Intrusion Detection Approach Based on DNA Signature

Sarab M. Hameed*, Omar Fitian Rashid
Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq.

Abstract

Intrusion-detection systems (IDSs) aim at detecting attacks against computer systems and networks or, in general, against information systems. Most of the diseases in human body are discovered through Deoxyribonucleic Acid (DNA) investigations. In this paper, the DNA sequence is utilized for intrusion detection by proposing an approach to detect attacks in network. The proposed approach is a misuse intrusion detection that consists of three stages. First, a DNA sequence for a network traffic taken from Knowledge Discovery and Data mining (KDD Cup 99) is generated. Then, Teiresias algorithm, which is used to detect sequences in human DNA and assist researchers in decoding the human genome, is used to discover the ***Shortest Tandem Repeat (STR)*** sequence and its ***position*** (i.e., ***pattern*** or ***keys***) in the network traffic. Finally, the Horspool algorithm is applied as a classification process to determine whether the network traffic is attack or normal. The performance of the proposed approach in terms of detection rate, accuracy, and false alarm rate are measured, showing the results are reasonable and accepted.

Keywords: DNA, Horspool Algorithm, Intrusion Detection, Shortest Tandem Repeate, Teiresias Algorithm.

كشف التطفل اعتمادا على توقيع الحمض النووي

سراب مجيد حميد*، عمر فتيان رشيد
قسم علوم الحاسبات، كليه العلوم، جامعه بغداد، بغداد، العراق.

الخلاصة

تهدف أنظمة الكشف عن التطفل في الكشف عن الهجمات ضد نظم الكمبيوتر والشبكات ، أو بشكل عام ضد نظم المعلومات. يتم اكتشاف معظم الأمراض في جسم الإنسان من خلال الحمض النووي (DNA) وفي هذه الورقة يتم استخدام تسلسل الحمض النووي لكشف التطفل من خلال اقتراح منهج للكشف عن الهجوم. والطريقة المقترحة في الكشف عن التطفل تتكون من ثلاث مراحل . أولا ، يتم إنشاء تسلسل الحمض النووي لحركة مرور الشبكة (KDD Cup 99) ثم استخدمت خوارزمية Teiresias للكشف عن تسلسل الحمض النووي في الإنسان و مساعدة الباحثين في استخدم الجينوم البشري لاكتشاف تسلسل أقصر تكرار (STR) و موقعها في حركة مرور الشبكة . وأخيرا ، يتم تطبيق خوارزمية Horspool كعملية التصنيف لتحديد ما إذا كانت حركة مرور الشبكة هجوم أوحاله طبيعيه . يتم قياس تقييم الأداء على نوعية الحل من حيث معدل اكتشاف والدقة ومعدل الانذار الكاذب، وتبين ان النتائج منطقيه ومقبولة.

1. Introduction

An intrusion detection system (IDS) is a device or software application that monitors network or system activities for malicious activities or policy violations and produces reports to a management

*Email: sarab_majeed@yahoo.com

station. The aim of IDS is to detect attacks against computer systems and networks. To understand the meaning of intrusion detection, one can use an analogy to the common "thief alarm". Just like the thief alarm, intrusion detection works on a computer system or network and is enabled to detect possible violations of security policies and raise an alarm to notify the proper authority [1]. Intrusion can be categorized as follows:

a. Denial of service (DoS): is one which degrades or disables a server, host, or network. The two most common approaches are to flood the target with data, or to send malformed data causing the target to crash due to a bug. Attacker tries to prevent legitimate users from using a service.

b. Probe: scan the networks to identify valid IP addresses and to collect information about them (e.g. what services they offer, operating system used).

c. Remote to Local (R2L) attacks where an attacker who has the ability to send packets to a machine over a network (but does not have an account on that machine), gains access to the machine

d. User to Root (U2R) attacks where an attacker who has an account on a computer system is able to misuse/elevate her or his privileges by exploiting vulnerability in computer mechanisms, a bug in the operating system or in a program that is installed on the system.

Furthermore, IDS can be categorized into different ways. The major categorizations are active and passive IDS, Network IDS (NIDS) and Host IDS (HIDS), and misuse detection and anomaly detection. In a passive system IDS, sensor detects something that looks like a gap in the system, logs the information and signals an alert on the console but does not take any preventive measures to stop the attack. On the other hand, an active IDS responds to the suspicious activity in the same way as the passive IDS with the additional ability to take action on the attack. NIDSs analyze network packets captured from a network segment, while HIDSs examine audit trails or system calls generated by individual hosts. In misuse detection identifies intrusions based on features of known attacks while anomaly detection analyzes the properties of normal behavior [2].

This paper investigates the using of Deoxyribonucleic Acid (DNA) bases for solving intrusion detection problem. The structure of the paper is organized as follows: section 2 presents the related work. Then, in section 3, the DNA concept is described. Section 4 clarifies the process of the proposed ID approach. Finally, the results of the proposed ID approach and the conclusions are demonstrated in section 4 and 5 respectively.

2. Related Work

Several researches works have been carried out in the field of IDS. The following presents some of these researches:

In [3], heuristic rules were proposed for two R2L attacks warezmaster and warezclient. An average of 53.08% detection rate is achieved for the two specific attacks in the R2L category.

In [4], a set of machine learning and pattern recognition algorithms against the four attack categories (probes, DoS, R2L, and U2R) found in the Knowledge Discovery and Data mining (KDD Cup 99) intrusion detection dataset are evaluated. They use nine algorithms to test the same dataset and then they compare the performance of all algorithms.

An anomaly detection approach applied in [5] to create a list of suspicious items has been used. Then a signature detection approach is used to classify these suspicious items into three categories, namely false alarms, attacks, and unknown attacks. The approach is based on an assumption that a high detection rate can be achieved by the anomaly detection component because missed intrusions in the first step will not be found by the follow-up signature detection component.

A novel hybrid IDS has been proposed in [6] which consisting of an anomaly detection module`, a misuse detection module and a decision support system A decision tree algorithm was used to classify various types of attacks. The final system was evaluated and experimental results showed that the proposed hybrid approach gave better performance over individual approaches.

In [2], a hybrid IDS combining both misuse detection and anomaly detection components is proposed, in which a random forests algorithm was applied firstly in the misuse detection module to detect known intrusions. Evaluations showed that their misuse detection module generated a high detection rate with a low false positive rate and at the same time the anomaly detection component has the potential to find novel intrusions.

A model based on hybrid fuzzy logic and neural network has been proposed in [7] for intrusion detection. This model has the ability to recognize an attack, to differentiate one attack from another and to detect new attacks with high detection rate and low false negative.

In [8], a normal signature sequence and alignment threshold value from processing the system training data and encode observed network connection into corresponding Deoxyribonucleic Acid (DNA) nucleotides sequence are generated, then to align the signature sequence with observed sequence to find similarity degree value and decide whether the connection is attack or normal.
In [9], an IDS using neural network based modeling is proposed. They use a combination Error-Back Propagation and Levenberg-Marquardt neural networks. The algorithm proved to be capable of capturing all intrusion attempts presented in the network communication while not generating any false alerts.

3. Deoxyribonucleic Acid

The foundation of DNA analysis requires an understanding of the basic components of DNA. DNA is the genetic material found in most organisms, including humans. The main role of DNA molecules is the long-term storage of information. The information in DNA is stored as a code made up of four chemical bases: adenine (A), thiamine (T), cytosine (C), and guanine (G). DNA bases pair up with each other, A with T and C with G, to form units called base pairs. Each base is also attached to a sugar molecule and a phosphate molecule. The order, or sequence, of these bases make individual DNA unique and determines the information available for building and maintaining an organism [10] Short Tandem Repeat describes a pattern that helps determine an individual's inherited traits. Tandem repeats occur in DNA when a pattern of two or more nucleotides is repeated and the repetitions are directly adjacent to each other [11].

4. The Proposed ID Approach

The proposed approach is misuse detection inspired from the human genome to detect the intrusion. The information source that feeds the proposed approach is KDD Cup99 attack dataset which is a public repository to promote the research works in the field of intrusion detection. Each record in the dataset has 41 features as shown in table 1. Twenty two features explain the connection while the remaining 19 features describe their connection properties to the same host with the last two seconds. The common criterion for categorizing computer intrusions is according to the attack type. Attacks fall into four main categories namely probes, DoS, R2L, and U2R [12]. KDD Cup99 dataset has only twenty two attack types and they are mostly of denial of service category.

Table 1- KDD Cup99 Features

Feature Name	Feature Type
protocol type, Service, Flag, Land, logged in, is hot login, is guest login	Discrete
duration, source byte destination bytes, wrong fragment, urgent, hot, failed logins, # compromised, root shell, su attempted, # root, # file creations, # shells, # access files, # outbound cmds, Count, srv count, serror rate,srv serror rate, rerror rate, srv rerror rate, same srv rate, diff srv rate, srv diff host rate, dst host count, dst host srv Count, dst host same srv rate, dst host diff srv rate, dst host same src port rate, dst host srv diff host rate, dst host serror Rate, dst host srv serror rate, dst host rerror rate, dst host srv rerror rate	Continuous

The proposed approach consists of three stages: Encoding technique, STR Sequence Extraction and matching process. The following subsections present the details of each stage.

4.1 Encoding Technique

The purpose of this technique is to convert the network traffic in the dataset to a DNA sequence. The challenge is to encode different network traffic attribute types and values into DNA nucleotides. The idea of the current DNA encoding is inspired from [8], however, with the following characteristics.
The attributes of the network traffics have all forms of the continuous and discrete values with significantly varying ranges. Each digit in a continuous attribute occupies two DNA nucleotides which is applicable to handle all possible values as shown in table 2

The discrete attributes are *protocol type*, *service and Flag*. *Protocol type* consists of 3 discrete values: **TCP, UDP**, and **ICM**. S*ervice* attribute consists of 71 discrete values. Finally, the *Flag* attribute has 11 discrete values: **SF, OTH, RSTO, S0, S1, S2, S3, REJ, RSTR, RSTOSO,** and **SH**.

In the proposed DNA encoded, each discrete attribute occupies the same number of nucleotides. Thus, three nucleotides are used for each value of flag and protocol discrete attributes. For service attributes two cases are carried out to handle the service values. The first case uses 4 nucleotides to handle all 72 values and the second one uses 3 nucleotides for the first 64 values and the rest 8 values is defined by using 4 nucleotides. Furthermore, different block sizes in front of each discrete attribute are inserted as a header to distinguish it from continuous attribute and to give more possible values. Many attempts are carried out to find a suitable block size. These are as follows:

A. One nucleotide is added before discrete attributes such as for flag attribute: ACCC, for protocol type: TCCC, and for service: CCCC.
B. Two nucleotides are added before discrete attributes such as for flag: AACCC, for protocol type: TTCCC, and for service: CCCCC.
C. Three nucleotides are added before discrete attributes such as for flag: AAACCC, for protocol: TTTCCC, and for service: CCCCCC.
D. Four nucleotides are added before discrete attributes such as for flag: AAAACCC, for protocol type: TTTTCCC, and for service: CCCCCCC.

After that it is found experimentally that the block size with three nucleotides provides the best results. Hence, flag, protocol type and service attributes are encoded according to tables (3, 4, 5) respectively. Additionally, the discrete attributes including *Land, logged in, is hot login,* and is guest login which take either 0 or 1 are treated as continuous attributes.

Table 2- Nucleotide Sequence for Digits and '.'

Digits	Nucleotides	Digits	Nucleotides
0	AA	5	CT
1	AC	6	CC
2	AG	7	GC
3	AT	8	GT
4	CG	9	TT
.	GA		

Table 3- Flag-Nucleotide Sequence

Flag	Nucleotides	Flag	Nucleotides
SF	AAAACT	SI	AAATCG
OTH	AAAAGC	S2	AAATCT
RSTO	AAACAG	S3	AAAGAG
SO	AAAGAC	REJ	AAACAA
RSTR	AAAATC	RSTOSO	AAAGTC
SH	AAAATG		

Table 4- Protocols-Nucleotide Sequence

Protocol	Nucleotides
TCP	TTTTCC
UDP	TTTATT
ICMP	TTTGTA

Table 5- Services-Nucleotide Sequence

Service	Nucleotides	Service	Nucleotides
HTTP	CCCAAA	FINGER	CCCCTA
DOMAIN U	CCCAGA	FTP DATA	CCCCGG
ECR I	CCCGGA	POP 3	CCCTTC
SMTP	CCCGGG	AUTH	CCCTTT
ECO I	CCCTGA	OTHER	CCCTAT
TELNET	CCCGAA	PRIVATE	CCCCGT
NTP U	CCCACT	VMNET	CCCAGG
URP I	CCCTAA	BGP	CCCGTA
Z39 50	CCCTGC	DOMAIN	CCCATG
FTP	CCCAGT	GOPHER	CCCCTC
SSH	CCCATC	REMOTE JOB	CCCCAA
WHOIS	CCCCCT	RJE	CCCCTG
CTF	CCCCTT	HOSTNAMES	CCCTTA
LINK	CCCGGT	CSNET NS	CCCGTG
SUPDUP	CCCGAC	POP 2	CCCCCA
ISO TSAP	CCCAAC	SUNRPC	CCCATT
UUCP PATH	CCCTAC	LDAP	CCCCGA
NNTP	CCCTCC	HTTP 443	CCCGTT
IMAP4	CCCAAT	EXEC	CCCCAC
SQL NET	CCCTTG	NETBIOS DGM	CCCACG
LOGIN	CCCTCT	COURIER	CCCGTC
SHELL	CCCACA	NETBIOS SSN	CCCGGC
PRINTER	CCCCAG	KSHELL	CCCACC
EFS	CCCCAT	DISCARD	CCCATA
DAYTIME	CCCTGT	NAME	CCCAAG
SYSTAT	CCCGAG	KLOGIN	CCCAGC
NETSTAT	CCCCGC	IRC	CCCCCG
TIME	CCCTCA	Xll	CCCTCG
ECO	CCCTGG	NNSP	CCCGCC
ICMP	CCCGCG	PM DUMP	CCCGCT
MTP	CCCGCA	TFTP U	CCCGAT
NETBIOS NS	CCCTAG	TIM I	CCCAAAT
UUCP	CCCAGAA	HARVEST	CCCTCGG
AOL	CCCGGCC	HTTP 2784	CCCACGG
HTTP 8001	CCCTTAA	URH I	CCCTTGG
RED I	CCCTTCC		

4.2 Teiresias Algorithm STR Sequence Extraction

Teiresias algorithm [13] is an interested algorithm for finding patterns in human DNA and in this paper; it is used to discover the STR sequence for each network traffic attack. Various attempts are followed to find the best keys (i.e. STR sequence). The tried keys have lengths with various number of DNA nucleotides, starting from four to sixteen nucleotides. The symbols for these keys are given such as for four nucleotides length is represented by AACG, for five nucleotides length is represented by AACGT and so on. It is found that the key with length of five letters is the most appropriate key that gives the best result of verification. However, in addition to the five characters length as the best key, there is another series of various keys of five characters that produce good results. Algorithm 1 outlines how the STR sequence is extracted from intrusion dataset network traffic using Teiresias algorithm. This algorithm is applied for each attack type to determine the key and the position for each attack and the extracted STR sequences are stored.

Algorithm 1: STR Sequence Extraction

Input : Number of Samples (*N*), Key Length (K_l), Network Traffic Samples $(NT_{ij} : 1 \le i \le N, \quad 1 \le j \le 41)$

Output: Set of STR Sequence for Attack Type (SK)

1: **/* Convert the network traffic samples to DNA bases depending on DNA encoding technique */**

2: **/*Split the network traffic samples into two groups:**
one contains Attack samples $(AS_{ij} : 1 \le i \le AS_l, \quad 1 \le j \le L_i)$
and the other contains Normal samples $(NS_{ij} : 1 \le i \le NS_l, \quad 1 \le j \le L_i)$ ***/**

3: **/*Divide each sample of attack and normal to block length equal to** K_l ***/**

$A \leftarrow 0$ **/* A is number of attack block */**

For all $i \in \{1,..., AS_l\}$ do

For all $j \in \{1,..., L_l\}$ do

Set $AB(A)$ to substring of $AS(i, j)$ equal to K_l.

Increment A

End for

End for

$N \leftarrow 0$ **/* N is number of attack block */**

For all $i \in \{1,..., NS_l\}$ do

For all $j \in \{1,..., L_l\}$ do

Set $NB(N)$ to substring of $NS(i, j)$ equal to K_l.

Increment N

End for

End for

4: **/* Select the attack blocks which are not matching with the normal blocks */**

$R \leftarrow 0$ **/* R is number of selected attack block */**

For all $i \in \{1,..., A\}$ do

$IsMatch \leftarrow 0$

For all $j \in \{1,..., N\}$ do

If $AB(i) = NB(j)$

$IsMatch \leftarrow 1$

End for

If $IsMatch = 0$

$SB(R) \leftarrow AB(i)$ **/* SB is a vector of selected attack block */**

Increment R

End if

End for

5: **/*Calculate the repetition of the selected attack blocks */**

For all $i \in \{1,..., R\}$ do

$Sum \leftarrow 0$

For all $j \in \{1,..., R\}$ do

If $SB(i) = SB(j)$

Increment Sum

End for

RB(i)= Sum **/* RB is a vector of repetition for each attack block */**

End for

6: **/*Calculate the Maximum repetition of the selected attack blocks */**

$Max \leftarrow RB(1)$

For all $i \in \{2,...,R\}$ do
  If $Max < RB(i)$
    $Max \leftarrow RB(i)$
End for
$Count \leftarrow 0$
$No \leftarrow 0$
For all $i \in \{1,...,R\}$ do
  While (Count < STR_n) /* STR_n is number of STR sequence */
    If $RB(i) = Max - No$ then
      SK (Count) = Rb(i) ***/ SK is a Set of STR Sequence /***
      Count= Count +1
    End if
  End while
  Increment No
End for

After the use of Teiresias algorithm another procedure is applied to find common key between different attacks to reduce key numbers (less match need), then the best STR sequence is found; the position of this key in network traffic is found. Algorithm 2 outlines the steps of finding the best keys.

Algorithm 2: Best Keys Selection

Input: Set of STR Sequence (SK), Number of attack (AN), Number of STR sequence (Count)
Output: Best STR sequence (BS)

1:

$m \leftarrow 0$
For all $i \in \{1,...,AN\}$ do
  $IsMatch \leftarrow 0$
  $j \leftarrow 1$
  While $j <= Count \; and \; IsMatch = 0$
    For all $k \in \{1,...,AN\}$ do
      For all $l \in \{1,...,Count\}$ do
        If $SK(i,j) \neq "" \; and \; SK(i,j) = SK(k,l) \; and \; i \neq k \; and \; j \neq l$ then
          $BS(m) \leftarrow SK(i,j)$
          Increment m
          For all $s \in \{1,...,Count\}$ do
            $SK(i,s) \leftarrow ""$
            $SK(i,s) \leftarrow ""$
          End for
          $IsMatch \leftarrow 1$
        End if
      End for
    End for
    Increment j
  End while
End for
For all $i \in \{1,...,AN\}$ do
  $IsMatch \leftarrow 0$
  For all $j \in \{1,...,Count\}$ do
    If $SK(i,j) \neq "" \; and \; IsMatch = 0$ then
      $BS(m) \leftarrow SK(i,j)$
      Increment m
      $IsMatch \leftarrow 1$
    End if
  End for
End for

4.3 Matching Process

After STR sequences of the network traffic is found by Teiresias algorithm, the differentiation between normal and attack is done by Horspool algorithm using two different methods. Either the STR sequences or STR sequences and their positions are used to differentiate between attacks from normal of network traffic.

To Apply Horspool algorithm, a pre-computed table called bad character table is created that indicates how much to shift based on network traffic DNA nucleotide causing a mismatch of the STR sequence. Bad character table is created depending on the DNA bases (A, C, G, T) that found in keys. One nucleotide at a time is checked. If the nucleotide is mismatch, then check if this nucleotide is found in the key then shift key by the right most distance of that nucleotide and if nucleotide is not found then shift key position by distance equal to key length. Algorithm 2 clarifies how to build bad character table that used to indicates how much to shift in Horspool algorithm,

To apply Horspool search algorithm a pre computed table called *bad character table* is created to indicate how much to shift based on network traffic DNA nucleotide causing a mismatch of the STR sequence. The steps of creation bad character table are outlined in Algorithm 3.

Algorithm 3: Bad Character Table Creation	
Input : STR sequence(BS) , Key Length (K_l), Network Traffic Samples $(NT_{ij} : 1 \le i \le N, \quad 1 \le j \le 41)$, Network length (NT_l) **Output:** Bad character table *(Bct)*	

	For all $i \in \{1,...,NT\}$ do
	For all $j \in \{1,...,K_l\}$ do
	Set p to substring of NT equal to K_l .
	Set c to substring of $BS(j)$ equal to K_l .
	If $c \neq p$ then
	$Bct(i) \leftarrow K_l$ if c is not first m-1 characters of the p.
	$Bct(i) \leftarrow$ Distance of the right most c in the first $K_l - 1$ characters of the p.
	End if
	End for
	End for

Then, STR sequence for an attack (key) is slid across the network traffic (NT) from left to right, but the real symbol comparisons between them are passed from the STR sequence's right to left. Algorithm 4 clarifies the steps of Horspool to check whether the network traffic sample is normal or attack.

Algorithm 4: Horspool for ID	
Input: STR sequence(BS), Key Length (K_l), Network Traffic Samples $(NT_{ij} : 1 \le i \le N, \quad 1 \le j \le 41)$, Network length (NT_l) ,Bad character table (Bct) **Output:** Decision either Normal or Attack	
1: **2:**	**/* Convert the network traffic samples to DNA bases depending on DNA encoding method */** **/* Repeat until a matching** *STR sequence* **is found or** *network traffic* **ends*/** $IsFound \leftarrow 0$ While $i \le NT_l$ *and* $IsFound = 0$ Set $Block$ to substring of NT equal to K_l . $match \leftarrow 0$

3:	**/*STR sequence checking: If first character of "*key*" is matching first character of *NT* then shift the position of pointer otherwise shift the key*/**
	For all $j \in \{K_l,...,1\}$ do
	If $Substring(Block, j) = Substring(BS, j)$ then
	Increment c
	If $c = K_l$ then
	$IsFound \leftarrow 1$
	End if
	End for
	$i \leftarrow i + Bct(K_l - c)$
	End while

4:	/* **Decided the network traffic samples is attack or normal** */ If $IsFound = 0$ then the network traffic is normal else the network traffic is attack

5. Performance Assessment

This section evaluates the performance of the proposed approach for solving ID problem. The network traffic data from KDD Cup 99 dataset was used to evaluate the proposed approach. The evaluation is presented in terms of detection rate (DR), false alarm rate (FAR) and accuracy. The formula for calculating these measures are given as in equations (1, 2 and 3) respectively [14].

$$DR = \frac{TP}{TP+FN} \quad (1)$$

$$FAR = \frac{FP}{TN + FP} \quad (2)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \quad (3)$$

Where
TP occurs when IDS correctly classified as intrusion
FP occurs when a legitimate action misclassified as intrusion
TN is produced whenever a normal action is correctly classified as a legitimate action
and
FN occurs when an attack is not detected by IDS.
To conduct the experiment, the dataset is divided into two parts training dataset and testing dataset. The training dataset is used to find the STR sequence of the network traffic (i.e., generate DNA signature for normal and attack network traffics), however testing dataset is used to evaluate the performance of the proposed approach. In this paper, 180 samples are selected randomly as training dataset and 4000 samples randomly selected as testing dataset. The evaluation results of the proposed approach are presented as shown in table 6- when STR sequence or STR sequence and their positions are used. Table 7- demonstrates the DR result for each type of attacks in details.

Table 6- DR, FAR and Accuracy Results of the Proposed Approach

Measure	**STR Sequence without Positions**	**STR Sequence with its Positions**
DR	99%	97.57 %
FAR	27.2%	1%
Accuracy	94.4%	97.82%

Table 7- DR for each attack Type

Class	STR Sequence without Positions	STR Sequence with its Position
DoS	99.29 %	97.79 %
Probe	98.39%	97.65%
R2L	99.01%	96.73%
U2R	96.15%	92.30%

6. Conclusions

This paper is concerned with the problem of intrusion detection and how the DNA bases are utilized. DNA sequence generation is established with different network parameter values, which is used to convert the network traffic to DNA bases. The Teiresias algorithm discovers the STR sequence for each network traffic type. Then Horspool algorithm is used as a classification process between normal and attack network traffic. It has been shown that the extracted STR sequence (key) for each attack type using Teiresias algorithm is able to detect computer network attacks and the matching process using Horspool algorithm with STR sequences only or STR sequences and their position has an important effect on performance of the proposed approach.

References

1. Axelsson, S., **1999** "Research in Intrusion-Detection Systems A Survey", the Swedish National Board for Industrial and Technical Development. August 19,
2. Zhang, J. and Zulkernine, M., **2006** "A Hybrid Network Intrusion Detection Technique Using Random Forests" Proceedings of the First International Conference on Availability, Reliability and Security, Vienna, University of Technology, Austria, pp: 121–132, April.
3. Sabhnani, M. and Gursel, S., **2003** "KDD Feature Set Complaint Heuristic Rules for R2L Attack Detection", Security and Management, pp: 310-316.
4. Sabhnani, M. and Serpen, G., **2003** "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context, In Proceedings of the International Conference on Machine Learning, Models, Technologies and Applications", Vol. 1, pp: 209-215.
5. Tombini, E., Debar, H., Me, L., Ducasse, M., Telecom, F. and Caen, F., **2004** "A Serial Combination of Anomaly and Misuse IDSes Applied to HTTP Traffic", ACSAC '04 Proceedings of the 20th Annual Computer Security Applications Conference, pp: 428-437,.
6. Depren, O., Topallar, M., Anarim, E. and Ciliz, M., **2005** "An Intelligent Intrusion Detection System for Anomaly and Misuse Detection in Computer Networks", Expert systems with Applications 29(4), pp: 713–722.
7. Jawhar, M. and Mehrotra, M., **2010** "Design Network Intrusion Detection System using Hybrid Fuzzy-Neural Network", International Journal of Computer Science and Security, Vol. 4, Issue 3,.
8. Ibaisi, T., Abu-Dalhoum, A., Al-Rawi, M., Alfonseca, M. and Ortega, A., **2008** "Network Intrusion Detection Using Genetic Algorithm to find Best DNA Signature", *WSEAS Journal*.
9. Linda, O., Vollmer, T., Manic, M. **2009** "Neural Network Based Intrusion Detection System for Critical Infrastructures" Neural Networks, IJCNN International Joint Conference, pp:1827-1834.
10. Soram, R. and Khomdram, M., **2010** "Biometric DNA and ECDLP Based Personal Authentication System: A Superior Posse of Security", IJCSNS *International Journal of Computer Science and Network Security*, 10(1).
11. Oki, E., Oda, S., Maehara, Y. and Sugimachi, K., **1999** "Mutated Gene Specific Phenotypes of Dinucleotide Repeat Instability in Human Colorectal Carcinoma Cell Lines Deficient in DNA Mismatch Repair", *Oncoqene Journal, Mar*.
12. Mahoney, M. and Chan, P. **2002** "Learning Models of Network Traffic for Detecting Novel Attacks", supported by DARPA (F30602-00-1-0603).
13. Rigoutsos, I. and Floratos, A., **1998** "Combinatorial Pattern Discovery in Biological Sequences: *The Teiresias Algorithm, Bioinformatics*", 14(11998), pp: 55-67.
14. Wu, S. and Benzhaf, W., **2010** "The Use of Computation Intelligence in Intrusion Detection Systems", *Applied Soft Computing*, 10(1), pp: 1–35, January.