Statistic Approached Dynamically Detecting Security Threats and Updating a Signature-Based Intrusion Detection System's Database in NGN

Gunay Abdiyeva-Aliyeva
UNEC Business School, Azerbaijan State Economic University, Baku, Azerbaijan
Email: gunay.abdiyeva-aliyeva@unec.edu.az

Mehran Hematyar
Cyber Security, Azerbaijan Technical University, Baku, Azerbaijan
Email: mehran@aztu.edu.az

***Abstract*—Cyber-attacks threatening the network and information security have increased, especially during the current rapid IT revolution. Therefore, a monitoring and protection system should be used to secure the computer networks. An intrusion detection system is very crucial on the market since it helps to control the network traffic and alerts the users during illegal access to the network. IDS is divided into three types: signature-based IDS, anomaly-based IDS, and both. Automatically updating the attack list to overcome new attack types is one of the main challenges of signature-based IDS. Most IDS or websites use recently detected attack signatures to update their databases manually or remotely. This article proposes a new AI model that uses a filter engine that functions as a second IDS engine to automatically update the attack list by AI. The results show that using the proposed model can improve the overall accuracy of IDS. The proposed model uses an IP-Factor (IPF) and Non-IP-Factor (NIPF) blacklist that can automatically detect the threats and update the IDS database with new attack features without manual intervention, as well as define new attack features based on similarity.**

***Index Terms*—intrusion detection system, signature-based, anomaly-based, traffic, AI based IDs, artificial intelligence**

I. Introduction

Having popularity of technology, the Internet (WAN), and Local Area Networks (LAN) in the past decades, the number of security attacks has been growing and developing rapidly, more than detection and defense. This can violate the privacy, integrity, and accessibility of computers and networks while performing critical activities, such as changing data and disabling services. Even with most advanced protection systems, computer systems are not highly (more than 96%) secure. Many companies have purchased security systems to protect from possible computer and network attacks, including firewalls, antivirus software, intrusion detection systems, access control, and encryption mechanisms [1]. Each of these mechanisms has disadvantages and deficiencies. For example, the focus of firewalls is only against data transmission which is not authorized and they do not provide anti-virus, anti-malware, or anti-spyware functions. Intrusion detection software cannot process encrypted software packages. Execution of antivirus software can cost too much computer memory and hard disk space, resulting in slower computer speed. The disadvantage of these security mechanisms is that intruders can be used, so they must be confused. Although IDS can be used with the Help of a firewall in a network, these two tools should not be considered the same tool [2]. (See Fig. 1, Table I and II).

Figure 1. Statistics on intrusions 2015-2020 (https://www.statista.com/).

TABLE I. General Structure of the Proposed Model 2015-2020

Year Cases	2015	2016	2017	2018	2019	2020
Number Of security breaches	1.210M	1.211M	1.20M	1.209M	1.207M	1.205M
Email spam rate	52%	53%	53%	55%	55%	56%
New malware variants	354M	355M	357M	359M	361M	363M

Manuscript received April 21, 2022; revised August 29, 2022.

doi: 10.12720/jait.13.5.524-529

TABLE II. STATISTICS ON INTRUSIONS SUCH ARE WEB ATTACKS AND RANSOMWARE FOR 2015-2020 (WWW.STATISTA.COM)

Year Cases	2015	2016	2017	2018	2019	2020
Number of web attacks blocked per day	1.210M	340K	229K	190K	160K	142K
Number of detected ransomware	34	34,556	338K	335	334, 450	333, 105
Average ransom amount	354M	229K	340,556	368, 205	380, 250	400, 510

New types of attacks can be carried out via many security systems as well as IDS and firewalls. Based on the factor, a strong, fast, and reliable IDS is urgently needed to control and prevent devices and networks from such events. In 2016 set the appropriate frequency threshold levels for improving the security of the databases based on the attack signatures and catching intrusion detection is very important problem with SIDS. Based on this factor, a unified algorithm (CA-NIDS) uses three databases to enable SBS to use a unified algorithm, an attack signature database, a new attack database, and a normal traffic database.

They selected a combination algorithm in order to analyze the engine, used 12 thresholds to sort the matching score values below the intrusion threshold, and entered classification values greater than or equal to the intrusion threshold value. In the article, we propose a new model that uses multiple smaller databases to install a filtering engine to detect new attacks after the IDS engine function [3].

II. MOTIVATION

Since most signatures IDS has many challenges, researchers are motivated to solve some of them. Every device or program at first cannot detect any behavior cause of is no sample.

A. The Initial Step

A new attack signature is hard to detect in time, since the signature-based type of IDS relies on a signature database to detect attacks.

B. The Second Step

The second activity is to dynamically update the database of known attack features and set a threshold for the frequency level to manage intrusion detection, the two considerably biggest problems facing feature-based, updated, and dynamically renewable IDS.

C. Third Step

This article describes the similarity between the signature of the new package and known signature, as the process of using two main factors (similarity and IP blacklist factor) to detect new attacks is explained. Also, the priority between these two main factors is seen as another problem which in turn should be solved.

The model proposed in this article will focus on processing large signature databases, detecting new attack signatures, and automatically updating IDS without administrator intervention, thereby improving signature-based IDS performance. It is divided into two small databases, the Smallest Signature Database (FSDB) and the additional database (CDB). FSDB will be distributed to three small databases based on protocol type and CDB [1]. However, the proposed model faces some challenges presented in the next section [4].

III. DIFFICULTIES AND SOLUTIONS

One of the main challenges discussed in the article includes a large count of signatures in IDS database. Therefore, these small signature databases can improve and improve the performance of the signature-based IDS, as software packages need to match fewer signatures but a huge database can decrease the performance and efficiency. When IDS is exposed to a large number of network traffic exceeding the monitoring potential of its, all it is able to do is dropping the packets. As a result, it may fail to detect the dangerous attacks. Increasing the performance of the model is one of the main challenges in the article. The proposed model, in fact, solves this problem by distributing complementary and small databases by protocol type to improve performance and reduces the amount of the time spent in the matching process. One of the other challenges is to detect new types of attacks due to incapability of signature-based IDS in detection of unknown attacks. Therefore, not updated database makes it easy to attack the network and override the IDS. Based on these factors, the proposed model provides a solution to this challenge by offering a new filtering engine for a double-check purposes as soon as the IDS engine, updating the complementary database and small database, and automatically updating without manual intervention without new intervention signatures. Another challenge is the way for a measurement of the similarity in the new packaged signature and the signature stored in the IDS database. The model mentions this problem on the basis of IP blacklist factors and source IP, target IP, packet load, and many features of the protocol used for detection of new types of attacks. Another challenge concerns about the determination of threshold for similarity without negative influence on IDS performance. The proposed model solves this problem by using variety of similarity thresholds and evaluating its output. Determination of the correct priority between blacklists and IP elements is the last challenge in proposal of model [5].

IV. Suggested Solutions

Having inspected and analyzed the proposed model, it is developed to detect new types of attacks not stored in IDS Database and process large signature databases and then update IDS databases with new types of attack signatures dynamically. As a result of these processes, the accuracy and performance of the IDS are improved. The proposed model aims to solve several issues that in turn are not covered in previous studies, such as detecting new attacks with signatures not stored in the IDS database and automatically updating the CDB and small database with new attack signatures without new attack signatures. It identified and developed previous work on large IDS database problems.

The component of the proposed model is the IDS engine, which is a CDB that stores all rare signatures and is distributed to three small databases based on protocol type (TCP, UDP, and ICMP). This protocol type is the most commonly used signature (TCP, UDP, and ICMP). It is distributed to three small databases according to the filter engine and update engine. The purpose of the IDS engine is to capture incoming data packets, process them beforehand and sign them, and then match the extracted signatures with signatures stored in the signature database by the statistics.

If there is a match or detect an unknown activity, the engine sends a warning, logs the warning, and blocks this package. Otherwise, the packet will be rechecked by the filtering engine on the basis of two factors (similarity and IP blacklist).

The variety of the stages illustrated in the chart above will be described in detail below.

- IDS Engine Stage.
- Training Stage: 1- Collect the ready-to-use attack feature dataset with 12,000 different attack features.

Using the previous dataset, create two additional databases as follows:

- The First database that stores the usual signatures happening during the dataset is classed as the Frequent Signature Database (FSDB)
- The Second database that stores the remaining part of the signatures occurring during the training stage, is called as the Complementary Database (CDB).

the Favorite Signature Database (FSDB) is the first database that contains the most common signatures. On the other hand, Complementary Database (CDB) is the second database which stores other signatures that appear during the process of training phase. 3- Signatures in the CDB and FSDB are distributed to smaller databases according to the type of protocol signed, that is, the CDB is distributed to these three mini databases [1]:

- The TCP database contains all signatures that use the TCP protocol.
- The UDP database includes those that use the UDP protocol All signatures. FSDB is distributed in the following three small databases:
- The TCP database contains all signatures that use the TCP protocol. UDP database contains all signatures that used the UDP protocol.
- ICMP database contains all signatures that the used ICMP protocol.

A. Testing Stage

When a new package arrives, it will be pre-processed and signed according to the protocol used. Next, the package signature will be transmitted to both FSDB and CDB to compare with the signatures which are stored in the two databases. Signature of the software package will be compared to the database containing the protocol type alone in order to minimize time spent on pairing and improve model performance. As soon as the package's signature matches the stored signature, the package will be blocked and saved, and an alert will be generated. If the signature of the package does not match the stored signature, the signature passes via the filtering engine in order to be determined if it has a new type of attack.

B. Similarity Factor

Check the similarity of the new packed signature with the stored signature in addition to AI behavior analysis. (For similarity, use only the same protocol signature as the signature of the new package.)

C. IP Blacklist

Use the stored blacklist IP to check the IP of the new package. If the signature of the new package does not match the filtering engine, the new package is safe and is eligible to enter the network. In case the signature of the new software package follows the filter engine's rules, CDB will add this signature immediately and store its APP address in order not to be included into the blacklist. Of course, “the package is blocked and an alert is generated in the other hand we propose a temporary blacklist” [6].

D. Filtering Engine

Determination of whether the packets in IDS engine can trigger any attack through applying three major factors to carefully examine. Mentioned three factors are the similarity and blacklist of IPs detailed in the next section and the priority mechanism between them.

E. Similarity Factor with AI

In order to measure the rate of similarity between the signature of new types of packages and the stored signature, the filter engine applies four factors: the source of the new package, the destination IP, the package load, and the package using the new protocol. Each property is assigned a default value (score = 0). If the newly packaged signature matches one of these functions, the value (score) changes from 0 to 20 and the match rate is more than 25%. In order to determine the final value of the score, the rest of the attributes will be checked and then the results will be compared with the similarity threshold. In case there is no match, the similarity between the newly wrapped signature and the stored signature based on determined similarity threshold will be measured by the filtering engine (only the performance of the model using the same protocol with the new packaging to match the stored signature). To reduce and improve pairing time). In case of having a similarity (score> = set threshold), new types of packages will be

blocked and automatically updated with the CDB signature, IP blacklist, and the IP of the new package. If no similarity is detected, it means that the new packet is quite clean and eligible to safely be directed to the network. When the signature of the new data packet passes through the IDS engine, the filtering engine first starts working to carefully check the IP of the new data packet using the IP blacklist. In case of observed match, this package will be blocked and automatically updated with the CDB signature [7].

V. Applications and Results

As stated above, the purpose of the model is to use signature-based IDS and update the new attack type. IDS and attack tools use the Java programming language for programming and development. The IDS tool is deployed to a virtual machine that in turn is running on the Windows OS and the attack tool to the same computer.

The dataset used at www.statistia.com was changed in 2019. It contains 12,000 rules or signatures and it is considered as a database storing the known signatures in the IDS engine. Moreover, 800 various types of attack signatures are used as test datasets to attack the IDS engine with attack tools. Since we are at the stage of verifying the concept, 12,000 rules are allocated in two databases. Here 14% of the rules will be stored in the FSDB while the remaining 86% of the rules will be stored in the CDB. Similarly, based on the findings in the literature, no standard or agreement exists in determination of the frequency threshold where signature is divided frequently and infrequently.

After collecting the common dataset (12000 800), FSDB and CDB were assigned 12000 signatures, and each signature was assigned to a smaller database by protocol type to improve IDS performance and reduce pairing time.

Program Installed

- Virtual Machine ESXi 6.7.
- HP SFF 8300 New. Rev 3.
- VM Mikrotik RouterOS v 6.35
- Wireshark 3.2.5.
- Windows 10 64-bit Professional.
- TCP /UDP Listener.
- NetBeans IDE.
- Jdk-8u261-8_2 windows-x64.exe.

Implementation and Testing

In this part, the experimental process implemented to perform and apply the proposed model is described. Based on the results of the proposed model, the analysis of outputs directly produced from the two tests performed with variety of machine specification is implemented.

VI. First Test

This test was performed with the following attacker and IDS machine specifications, as shown in Table III and Table IV.

TABLE III. Specifications of Attacker and IDS Machines

Attacker System			
RAM	CPU	Hard Drive (SSD)	OS
24 GB DDR3	Quad-core 2.4 GHz + i5 3rd gen Intel Processor	120 GB	Windows 10 – 64bit Professional
IDS System			
RAM	CPU	Hard Drive (SSD)	OS
4 GB DDR3	Dual-core 2.4 GHz + i3 3rd Intel Processor	24 GB	Windows 10 – 64bit Professional

Having sent 500 malicious packets directly from attacking tool to IDS, the attained results of the test 1 were received, as shown in Table IV.

TABLE IV. Test Outputs of the Proposed Model with One Large Database

Similarity Threshold	Initial Signatures count	Current Signatures count	Blocked packets	Allowed packets count	Detection ratio (Accuracy)	Time elapsed
20	12000	12100	806	0	100%	12
40	12000	12105	859	6	99.3%	16
60	12000	12010	840	6	98.2%	16
80	12000	12005	807	6	99.6%	18

VII. Second Test

As performed in the first test, same attacker and IDS machine specifications are used in the second test. The test is implemented, at first, with 12000 stored rules and signatures which are depicted in Table V.

TABLE V. How the Signatures Were Distributed Based on Protocol Type

Attacker System				
RAM	CPU	Hard Drive	OS	RAM
12 GB DDR 3	dual-core 2.4 GHz + i5 3rd Intel Processor	500 GB	Windows 10– 64bit Professional	12 GB DDR 3
IDS				
RAM	CPU	Hard Drive	OS	RAM
4 GB DDR 3	Dual-core 2.4 GHz + i5 3rd Intel Processor	120 GB	Windows 7– 64bit Professional	4 GB DDR 3

The results attained from the second test provided after 800 malicious packets sent from attacking tool to the IDS that is described in Table VI.

TABLE VI. TEST OUTPUTS OF THE PROPOSED MODEL WITH MULTIPLE SMALLER DATABASES

Similarity Threshold	Initial Signatures count	Current Signatures count	Blocked packets count	Allowed packets	Detection ratio (Accuracy)	Time elapsed (Performance)
20	12000	12100	806	0	100%	12
40	12000	12105	859	6	99.3%	16
60	12000	12010	840	6	98.2%	16
80	12000	12005	807	6	99.6%	18

VIII. RESULTS DISCUSSION

In this section, the final results of the performed tests during the experiment period will be discussed.

A. Results of Test 1

In this test, the way that the proposed model performed whilst using a huge volume of signature database that includes 12000 rules or signatures and 800 not-trained packets in testing is described. Based on this test, the portion of the detection, number of the detected packets and amount of time spent on testing are shown. Moreover, updated version of the historically given signatures database with recently caught attack signatures is provided.

B. Results of Test 2

In this test, how well the proposed model produced right results while utilizing variety of multiple smaller databases which is on the basis of protocol type with a total of 12000 rules or signatures and 500 unseen packets. In addition, positive improvement is observed that while using various databases that are smaller than a single database and detected new attack signatures and changed the version of the signature databases with a recently detected types of attack signatures without human aid by using the filtering engine.

While experimenting the model, by using the IDS and attacking tools that are programmed in Java programming language, the model is tested. Based on the performance of the model in testing, in terms of minimization of the size of the large volume of the databases of signatures, the improvements are attained. This method provided overall accuracy of the IDS that in turn determines a new type of attack signatures on the basis of the similarity and IP blacklists factors working together and updated the targeted databases with the new attack signatures without human interference [8]. A summary of the outputs is shown in Table VII.

TABLE VII. RESULTS OF THE PROPOSED MODEL'S TESTS IN TERMS OF ACCURACY AND PERFORMANCE

Similarity Threshold	Proposed model with one large database			Proposed model with multiple smaller database		
	Detection Ratio	Time Spent	DB Status	Time Spent	Detection Ratio	DB Status
20	100%	24 Second	12100	100%	20 Seconds	12100
40	99.2%	25 Second	12105	99.22%	21 Seconds	12105
60	99.2%	26 Second	12010	99.22%	23 Seconds	12010
80	99.2%	28 Second	12005	99.21%	24 Seconds	12005

Regardless of the machines' specifications that are applied in both of the models, the attained results of the model proposed by us are compared with the results of the [9]'s model from the point of accuracy. The results are illustrated in Table VIII.

TABLE VIII. COMPARISON OF MODEL ACCURACY RESULTS BETWEEN THE OUTPUTS OF BOTH THE PROPOSED MODEL AND [9]'S

Model	Accuracy
Proposed model	99.41
[9]'s Model	96.5

Regardless of the machines' specifications that are applied in both of the models, the attained results of the model proposed by us are compared with the results of the [10]'s model from the point of accuracy. The results, having sent 500 malicious packets, are depicted in Table IX.

TABLE IX. A COMPARISON BETWEEN THE OUTPUTS OF BOTH THE PROPOSED MODEL AND [10]'S MODEL IN TERMS OF PERFORMANCE

MODEL	TIME SPENT
Proposed model with multiple databases	0.044 s each
[10]' s model with multiple databases	0.001844 s each
Proposed model with one large database	0.0515v s each
[10]' s model with one large database	0.017780 second each

IX. CONCLUSION

This paper first emphasized the motivation to write this article. Then he introduced possible difficulties and contributions. Then, the model proposed by us is designed and output. This article describes how IDS performs when deploying large databases to smaller databases. As for the contribution of this article, the proposed solution provides detection of new attacks with unknown IDS signatures. This solution usually connects to a proposed filtering engine that uses two factors to detect new attacks.

Four similarity factors based on the measure of similarity between the characteristics of the signature and the stored signature of a new package: the source of a new package, the IP target, and the data packet load of a new package.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Gunay Abdiyeva-Aliyeva and Mehran Hematyar conducted the research; analyzed the data and wrote the paper; all authors approved the final version.

ACKNOWLEDGMENT

The authors wish to thank Mr. Mammadzada Asiman for edition of the paper in terms of the writing style.

REFERENCES

[1] A. H. Almutairi and N. T. Abdelmajeed, "Innovative signature-based intrusion detection system," in *Proc. International Conference on the Frontiers and Advances in Data Science on Information Technology*, 2017, pp. 1-7.
[2] H. Debar, "An introduction to intrusion detection systems," *Proceedings of Connect*, pp. 1-18, 2000.
[3] [Online]. Available: https://www.statistia.com
[4] O. Folorunso, F. E. Ayo, and Y. E. Babalola, "Ca-NIDS: A network intrusion detection system using a combinatorial algorithm approach," *Journal of Information Privacy and Security*, vol. 12, no. 4, pp. 181-196, 2016.
[5] P. Innella and O. McMillan, "An introduction to intrusion detection systems," 2001.
[6] Y. Mutep, A. Yousef, N. T. Abdelmajeed, "Dynamically detecting security threats and updating a signature-based intrusion detection system's database," *Procedia Computer Science*, vol. 159, pp. 1507-1516, 2019.
[7] Kaspersky Security Bulletin 2019. Statistics. (12 Dec 2019). [Online]. Available: https://securelist.com/kaspersky-security-bulletin-2019statistics/95475/?utm_source=securelist&utm_medium=blog&utm_campaign=gl_ksb-stats_ay0073&utm_content=banner&utm_term=gl_securelist__ay0073_banner_blog_ksb-stats
[8] M. Uddin, K. Khowaja, and A. A. Rehma, "Dynamic multi-layer signature based intrusion detection system using mobile agents," *International Journal of Network Security & Its Applications*, pp. 129-141, 2010.
[9] L. Rademacher, "The disadvantages of intrusion detection systems," 2017.
[10] S. D. Sheenam, "Comprehensive review: Intrusion detection system and techniques," *IOSR Journal of Computer Engineering*, vol. 18, no. 4, pp. 20-25, 2016.

Gunay Abdiyeva-Aliyeva was born in 1985 in Baku, Azerbaijan. She graduated with Information System Engineering BSc (2003-2007) and a Master's degree (2008-2010) from Azerbaijan Architecture and Construction University. In 2018, she completed her doctorate at the Institute of Control Systems of the Azerbaijan National Academy of Sciences. She has a Doctorate in science - System analysis, control, and information processing. In 2020, she was admitted to the doctorate at the same institute. Her dissertation work is "Development of systems for the detection and prevention of cyber-attacks using artificial intelligence methods". Also she won a postdoc grant from the Islamic Development Bank. With the consent of the same dissertation, she continues her postdoctoral research in the Department of Digital Forensics of Firat University.
She has published more than 40 articles and attended about 15 international symposiums in this area.
She is a member of IEEE. Since 2020 she had been working at the Azerbaijan State Economic University as a docent.

Mehran Hematyar was born in 1986, in Tehran, Iran. He graduated with the Information security Engineering BSc (2003-2007) from Tehran University and a Master's degree (2017-2019) from Azerbaijan Technical University. In 2021, he was a candidate for the Ph.D. in the Department of Digital Forensics of Firat University. His dissertation work is "Development of systems for the detection and prevention of cyber-attacks using artificial intelligence methods".
He is currently resident in Baku, and working as a senior lecturer in BEU, UNEC, DGKA, and AZTU since 2021, so he continues his research here. He had published more than 10 articles and attended about 15 international symposiums in this area.
He is a member of IEEE, Cisco Engineers, Huawei, Hikvision, Mikrotik, and Bosch security systems societies. Also, he works for Cybernetics co in the network security department as CTO.