-scores, which is the weighted harmonic mean of precision and recall, is usually used.

$$\boldsymbol{F - score = \frac{2 * P * R}{(P + R)}} \quad (\mathbf{3})$$

The F-score scores the balance between precision and recall. The F-score is a measure of the accuracy of a test.

First, Snort is run only on the dataset. Second the hybrid system is run on the dataset. The results are compared to determine the relative performance of proposed technique. These evaluations measure probability of detection and probability of false alarm for each system under test.

7.1 Performance of Snort on DARPA 1999 Dataset

Snort is tested on DARPA 1999 dataset (fourth and fifth weeks including attack) and the detected attacks are listed day by day. Attacks detected on a daily bases are shown in Fig. 6 Snort has detected 33 attacks out of 201 attacks available in DARPA 1999 dataset.

7.2 Performance of the Hybrid System on DARPA 1999 Dataset

Attacks detected by Snort and by the proposed hybrid system (Snort + NSA) are shown in Fig. 7. It is obvious that integrating NSA into Snort as a preprocessor increased the number of detected attacks. This shows the contribution of newly added preprocessor NSA to Snort IDS. Number of attacks detected by Snort increases from 33 to 102 in Snort + NSA version of the IDS. The reason for this increase is NSA making anomaly detection on packet headers and detecting attacks that Snort is unable to detect using rule definition files. This shows the contribution of newly added preprocessor

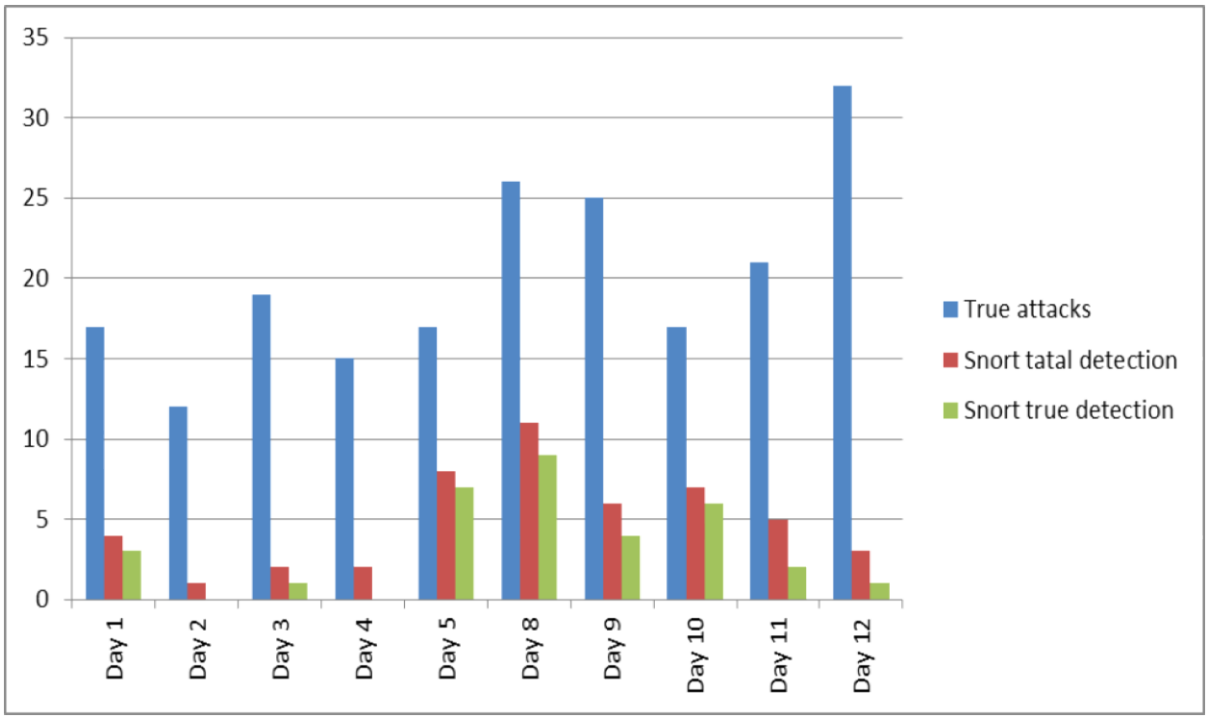

Fig. 6 Attacks detected by Snort on a daily bases.

Table 2 The important metrics of Snort on a daily bases.

	Day1	Day2	Day3	Day4	Day5	Day8	Day9	Day10	Day11	Day12
Recall	17.65%	0	5.26%	0	41.18%	34.62%	16%	35.29%	9.52%	3.12%
Precision	75%	0	50%	0	87.5%	81.81%	66.66%	85.71%	40%	33.33%
F-score	28.58%	0	9.51%	0	56%	48.65%	25.81%	50%	15.38%	5.71%

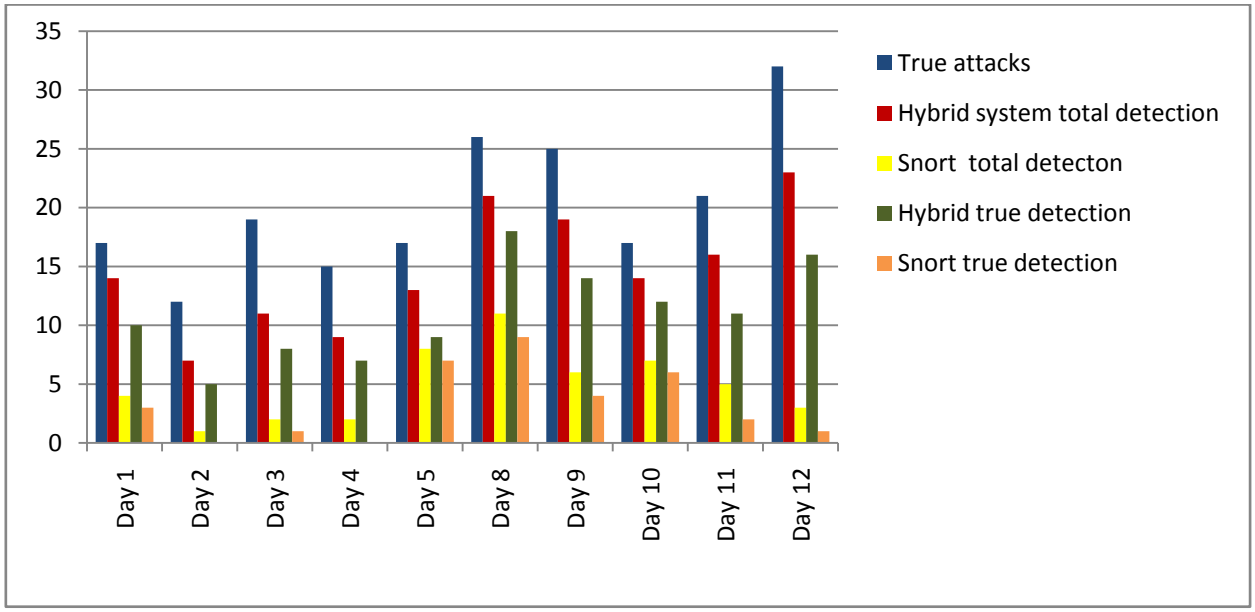

Fig. 7 Attacks detected by the hybrid system on a daily bases.

Table 3 The important metrics of the hybrid system on a daily bases

		Day1	Day2	Day3	Day4	Day5	Day8	Day9	Day10	Day11	Day12
Recall	Snort	17.65%	0	5.26%	0	41.18%	34.62%	16%	35.29%	9.52%	3.12%
	Hybrid System	58.82%	41.67%	42.11%	46.67%	52.94%	69.23%	56%	70.59%	52.38%	50%
Precision	Snort	75%	0	50%	0	87.5%	81.81%	66.66%	85.71%	40%	33.33%
	Hybrid System	71.43%	71.42%	72.73%	77.78%	69.23%	85.71%	73.68%	85.71%	68.75%	69.57%
F-score	Snort	28.58%	0	9.51%	0	56%	48.65%	25.81%	50%	15.38%	5.71%
	Hybrid System	64.51%	53.10%	53.33%	58.34%	60%	76.59%	63.63%	77.42%	59.46%	46.51%

8. CONCLUSION AND FUTURE WORK

Signature-based systems can only detect attacks that are known before whereas anomaly-based systems are able to detect unknown attacks. Anomaly-based IDSs can detect attacks whose signatures are unknown. NSA is added to signature-based IDS namely Snort as a preprocessor in this study. DARPA 1999 dataset which was created in MIT Lincoln Laboratories is used to evaluate the performance of new constructed hybrid IDS.

Firstly, Snort is tested on DARPA 1999 dataset and the number of attacks detected (33 attacks). Secondly, anomaly detection system, NSA has been merged to the preprocessor stage in the snort. The proposed hybrid system is tested on the same data and it detect (102 attacks), these results show that the hybrid IDS is more efficient than using snort on its own where it used the advantages of negative selection for detecting unknown attacks.

In the future work, the aim is to enhance the rate of false positive for the hybrid system.

9. REFERENCES

[1] Pandey, Aakanksha, and Nilay Khare. "String Matching Technique Based on Hardware: A Comparative Analysis." Advances in Computing and Information Technology. Springer Berlin Heidelberg, 2012. 339-347.

[2] Prabha, K., and S. Sukumaran. "Improved Single Keyword Pattern Matching Algorithm for Intrusion Detection System." International Journal of Computer Applications 90.9 (2014).

[3] Uddin, M., Rahman, A. A., Uddin, N., Memon, J., Alsaqour, R. A., & Kazi, S. (2013). Signature-based Multi-Layer Distributed Intrusion Detection System using Mobile Agents. IJ Network Security, 15(2), 97-105.

[4] Shen, J. and J. Wang. Network intrusion detection by artificial immune system. in IECON 2011-37th Annual Conference on IEEE Industrial Electronics Society. 2011. IEEE.

[5] Jinyin, C. and Y. Dongyong. A study of detector generation algorithms based on artificial immune in intrusion detection system. in Computer Research and Development (ICCRD), 2011 3rd International Conference on. 2011. IEEE.

[6] Aziz, A. S. A., Salama, M. A., Hassanien, A. E., & Hanafi, S. E. O. (2012, September). Detectors Generation using Genetic Algorithm for a Negative Selection Inspired Anomaly Network Intrusion Detection System. In FedCSIS (pp. 597-602).

[7] Zhou, Z., Zhongwen, C., Tiecheng, Z., & Xiaohui, G. (2010, May). The study on network intrusion detection system of Snort. In Networking and Digital Society (ICNDS), 2010 2nd International Conference on (Vol. 2, pp. 194-196). IEEE.

[8] Kumar, S. and R. Joshi. Design and implementation of IDS using Snort, Entropy and alert ranking system. in Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), 2011 International Conference on. 2011. IEEE.

[9] Peng, Y. and H. Wang. Design and implementation of network instruction detection system based on snort and NTOP. in 2012 International Conference on Systems and Informatics (ICSAI2012). 2012.

[10] Hussein, S.M., F.H.M. Ali, and Z. Kasiran. Evaluation effectiveness of hybrid IDs using snort with naive Bayes to detect attacks. in Digital Information and Communication Technology and it's Applications (DICTAP), 2012 Second International Conference on. 2012. IEEE.

[11] Pastrana, S., Tapiador, J. E., Orfila, A., & Peris-Lopez, P. (2015). DEFIDNET: A framework for optimal allocation of cyberdefenses in Intrusion Detection Networks. Computer Networks, 80, 66-88.

[12] Balzarotti, D., Testing network intrusion detection systems. 2006, Politecnico di Milano.

[13] Kabiri, P. and A.A. Ghorbani, Research on Intrusion Detection and Response: A Survey. IJ Network Security, 2005. 1(2): p. 84-102.

[14] Rehman, R.U., Intrusion detection systems with Snort: advanced IDS techniques using Snort, Apache, MySQL, PHP, and ACID. 2003: Prentice Hall Professional.

[15] Chakraborty, Nilotpal. "Intrusion Detection System and Intrusion Prevention System: A Comparative Study." International Journal of Computing and Business Research (IJCBR) ISSN (Online) (2013): 2229-6166.

[16] Johnson, L., Security Controls Evaluation, Testing, and Assessment Handbook. 2015: Elsevier Inc.

[17] Skarfone, K. and P. Mell, Guide to intrusion detection and prevention systems. 2007, National Institute of Standards and Technology, available at: csrc. nist. gov/publications/nistpubs/800-94/SP800-94. pdf.

[18] Powers, Simon T., and Jun He. "A hybrid artificial immune system and Self Organising Map for network intrusion detection." Information Sciences 178.15 (2008): 3024-3042.

[19] Ma, L. and Y. Chen. An improved Algorithm of Generating Network Intrusion Detector. in 2nd International Conference on Electronic & Mechanical Engineering and Information Technology. 2012.

[20] Forrest, S., Perelson, A. S., Allen, L., & Cherukuri, R. (1994, May). Self-nonself discrimination in a computer. In null (p. 202). IEEE.

[21] Meghanathan, N., D. Nagamalai, and N. Chaki, Advances in Computing and Information Technology: Proceedings of the Second International Conference on Advances in Computing and Information Technology (ACITY) July 13-15, 2012, Chennai, India - Volume 1. 2012: Springer.

[22] Thomas, C., V. Sharma, and N. Balakrishnan, Usefulness of DARPA dataset for intrusion detection system evaluation, in Data Mining, Intrusion Detection, Information Assurance and Data Networks Security. 2008.