Automated Flow-based Rule Generation for Network Intrusion Detection Systems

Naser Fallahi
Department of Computer Science and Engineering and IT
Shiraz University
Shiraz, Iran, 7134851154
Email: Nfallahi@cse.shirazu.ac.ir

Ashkan Sami
Department of Computer Science and Engineering and IT
Shiraz University
Shiraz, Iran, 7134851154
Email: Sami@cse.shirazu.ac.ir

MortezaTajbakhsh
Department of Computer Science and Engineering and IT
Shiraz University
Shiraz, Iran, 7134851154
Email: Tajbakhsh@cse.shirazu.ac.ir

Abstract—**Snort is a popular open-source Intrusion Detection System (IDS). Since rules are updated offline and network environment changes dynamically, Snort has a low detection rate especially for new types of attacks. Since attack signatures are not stored in the system, attackers could intrude without being detected. The aim of this research is to automate rule generation for system by use of logs of performed attacks. This approach has been implemented using two data mining algorithms called Ripper and C5.0. Automatic rule generation improves security of Snort and attacks are detected better. Five types of attacks, like Denial of service and Brute Force, have been investigated in this work and tested on newly released ISCX 2012 dataset which has 84.42 Gigabytes. By processing the dump, it can be used to generate general rules and eight new features from known features of streams. Detection rate of more than 99 percent was obtained for some attacks, which represent sensible impact of this approach on Snort software.**

Keywords-network intrusion detection systems; signature-based detection; automatic rule generation; computer security; data mining algorithms.

I. INTRODUCTION

Intrusion detection system is a device or software application that monitors network or system for malicious activities or policy violations and reports them to a management station. IDS comes in a variety of flavors and approacheswith the goal of detecting suspicious traffic in different ways.Two main categories are: network based (NIDS) and host based (HIDS) intrusion detection systems. HIDS is installed only on one computer.NIDS usually scan input and output traffic from network access point, and generally has several sensors in different points to receive network traffic. Received features from this traffic are sent to a central analysis database in order to detect intrusion activities based on different intrusion detection methods.

Based on their detection methods,IDSs are also categorized into two main types: anomaly-based and signature-based detection methods [1]. Anomaly-based IDSs monitor network traffic and compare it against established baselines. The baselines will identify what is “normal” for that network which may include: the sort of bandwidth or protocols that are generally used or the ports and devices generally connected to each other. When the traffic is anomalous or significantly different from the baseline, IDS alerts the administrator or users. If the baselines are not intelligently configured, an alarm for a legitimate use of bandwidth is issued, False Positive, which is the main burden of anomaly-based methods [2]. In contrast, signature-based IDSs do not issue many false alarms. Since in signature-based IDS packets are monitored on the network and are compared against a database of signatures or attributes from known malicious threats (similar to the way most antivirus softwares detect malwares). The issue there is the lag between discovering a new threat in the wild and applying the signature for detecting that threat to the IDS. In other words, signature-based IDS would suffer from higher rate of False-Negatives or are unable to detect the new threats [3].

This research tries to improve signature-based IDSs by automating signature generation. A mitigation of the shortcoming of these two methods is to speed up the process of signature generation for signature-based IDSs. Intrusion detection systems, which their architecture is network based, perform in network platform, and scan the network traffic and analyze it in all different network layers to detect indications of intrusion activities or attacks [4]. The focus of this work is on the architecture of network based IDS.

Traditional Intrusion detection systems inspect each packet payload for detecting known attacks from normal behaviors [5, 6]. One of the problems of packet inspection, in addition to difficulty, is that it makes the operation almost impossible at several gigabytes per second speeds [7]. Therefore, it is necessary to investigate some alternatives for packet inspection in high-speed lines. One option which recently has drawn interest of researchers and operators is intrusion detection based on data flow, which is the focus of this article.

ISCX benchmark dataset [8] containins normal traffic and also five types of attack namely: denial of services, distributed denial of services, brute force, brute force SSH [9] and infiltrating the network from the inside [8], which are generated separately and in different days. After merging the normal traffic and all five types of attacks and running two data mining algorithms on it, the rules are obtained for each attack that some of them have more than 99 percent detection rate. Also for each attack of distributed denial of services, brute

978-1-4673-8789-7/16/$31.00 ©2016 IEEE

force and brute force SSH, a rule is obtained with F-measure more than 0.9.

Basically, this research is to enhance signature-based intrusion detection systems in computer networks by automating rule generating and updating IDSs with the newly generated rules. This intrusion detection is based on data flow. This approach is implemented using Ripper [10] and C5.0 [11] algorithms. The followings are the major contributions of this paper: 1-Speed of attack detection is increased by Flow-based approach in comparison with packet-based approach.Use flow-based detection approach that is a new, effective and up-to-date has been investigated. 2-Selected features for detecting attacks do not belong to a particular network, andconsequently they express general rules.3-Eight new features are generated for flow-based intrusion detection.The results show that these eight features have a significant impact in detection capability. 4-This is the first work in the field of automated rule generation, which has been carried out by considering all attacks, all protocols and data to be integrated on ISCX 2012 dataset.

The rest of the paper is organized as follows: second section reviews related works. In the third section, the proposed architecture and features are explained. The fourth section is about the evaluation of the proposed algorithm on dataset. In the fifth section, after research conclusion, future works are discussed. There are references in the last section.

II. RELATED WORKS

Intrusion detection in computer networks is one of the important researching topics in recent years. Bhuyan et al. [12] had discussed about methods, systems, tools and datasets provided, in details. More specifically, Sperotto et al. [13] have investigated advantages of flow-based intrusion detection and related works in this field. After introduction of Snort in 1998, automatic intrusion detection was investied more seriously and related research in this area have provided different approaches, Catania and garino[14] in addition to exploring the solutions, have also proposed future challenges.

A. Researches Related to Automatic Rule Generation

Gang et al. [15] proposed a method based on genetic algorithm for intrusion detection in network and then implemented a software to perform that method. Bankovic et al. [16] proposed a signature-based detection system based on genetic algorithm approach. Research evaluation was applied just on a part of dataset which contained three attacks named “portsweep”, “smurf” and “neptune” and results showed more than 90 percent detection rate. Vollmer et al. [17] proposed a genetic algorithm for automatic rule generation. In this method, 11 features of ICMP network packet were chosen and genetic algorithm was performed on them. Experimental results on 10 tests showed 100 percent rule alarm rate. Gomez et al. [18] proposed a Pareto-based multi objective evolutionary algorithm for automatic rule generation in a rule-based intrusion detection system.

B. Related Research on ISCX 2012 Dataset

Catania and Garino [19] proposed MIDAS (autonoMous Intrusion Detection Assistance System), a prototype NIDS based on ML techniques. Kumar and Kumar [20] proposed a novel evolutionary approach for effective intrusion detection based on benchmark datasets. Milliken et al. [21] proposed an effect analysis of pairs of attributes in order to improve intrusion detection using an ensemble-based classification approach.Tan et al. [22] proposed DoS attack detection system, which is developed based on a widely used dissimilarity measure, namely Earth Mover’s Distance (EMD). The results presented in the system evaluation section illustrate that their detection system can detect unknown DoS attacks and achieves 99.95% detection accuracy on KDD Cup 99 dataset and 90.12% detection accuracy on ISCX 2012 IDS evaluation dataset with the capability of processing of 59,000 traffic records per second, approximately.Costa et al. [23] introduced Optimum-Path ForestClustering method for detecting anomaly or malicious behaviors in network. Obtained detection rate of this algorithm on ISCX dataset is 96.37.

III. PROPOSED ARCHITECTURE AND ITS FEATURES

In this section, the proposed architectureforthe work is described. This architectureconsistsoftwo phases:trainingand testing, which isshownin Figure 1.

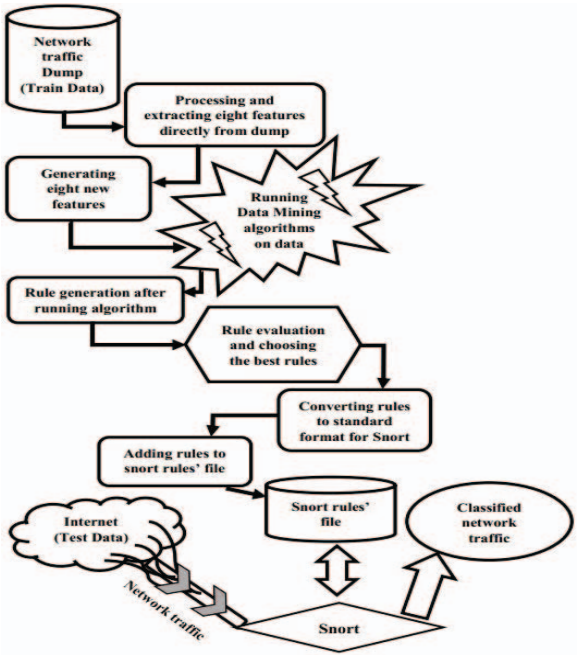

Figure 1. Algorithm architecture of automatic rule generation

Inthe first part ofthe training phase, there is the network traffic(train data). This data is in the form of raw and dump file that separately have a special attack each day. Files for network traffic have tags ofnormal or attack but type of attack is specified according to its file name. Tosimulate thereal environmentandthe dynamicof the network, we merged all thefive daysof attacksinto one file.Then we changed the labelso thatwe can detect attack trafficas well asits type. To implement thealgorithmwe followed twosteps. In the first step,intendedfeaturesare extracted from raw dataandare storedinaMicrosoft Excel file. In the pre-processing stage, some features were deleted and a series of new features are

introduced.Then, Ripper and C5.0algorithms are performed with 10-fold Cross Validation,and rules extractedfromdecision treeare the algorithm output. After evaluating thegenerated rulesandchoosing the best of them, we convert them to the standard format to use in Snort software. In the next step, these standardized rules are added the Snort rule file.In thetestphase, the testingnetwork trafficenters Snort, and normal and anomaly traffic classification performs, so that normaltraffic passes from Snort and is prevented from attack traffic. Weka3.7.12 implementation of RIPPER and C5.0 algorithm using Clementine 12.0 were used.

A. Data Flow and its Features

Currentcontent-basedintrusion detectionsystemscould reach theprocessing rate of100 to 200Mbps [7]. Therefore,researchers andoperators began to investigatealternative approaches, such as flow-based intrusion detection.In this approach, instead of the contents ofeach packetindividually,thedata over the network is analyzed [13]. In other words,flow-based intrusion detection handles lessamountof data compared withtraditionalintrusion detection systems. Sperotto et al. [13] believe that thedetectionbased on theflowcanbe consideredas a supplementto packet inspection, and should notbe consideredas an alternative. According to the definitionprovided byIPFIX fortheIP flow [24] a flow defined as a set ofIPpacketspassinganobservationpointin the networkduring aspecified time interval. Allpackagesthatbelongtoaparticular flowhavea set ofcommonfeatures. In terms ofIPFIX,commonproperties are called thekeysof flow: for example, these are source and destinationaddress, source and destinationport numbers, and IP protocol.

B. Extracted Features

In this section weintroduce andinvestigate theextracted features that are essential for classification of normalandanomaloustraffic. In Table 1, formula of new features are proposed, and in Table 2, names of 17 final features that are applied for model construction and new rules generation with their concepts are shown. Nine features are extracted directly from raw data, which are defined based on basic features for data flow in section 3. These nine features include destination port, protocol, sending flow direction, flow duration time in seconds, the totalvolume andtotal number ofpacketssent and receivedin adata flow and data flow tag. Since our purpose is to generate expressive rules, so the number of features could be ignored and it is not used for generating rules. Source and destination addresses are features that could be ignored, because these addresses in each network could be different. Due to some attacks that perform on a specific destination port, having that port in generated rule the attack could be detected. Thus, we ignore feature of source port and consider the destination port address to construct model and generate the rules. Regardingthe totalnumber and volume ofpacketssent andreceived ina flow, the average volume ofthese packets isalsoeffectivein detectingattacks.To obtainthis feature, we divided the total volume by the total number of packets sentto have the averagevolume of thepacketshavebeen sent.Similarly,the average volume of thereceived packets is also present.To obtain the number of the input to the output packets rate, the total number of packets received divides the total number of packets sent. Similarly, we can obtain the rate of input to the output packets volume.

TABLE I. NEW GENERATED FEATURES AND CALCULATION FORMULAS

Features Name	Calculation formulas
Avg_Src_Packet_Size	totalSourceBytes / totalSourcePackets
Avg_Des_Packet_Size	totalDestinationBytes/totalDestinationPackets
IOPR	totalSourcePackets/totalDestinationPackets
IOBR	totalSourceBytes/totalDestinationBytes
Avg_Src_Packet_per_sec	totalSourcePackets/ Duration (Second)
Avg_Des_Packet_per_sec	totalDestinationPackets/ Duration (Second)
Avg_Src_Byte_per_sec	totalSourceBytes/ Duration (Second)
Avg_Des_Byte_per_sec	totalDestinationBytes/ Duration (Second)

TABLE II. EXTRACTED FEATURES AND THEIR CONCEPTS

Features Name	Definitions based on data flow
totalSourceBytes	Total Source Bytes
totalDestinationBytes	Total Destination Bytes
totalSourcePackets	Total Source Packets
totalDestinationPackets	Total Destination Packets
*Avg_Src_Packet_Size	Average Source Packet Size
*Avg_Des_Packet_Size	Average Destination Packet Size
*IOPR	Input-Output packet rate
*IOBR	Input-Output byte rate
*Avg_Src_Packet_per_sec	Average source packets per second
*Avg_Des_Packet_per_sec	Average destination packets per second
*Avg_Src_Byte_per_sec	Average source bytes per second
*Avg_Des_Byte_per_sec	Average destination bytes per second
Direction	Direction of packets
protocolName	Protocol name
destinationPort	Destination port
Duration(Sec)	Duration based on second
Tag	Flow tag (type of attack or normal)

Rate of sending and receiving total volume and number of packets to the time, is another effective features on attacks detection. For this purpose, the total number of sent packets is divided by time interval. Similarly, rate of received packets to time is calculated. In addition, to get the rate of sent and received packets, total volume of sent and received packets is divided by the time. These are eight new features extracted from main features of data flow. Second column of table 2

states the concept of each feature in the form of a data flow. Eightnew featureswith an asteriskat the beginning are distinguishedfromtheother features.

According to represented explanations, we have totally 17 features, nine main features of them are extracted directly from raw data and eight other new features obtained from these features. The last feature also determines normal or attack tag of a data flow. Dueto the fact thatin this studywe havefiveattacks (multi label classification were used) which are merged in one file, so to distinguish between attacks, data flow tag related to attacks is converted from "Attack" to "Attack type" so that in addition to attack detection, the attack type is also detected. For example, we convert tag of denial of service from "Attack" to "Dos Attack".

C. Introduction of ISCX 2012 Dataset

In order to evaluate proposed methods and algorithms, ISCX 2012 dataset [8] has been used. The advantage of using such a dataset is the inclusion of a labeled flow file that supports the use of supervised data mining algorithms. This dataset consists of a week of traffic data, which is divided into seven different files. Each file is in packet capture format, covering one day having onetype of attack. The captures contain 84 gigabytes of traffic generated over 7 days; 68,792 out of 2,450,324 flows contain attacks. This set of data are labeled from 11th to 17th of June. 11th and 16th day datado not have attack; that is why these two days are not used in the training and testing phase. This dataset contains five types of attacks that their names are in table 3, and their definitions proposed in [8, 9]. Furthermore, details of data flow numbers and their attacks are in table 3.

TABLE III. DETAILS OF SELECTED DATA FLOW FROM ISCX 2012 DATASET

Date and type of attacks	Number of data flow	Number of attacks in data flow
12june-Brute Force	133,193	2,082
13june-Infiltrating The Network From inside	275,528	20,358
14june- HTTP Denial of Service	171,380	3,771
15june-Distributed Denial of Service	196,034	37,378
17june-Brute Force SSH	211,457	5,203
All numbers of Flows integrated in one file	**987,592**	**68,792**

In the last line of table 3, the total number of data flows and total attacks are highlighted. All of these data flows are applied to generate rules in the form of a merged file.

IV. EVALUATION OF PROPOSED APPROACH AND OBTAINED RESULTS

A. Evaluation Criteria

To evaluate this method, measures are represented in the forms of four formulas. These measures evaluate the worth of generated rules. Using these measures, we can decide which generated rule can be added to Snort rules' file. To this purpose, we considered P as the number of specific attack and N as the number of normal tags in tested data flow; accordingly, we have the following definitions: **True Positive (TP)**: number of data flows with specific attack tag, which have been detected correctly as the same type of attack using generated rule. **True Negative (TN):** number of normal data flow which has been detected correctly normal by using generated rule. **False Positive (FP):** number of normal or any type of attacks data flow which has not been detected correctly by using generated rule as the specific attack. **False negative (FN):** number of data flows of specific attack which have not been detected correctly normal or any type by using generated rule. According to these definitions, evaluation measures can be described in the forms of four formulas:

$$False\ Positive\ Rate = \frac{FP}{N} \quad (1)$$

$$Attack\ Detection\ Rate\ or\ Recall = \frac{TP}{TP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$F - Measure = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Formula (1) shows False Positive Rate (FPR) computed as the ratio between the number of normal or any type of attacks data flow that are incorrectly classified as specific attack, or False Positives and the total number of data flow. As we can see,thedistributionrate ofthe normaland anomaly(attack) data flow is not the samein data, or in other words, the number of normal data flow is several times of attacks, so to evaluate accurately we need formulas (2) and (3). Formula (2) shows the rate of specific attacks that are detected as the same attack correctly, to the total attacks to the total attacks of the same type that are detected as same attack correctly; this rats is also expressed as a percentage. In other words, Attack Detection Rate (DR) is computed as the ratio between the number of correctly detected specific attacks (True Positives) and the total number of attacks. Formula (3) shows that what percentage of attacks are detected as the same type of attack and has not been considered normal data flow as that type of attack by mistake. Formula (4) shows the harmonic meaning of two formulas (2) and (3). Ideal value for F-measure is 1. The closer to 1 the resultant value, the more valuable the generated rule. If obtained value for both formulas (2) and (3) is close enough to 1, then F-measure will be close to 1. In this research, the main measure to decide about generated rules will be the F-measure. Due to imbalance in normal and anomaly data, accuracy would not be a good measure for deciding.

B. Evaluation of Generated Rules

After running the Ripper and C5.0 algorithms on data, the rules are generated. The number of Ripper generated rules for attacks of brute force SSH, brute force, distributed denial of service, denial of service and infiltrating the network from inside are 8, 5, 12, 5 and 23, respectively. The number of C5.0 generated rules for attacks of brute force SSH, brute force, distributed denial of service, denial of service and infiltrating

the network from inside are 6, 1, 23, 30, 288 and 28, respectively. After investigating the generated rules, according to evaluation measures defined in section 4.A, the contents of best-obtained rules for any type of attacks resulting from algorithms implementation which have highest F-measure, was shown in tables 4 and 5.

TABLE IV. CONTENT OF BEST GENERATED RULES BY RIPPER ON ISCX 2012 DATASET

Content of Rules	TP	FP	Precision	Recall	F-Measure
if direction = R2L **and**totalDestinationBytes>= 2089 **and**totalDestinationPackets<= 13 **and**Avg_Src_Packet_Size(Byte) <= 151.142857 **then**Brute_Force_SSH_Attack	3483	2	99.94	66.94	0.801
if direction = R2L **and**totalDestinationBytes>= 2719 **and**totalDestinationBytes<= 3105 **and**totalSourceBytes>= 1399 **and** IOPR <= 0.941176 =>**then**Bruteforce_Attack	1985	0	1	95.34	**0.976**
ifAvg_Src_Byte_Per_Sec>= 3383.999918 **and** IOPR <= 0.460674 **and** IOPR >= 0.405556 **then**DDos_Attack	24967	422	98.33	66.79	0.795
ifAvg_Src_Packet_Size(Byte) <= 66 **and** direction = L2L **and**totalSourceBytes>= 320 **and**destinationPort<= 81 **and**Avg_Src_Packet_Per_Sec <= 2 **then**Dos_Attack	1213	0	1	32.16	0.486
ifdestinationPort>= 444 **and**Avg_Src_Packet_Size(Byte) <= 64 **and** direction = L2L **then** Infiltrating_Attack	16328	1150	93.42	80.20	0.863

Values of precision and recall are in percentage. Also in Table 4 and 5, generated rule for brute force attack has the highest F-measure value and we highlighted obtained highest F-measures of generated rules to be noticeable.

C. Discussion on Generated Rules

As shown in tables 4 and 5, rules are in the form of "if-then" that is "if" shows status and "then" shows rule's output (i.e. the type of attack). If all conditions of part "if" happen, then attack will be notified. Some attacks into network are implemented on specific ports. For example, brute force attack is implemented on the port 22 specifically.

TABLE V. CONTENT OF BEST GENERATED RULES BY C5.0 ON ISCX 2012 DATASET

Content of Rules	TP	FP	Precision	Recall	F-Measure
iftotalSourceBytes> 64**and**totalDestinationPackets<= 13**and** direction = R2L**and**destinationPort =22**then**Brute_Force_SSH_Attack	5088	17	97.78	99.68	0.987
iftotalSourceBytes> 1,391 **and**totalDestinationBytes> 2,665 **and**totalSourcePackets<= 18 **and**Avg_Des_Packet_Size(Byte) <= 228.667 **and** direction = R2L **then**Bruteforce_Attack	2041	5	99.75	98.03	**0.988**
if IOPR > 0.409 **and** IOBR <= 0.062 **and**Avg_Src_Byte_Per_Sec> 3165.999 **and**Avg_Src_Byte_Per_Sec<= 10145.004 **and** direction = L2L **then**DDos_Attack	32513	1877	94.54	86.98	0.906
ifAvg_Src_Packet_Size(Byte) > 64.387 **and**Avg_Src_Packet_Size(Byte) <= 66.055 **and** IOBR > 0.603 **and** direction = L2L **then**Dos_Attack	1572	92	94.47	41.68	0.578
iftotalSourceBytes<= 64 **and** IOBR > 0.985 **and** direction = L2L **then** Infiltrating_Attack	12002	792	93.80	58.95	0.723

In generating rule for this attack that has a high F-measure, this feature has been considered. Thus, based on explanation in section 3.B, destination port is largely effective on accuracy of generated rule.Another thing that can be noted is the effectiveness of significant new features in generated rules. As shown in Table 4 and 5,in all the rules generated by Ripper and all the rules except Rule of the brute force SSH attack generated by C5.0, thesenew features are used. Due to nature of infiltrating the network from inside, selected features could not achieve more than 0.9 F-measure. To achieve higher F-measure and detection rates of this type of attack,some other features along withthese features should be considered. Given thethe second scenario of ISCX2012 dataset, attacks designed to perform Dos attacks with low bandwidth and without need to flood the network is used, so features selected are unable to reach expected F-measure value(above 0.9) and the highest F-measure values obtained by Ripper and C5.0 are 0.486 and 0.578, respectively that is not very reliable.

D. Comparing Obtained Results

In this section, before presenting the conclusions, the related research is performed on ISCX 2012 dataset, and we compare the result with our presented approach. Intable 6, we compare our work with a related research, both performed on ISCX 2012 dataset.In this table, each five types of attacks are to separate days.Since intrusion detection can be seen as a classification problem, it is possible to apply common metrics used in the classification research area for comparison of previous works with our approach.To compare our approach with MIDAS, we use the precision criteria(formula 3). Value

of precision is stated in percentage. In MIDAS [21], a specific precision is not proposed for brute force attack, so we leave it blank in table 6. As shown in table 6 except for Brute Force SSH, in other attacks we obtained higher precision for all attacks. As you see in this table, the best value for each attack is specified in bold.

TABLE VI. COMPARING THE PRECISIONS OF THE BEST RULES OF OUR PROPOSED APPROACH WITH MIDAS

Attack type	MIDAS [21] Best rule precision	RIPPER Best rule precision	C5.0 Best rule precision
Brute_Force_SSH	**100**	99.94	97.78
Bruteforce	-	**100**	99.75
DDos	60	**98.33**	94.54
Dos	85	**100**	94.47
Infiltrating the network from inside	60	93.42	**93.80**

V. CONCLUSION

In thisstudy, we proposeda flow-based feature generation and automaticrule-generation intrusion detection systems in computer networks. According to investigation of recent works that are performed on ISCX 2012 dataset, this research is the first proposed work to generate rule on a whole dataset and its merged form. Also selecting these 17 features and their combination in order to generate rules is proposed for the first time. The main application of this study is for intrusion detection systems that need automatic updates. For future work, application of other features to dataset can detect other types of attacks. Also for data flow, time window can be considered, then in every window, the rate of sending and receiving packets and other features were investigated. This, in real-time and before attack is finalized, we can detect a suspicioustraffic data flow. In the other cases, we can investigate correlation between each other and achieve a subnet from it. One of the methods that can do this work is using evolutionary algorithm. For example, in genetic algorithm by using Mutation and Crossover operators, correlation feasibility can be investigated.

REFERENCES

[1] G. F.Alexandros,V. A.Siris, N. E. Petroulakis, and A. P. Traganitis, "Anomaly-based intrusion detection of jamming attacks, local versus collaborative detectio," wireless communications and mobile computing, doi: 10.1002/wcm.2341, 2013.

[2] M. Mahoney,andP.Chan, "Learning non stationary models of normal network traffic for detecting novel attacks," Proc. of 8th ACM SIGKDD Int. C. on KDD,1. (2002) 376-385.

[3] M. Kim, J.-H. Seo, I.-A. Cheong, and B.-N. Noh, "Auto-generation of Detection Rules with Tree Induction Algorithm,"in Fuzzy Systems and Knowledge Discovery, ed: Springer Berlin Heidelberg,vol. 3614, pp. 160-169, 2005.

[4] K. Scarfone, and P. Mell, "Guide to intrusion detection and prevention systems (IDPS)," NIST Special Publication 800-94, 2007.

[5] M. Roesch, "Snort, intrusion detection system,"June. 2015. [Online], Available: http://www.snort.org.

[6] V. Paxson, "Bro: a system for detecting network intruders in real-time,"Computer Networks, vol. 31, no. 23–24, pp. 2435–2463, 1999.

[7] H. Lai, S. Cai, H. Huang, J. Xie, and H. Li, "A parallel intrusion detection system for high-speed networks,"in Proc. of the SecondInternational Conference Applied Cryptography and Network Security (ACNS'04), pp. 439–451, May 2004.

[8] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Computers & Security, vol. 31, no. 3, pp. 357–374, 2012.

[9] S.-P. Oriyano, CEH: Certified Ethical Hacker Version 8 Study Guide: John Wiley & Sons, 2014.

[10] William.Cohen,"Fast effective rule induction," In Proceedings of the Twelfth International Conference on Machine Learning, pages 115–123, 1995.

[11] J. R.Quinlan, *"C5.0 Programs for machine learning,"* Morgan Kaufmann Publishers, San Mateo, California, 1993.

[12] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools,"ieee communiations surveys & tutorials, vol. 16, p. 34, 2014.

[13] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of IP flow-based intrusion detection,"Communications Surveys & Tutorials, ieee,vol. 12, pp. 343-356, 2010.

[14] C.A.Catania,andC.Garino, "Automatic network intrusion detection:Current techniques and open issues," Computers & Electrical Engineering, vol. 38, pp. 1062-1072, 2012.

[15] R.H. Gong, M. Zulkernine, and P. Abolmaesumi, "A software implementation of a genetic algorithm based approach to network intrusion detection,"In The sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2005 and the First ACIS International Workshop on Self-Assembling Wireless Networks(SNPD/SAWN 2005), IEEE Computer Society Washington, D.C., USA, pp. 246 – 253, 2005.

[16] Z. Bankovic, D. Stepanovic, S. Bojanic, and O. NietoTaladriz, "Improving Network Security Using Genetic Algorithm Approach," Journal of Computers & Electrical Engineering, vol.33, issue. 5-6, pp. 438-451, 2007.

[17] T. Vollmer, J. Alves-Foss, and M.Manic, "Autonomous rule creation for intrusion detection,"in Computational Intelligence in Cyber Security (CICS), 2011 IEEE Symposium on, pp. 1-8, 2011.

[18] J.Gomez, C.Gil, andR.Banos, "A Pareto-based multi objective evolutionary algorithm for automatic rule generation in network intrusion detection systems," Soft Computing, vol. 17, issue. 2, pp.255-263, 2013.

[19] C. Catania and C. Garcia Garino, "Towards reducing human effort in network intrusion detection," in Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013 IEEE 7th International Conference on, 2013, pp. 655-660.

[20] G. Kumar and K. Kumar, "Design of an Evolutionary Approach for Intrusion Detection," The Scientific World Journal,vol. 2013, p. 14, 2013.

[21] M. Milliken, B. Yaxin, and L. Galway, "The effect of attribute pairings in intrusion detection," in Computational Intelligence (UKCI), 14th UK Workshop on,pp. 1-6, 2014.

[22] Z. Tan, A. Jamdagni, X. He, P. Nanda, R. P. Liu, and J. Hu, "Detection of Denial-of-Service Attacks Based on Computer Vision Techniques,"Computers, IEEE Transactions on,vol. 64, pp. 2519-2533, 2015.

[23] K. A. P. Costa, L. A. M. Pereira, R. Y. M. Nakamura, C. R. Pereira, J. P. Papa, and A. Xavier Falcão, "A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks,"Information Sciences,vol. 294, pp. 95-108, 2/10/ 2015.

[24] B. Claise, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information," RFC 5101 (Proposed Standard), June. 2015. [Online]. Available: http://www.ietf.org/rfc/rfc5101.txt