

IMPOSTERS ANOMALY DETECTION

A. Tazerouti*, A. Ikram

School of Computing and Engineering, University of Gloucestershire, Cheltenham,
Gloucestershire, United Kingdom

Received: 13 August 2020 / Accepted: 23 October 2020 / Published online: 01 January 2021

ABSTRACT

Over the last two decades the world of cyber security has grown immensely, but despite the state-of-the-art security detection systems and intrusion detection systems (IDSs), unwanted malicious users still find their way around these security measures and gain access to secure systems. This study consists of shedding some light on the security issues in the intrusion detection systems, their vulnerabilities and drawbacks. A hypothesis is proposed to help mitigate these issues and obtain a fast and a more precise method for the detection of different malicious intruders and imposters, study their behavior and make a statistical comparison of data from the used IDSs and throughout the process. This study will state the current available technologies of IDSs, site their challenges and implement a new software-based methodology to increase the detection and reduce false alarm rates for the IDS.

Keywords: Cyber Security; Intrusion Detection System; Software-based detection; Keystroke Dynamics; Network-based detection.

Author Correspondence, e-mail: glositcs.18@gmail.com

doi: <http://dx.doi.org/10.4314/jfas.v13i1.14>

1. INTRODUCTION

1.1. Background

The increase in hyper connectivity caused by the phenomenon called internet of things (IoT) means that a massive number of devices are connected to the internet, exchanging data. This interaction offers a number of business and social opportunities with the said devices providing various services for businesses and citizens. The challenge, however, is in ensuring the security of these large amounts of data exchanged by the connected devices. The devices (and the network) are vulnerable to an ever-changing threat landscape. To this end, sometimes the devices behave in a misguided way due to an intrusion that was not detected, which brings us to the role of intrusion detection systems (IDSs).

According to [1], originally, IDSs showed up in the late 70s, at first administrators used to sit facing a console all day monitoring the user activities and reviewing any unusual activity or behaviour. This was (and in many cases is still) done manually by checking the audit logs with the naked eye after printing them on a folded paper, which will often end up stacked with other pile of papers of five feet high by the end of the week. This manual approach came to an end by the early 90s, when researchers developed a new IDS that reviews real-time audit data capable to detect attacks as they occurred.

According to [2], in the last two decades, t IDSs have become an area of research of growing interest in the computing security field, due to its importance to companies and organizations, who are keen to preserve the integrity, confidentiality, and availability of their data.

1.2. Overview

Intrusion is referred to as “when an intruder tries to break into a system or perform an unwanted or illegal action” [3]. There are two types of intrusion, internal and an external.

These two groups of intrusion contain several techniques, which include the exploitation of system mis configuration and software bugs, password hacking, traffic sniffing and exploiting the flaws and loopholes in a certain design protocol [3].

In other words, an IDS is a system that detects and reports intrusions accurately to the authority. Two categories of techniques are known for Intrusion detection, namely anomaly detection and misuse detection.

Anomaly detection is a bunch of techniques that characterize the normal behaviour from the

unordinary one such as execution time and CPU usage. The behaviours with a deviation are considered as an intrusion.

On the other hand, misuse detection techniques are the known methods used to penetrate the system. These penetrations are displayed as signatures, or patterns, that the intrusion detection system look for. The system responds only to the identified penetration, it might be a set of actions or a static string.

However, despite the major development in cyber security technologies, IDSs features are still not sufficient to detect several anomalous behaviours and even if they are they will not act in real time to prevent or mitigate the damages [4].

According to [5], anomaly detection IDSs are divided into three types: Network-Based IDS, Host-Based IDS and Hybrid IDS.

There is a difference between firewalls and NIDS. A firewall is a device that performs active and passive defense. It is situated in a network and filters outgoing and incoming traffic based on rules that are predefined. It is considered the first line of defense and has the ability to filter and block packets and unwanted traffic.

Firewalls regularly record the logs of multiple activities and analyzing them can be a level of intrusion detection since the analysis requires looking for patterns and spotting differences, but the firewall analysis is not suitable for the role since it consumes time and it is rarely performed, and on the top of that it weakens the firewall since it has to carry out several tasks at once [6].

NIDS, on the other hand, are passive systems, situated in the demilitarized zone (DMZ), capture and analyze the traffic in the network and spot the anomalies, and the only defensive action that could be made is sending an alert to the administration. To put it in a simple real-life example, firewalls are like a security guard, and the NIDS operate like a security camera.

Numerous research papers were published on intrusion detection systems. In these researches four techniques were mentioned, namely Anomaly-based detection, Knowledge-based detection, Multi-agent-based detection and Hybrid detection [5].

-Anomaly-based Detection

According to [7], anomaly-based detection used to analyze behavioral patterns and detect the ones that don't match the profile of the system. This technique can be divided into four aspects: nature of the data, type of anomaly, contextual, individual or collective and labeling of data

instances as normal and abnormal.

These aspects include the use of anomaly detection approaches such as supervised learning and semi-supervised learning [8].

-Knowledge-based Detection

According to [9], knowledge-based detection systems are computers that already have saved data or knowledge and are able to resolve intrusion issues.

Knowledge based detection contains three features, which are Knowledge-Base, Interface Engine and User Interface [10]. Knowledge based detections is a very beneficial IDS technique due to its numerous advantages, such as minimizing the human interaction, efficiency, consistency in replies and detailed explanation for the given solutions [10].

-Multi-Agent Based Detection

According to [11], the definition of an agent is basically an autonomous entity that has the ability to comprehend its entourage through sensors and then act on that through effectors. The agents have factual knowledge body just like humans, but their factual knowledge is based on machine learning, knowledge-based systems and heuristics. They also possess a belief and a goal, which enables them to adapt and overcome their environment.

A Multi-Agent System is a network of agents that can work cooperatively, or independently, in order to achieve a shared goal [12].

Multi-agent systems are widely applied in multiple domains such as intrusion monitoring, intrusion, detection systems, CPS system' fault monitoring and data gathering through techniques such as:

- Centralized Multi-Agent-Based Detection: a well-known centralized multi-agent detection technique that detect complex attacks like distributed denial of service attacks, spams and botnets, by using a group of local agents.
- Collaborative Multi-Agent-Based Detection: this approach is used for an autoconfiguring infrastructures like VANETs.
- Cooperative Multi-Agent-Based Detection: this approach assumes the cooperation of the agents together by their interaction, the goal is shared and can be a detection of an attack orto defend a system.

-Hybrid Detection

Hybrid detection is a technique that tends to combine more than one approach such as deep learning, classification, multi-agent systems, knowledge base and regression.

In the recent years, hybrid detection was frequently used as it is more adequate and provides a better accuracy in terms of common threats detection, [5].

1.3. Issues and Challenges

IDSs have their advantages and disadvantages, so to evaluate their performance, the latest proposed approaches ought to be compared against each other.

Even though IDSs bring several benefits, they have multiple drawbacks and challenges that need to be addressed, especially as these drawbacks are considered as vulnerabilities that could be exploited by attackers. Some of the vulnerabilities and drawbacks of anomaly detection that this study focuses on include:

- Some attacks may not be detected, such as an imposter who gets an access to an account that belongs to a legitimate user, particularly if the attacker fits the established profile of the user.
- Anomaly detection systems cannot detect a behaviour of a malicious user who realises that s/he is being profiled and slowly changes their behaviour to normal overtime.
- There are several high rate false positive and false negative cases
- Slow and late detection of attacks in current technologies of intrusion detection systems.

2. METHODOLOGY

Since the intrusion detection systems still have many unsolved issues, such as accuracy, detection speed and false alarm for false negative and false positive intrusion, this study aims at reducing the problem thus increasing the security level in intrusion detection systems.

The Research methodology adopted by this study from the point of view of application, is an applied research methodology using a quantitative approach because it is a methodology used to quantify issues through the generation of numerical data and convert it to a usable statistic.

The data comprises defined variables of behaviour patterns and signatures, these patterns are a result of several analyses arrived at from data collected from many users and computers. The

data is captured using tools such as Wire shark and Event Viewer and Keystroke Dynamics Software, with a view to analyse the usage and spot the different variations in the behaviour throughout time. A statistical graph is then used to depict the deviation between the abnormal behaviour and the normal one.

This study, therefore, aims to detect imposters and undetected intruders by analysing current and old behaviour of a certain user who has been using the system, and comparing the changes overtime. To do this, this study explores and develops a precise detection method through combination of three different types of intrusion detection systems techniques. These IDSs are, Network-Based IDS, Host-Based IDS, and Keystroke Dynamic IDS (fig. 1).

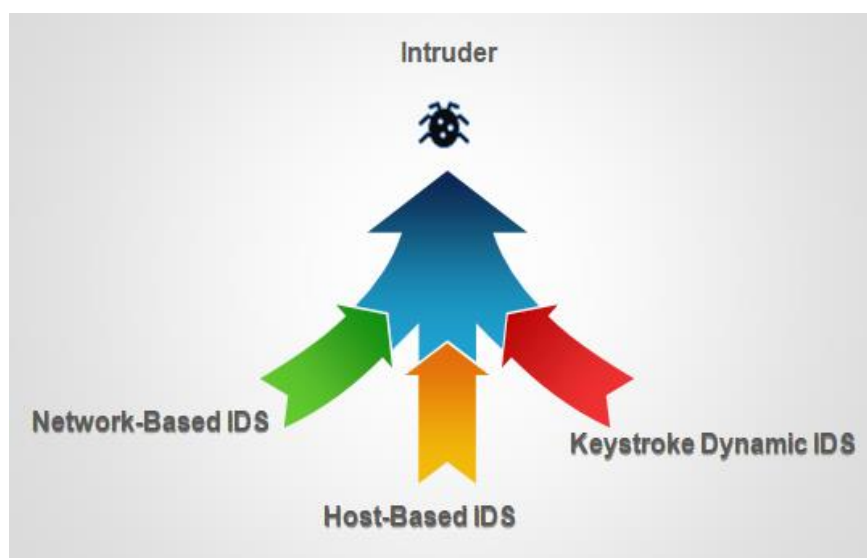


Fig.1. The Proposed Methodology

2.1. Implementation and Testing

2.1.1. Design and Implementation

In the world of cyber security, the human is always considered as the weakest security link, because of his detrimental mistakes, be they deliberate or unintentional. Often, lack of awareness leads to several consequences such as passwords getting stolen or revealed giving malicious users the opportunity to gain unauthorised access to data and systems.

As mentioned in the problem statement, some attacks cannot be detected in the current anomaly detection systems, such as an imposter who gets access to an account that belongs to a legitimate user, particularly if the attacker fits the established profile of the user.

The meaning of an attacker, whom fits a profile of a legitimate user, is an imposter who studies the daily usage behavior of a user in a particular system, and copies the same steps the user does in his routine, in order to hide his usage and remain undetected by the intrusion detection systems.

For this matter, this study consists of designing a software-based detection methodology that enables the detection of such intrusion by analyzing the usage of the user through a chain of connected software.

The design of this methodology consists of the following architecture in figure 2:

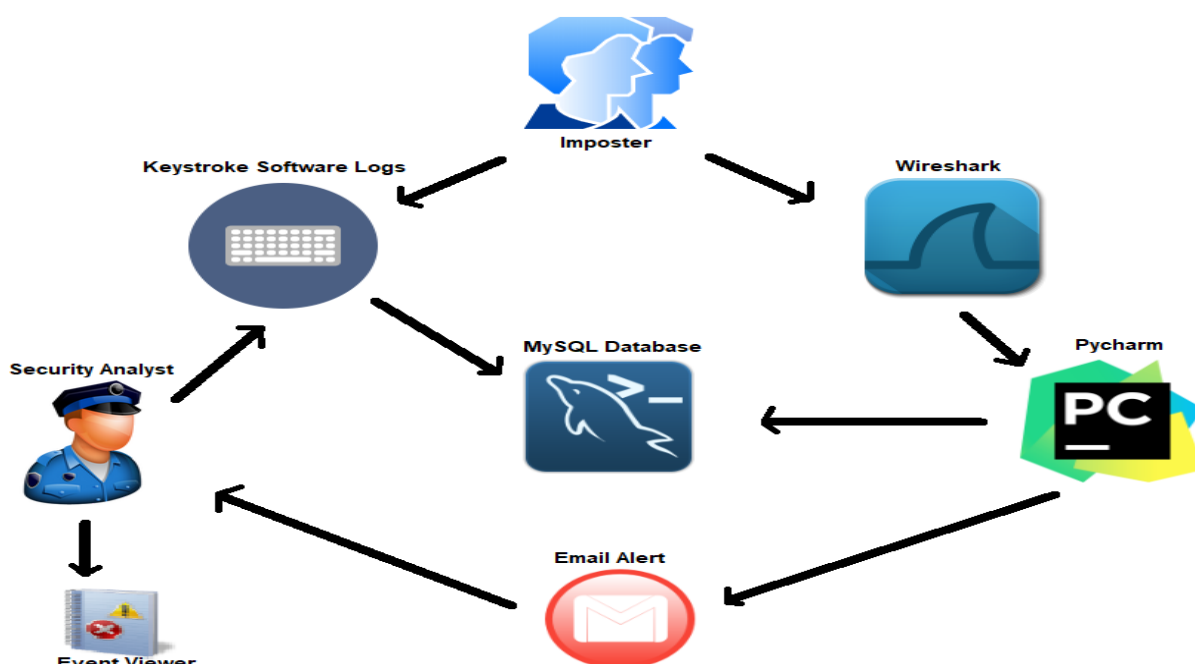


Fig.2. The Architecture of the Proposed Methodology

In this architecture there are eight main elements, each one of them has its own role, these elements are described as follows:

The Imposter: the unauthorized user that steals the password and studies the usage behavior of a legitimate user in order to steal information from the system and remain undetected.

Keystroke Software: is the software where the user types the password to log into the system, it displays the user name, user ID, successful login, wrong input password and four graphs that describe the typing method of the user such as the time between two keys pressure, time between two keys release, time between one release and one pressure, and time between one pressure and one release, based on X scale that represents the duration and Y scale that

represents the number of characters in the password, the mentioned variables are compared with the standard user's typing method as demonstrated in figure 3.



Fig.3. Keystroke Authentication Data (Original photo: A. Tazerouti)

The imposter detection in this stage is confirmed via the four graphs' average difference between the standard user and the other users, where the limit of minimum average duration difference is set in 0.5 millisecond if the average difference is over this duration, then the user is considered as intruder and require verification and analysis.

MySQL Database: by connecting to MySQL, and creating a new database that contains two tables:

The first one is named (users) it is assigned to keystroke Software to record the login time and date, the username and how many times the user was activated as in figure 4.

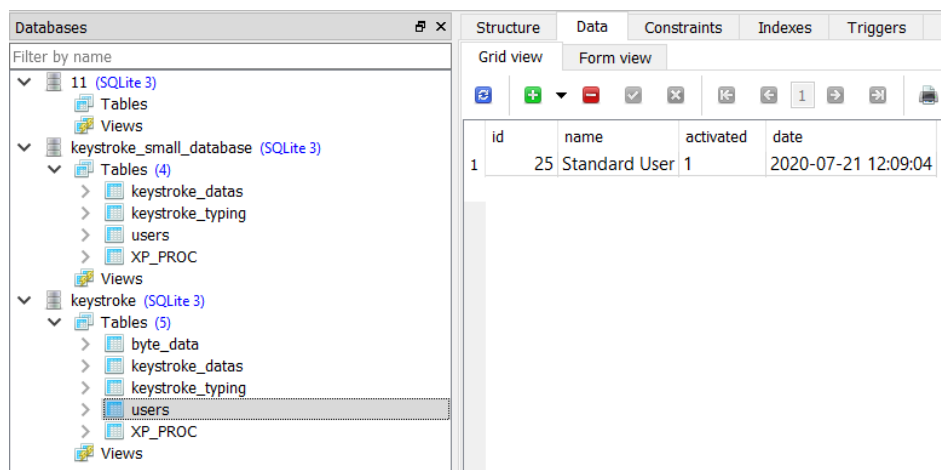


Fig.4. Keystroke Users Database Table

The second table is named (byte data) for the packet length of the captured data from Wireshark, the table is divided into 4 elements such as the user ID, username, the total packet length and type of usage.

The ID is the number assigned to the user after enrolling.

The username is the name of the user that used the specific IP address during the given time.

The total packet length is the total data used by the user when utilizing the internet in the given period of time, and calculated by megabytes (fig. 5).

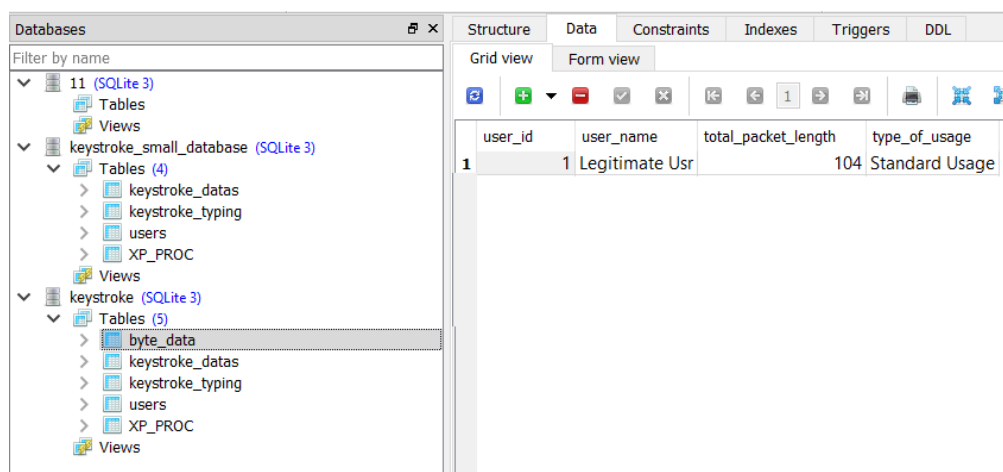


Fig.5. Packets Length Database Table

Wireshark: it basically captures all the network traffic that comes and go from the system's specific IP address and save it in a npcap file which will contain all the stored data during the capture period (fig. 6).

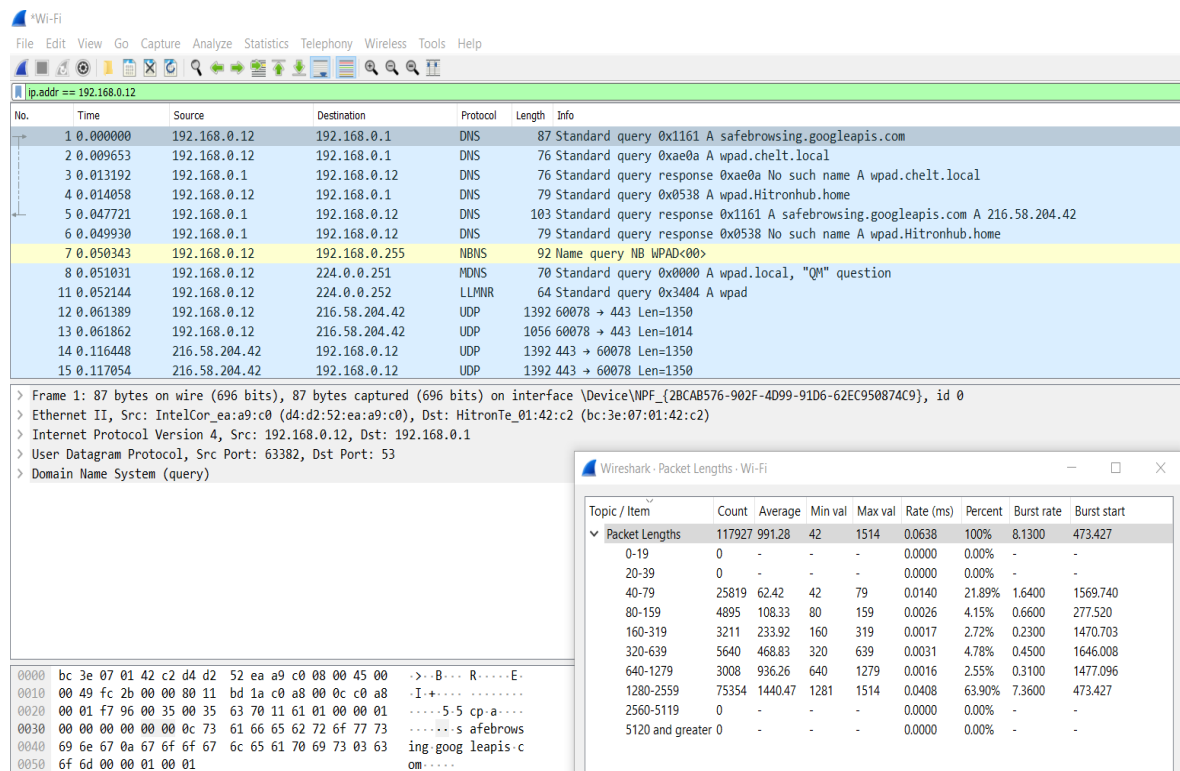


Fig.6. Wireshark Traffic Capture for a Specific IP Address

PyCharm: is a python-based IDE software, it contains an algorithm that extracts the packets length of the traffic captured by Wireshark and store it in the database.

The algorithm consists of a code that connects the Wireshark database to MySQL database, and extracts the packets length from the Wireshark database, put in a function that calculates the usage by number of packets x the length range of each packet and calculate it in megabytes instead of bytes, using the following code:

```
pkt_lists = rdpcap(pkt_file)
```

```
print(len(pkt_lists))
```

```
totalPktLength = 0
```

```
for pkt in pkt_lists:
```

```
    if IP in pkt:
```

```
total,PktLength += len(pkt)
newTotalPktLength = total,PktLength / 1000000
print(totalPktLength)
print(newTotalPktLength)
```

Then store the calculated data in the byte data table in the created database depending if it is limited usage or standard usage. The algorithm also connects to a server, with an inputted email address that serves to send a direct automatic email alert if the packets lengths is fifty megabytes under or over the standard usage

2.1.2. Testing

A. Functional Testing

In Order to confirm the results of this methodology, this study had to run an experimental test to confirm its functionality.

For the experiment, the three software were customized to meet the required data such as:

Keystroke Authentication: Assign a standard password so that the enrolled users use in order to authenticate the changes of each password typing behavior (fig.3).

Wireshark: Assign a specific IP address of the used computer in order to capture only the packets that are related to that specific IP address, figure 6.

Event Viewer: filter the events that happen in the last thirty minutes.

To begin, we should divide the experiment into two steps:

Step 1: Enrol standard user as a “Legitimate User” this legitimate user is the reference on how the behaviour should be, this user will be put into a thirty minutes experimental usage of a computer, where a specific destinations are assigned to him, which the enrolled user has to follow.

The guidelines of the usage in this study were:

- Write the assigned password in the Keystroke software.
- Run Google Chrome, once the user is on the internet, he is only allowed to watch two YouTube videos, access his email and do a google search.
- Access the program files in the local disk (C:)

After the end of the 30 Minutes experiment, we run the designed algorithm for this methodology

in order to obtain the ultimate usage behavior of the legitimate user and store it in the database tables 1 and 2 (fig. 4, 5).

Step 2: Enrol the first user labelled “User 1” and run the same previous points that were given to the standard user mentioned in step 1.

After assigning a password in the Keystroke authentication software, the graphs display the difference between the two users, the red one being the legitimate user and the green one being user 1 (fig. 7).

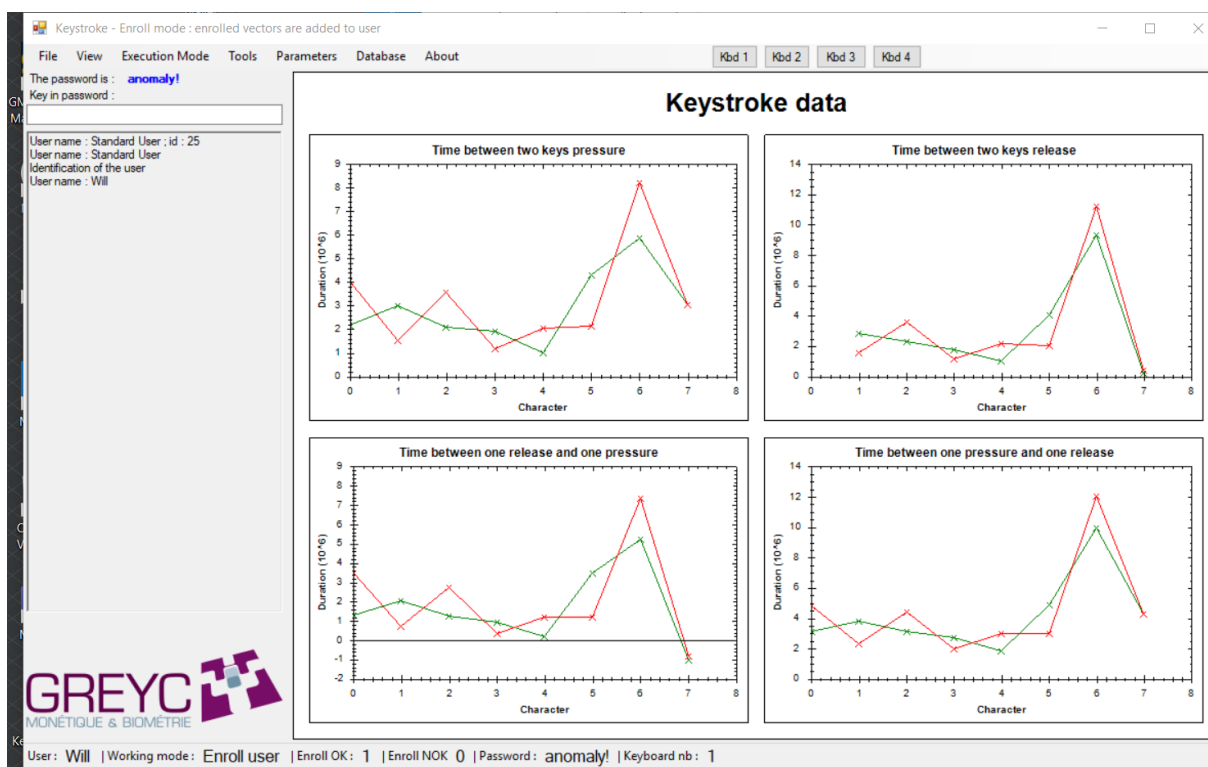
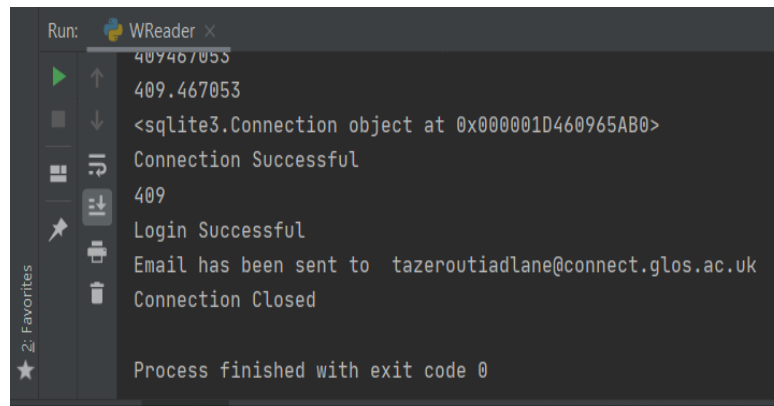


Fig.1. Comparison of the Keystroke data of the Standard user and User 1

After running the designed code see figure 8, it is shown that:



```

Run: WReader x
409467053
409.467053
<sqlite3.Connection object at 0x000001D460965AB0>
Connection Successful
409
Login Successful
Email has been sent to tazeroutiadlane@connect.glos.ac.uk
Connection Closed

Process finished with exit code 0

```

Fig.8. Results of Internet Usage

- The connection between the Wireshark database and the MySQL database was successful.
- The calculated packets length used during the given period.
- Login successful to the server and the email.
- Email alert sent to the administrator since the usage is over 150 megabytes

The results are stored in the database created tables as demonstrated in figures 9 and 10.

	id	name	activated	date
1	25	Standard User	1	2020-07-21 12:09:04
2	27	User 1	1	2020-07-23 17:16:26

Fig. 9. Keystroke Database Table

user_id	user_name	total_packet_length	type_of_usage
1	Legitimate Usr	104	Standard Usage
2	User 1	409	Limited Usage

Fig.10. Wireshark Packets Length Database Table

The third similarity check is through Event Viewer, where the final analysis is made, if the first two intrusion detection systems did not show an abnormal activity, Event viewer will be able to detect any small anomalies that happened on the running events on the given period.

B. Performance Testing

After confirming the functionality of the methodology, it is required to run an experiment with several users in order to demonstrate the detection accuracy.

For this experiment, the test is similar to the previous one except it is done with eight more users to increase the number of users to ten users overall, this test is made in order to demonstrate how ten users with the same guidelines of usage can be very different and output several usage anomalies.

The figure 11 demonstrates the final display of Keystroke graphs data, with the standard user in red color and the other ten users in green color.

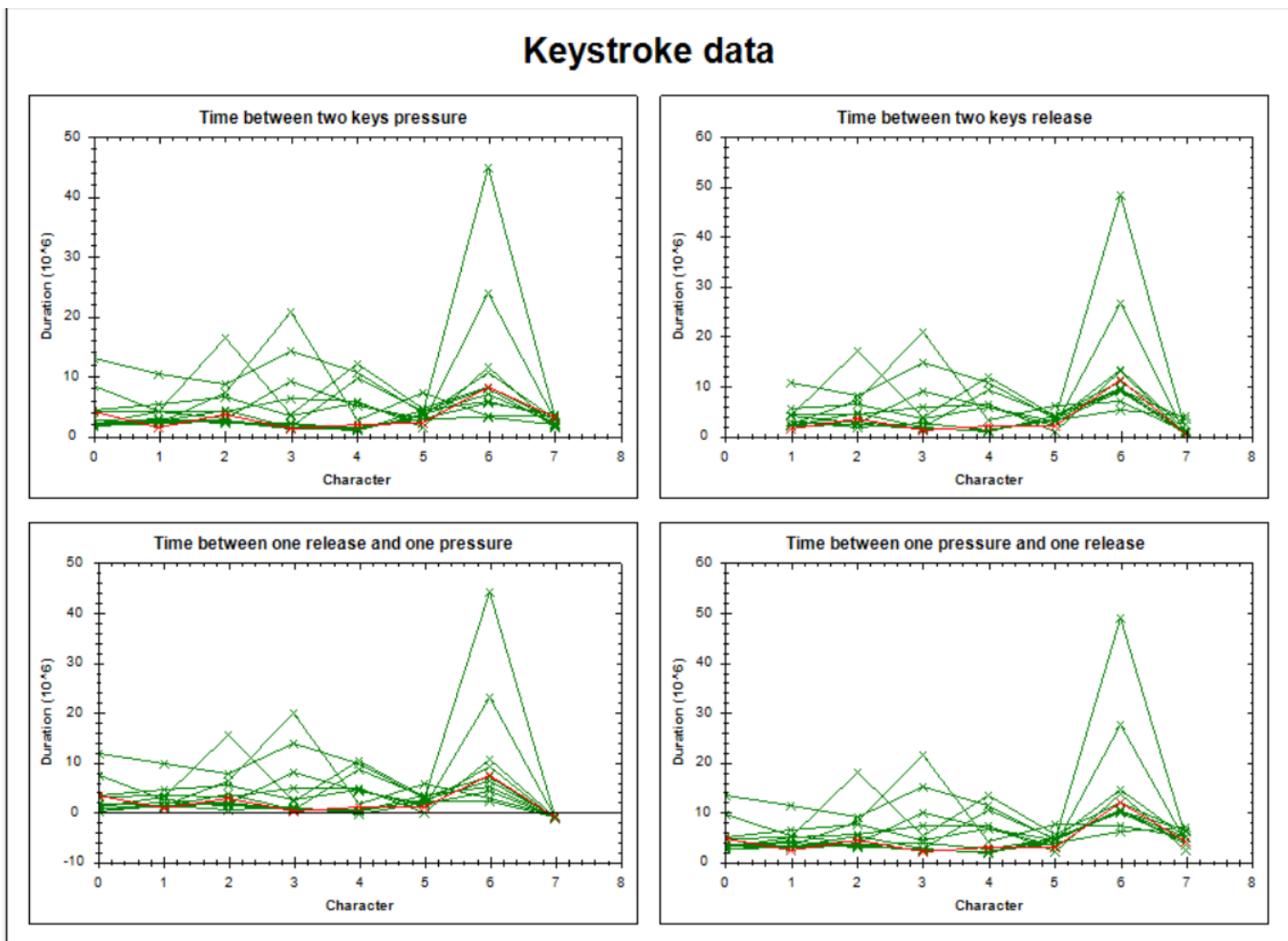


Fig.11. Keystroke Authentication display of the users

the average duration difference of each of the four graphs between the standard user and each user is calculated based on the following function that calculates the Absolute sum of difference between the legitimate user and the rest of the users, divided by eight, which is the number of characters in the given password.

$$\frac{\left| \sum (y^{\text{legitimate user}} - y^{\text{user}}) \right|}{8}$$

The figure 12 demonstrates the final table of user's logins by date and time of login.

	id	name	activated	date
1	25	Standard User	1	2020-07-21 12:09:04
2	27	User 1	1	2020-07-23 17:16:26
3	28	User 2	1	2020-07-23 21:01:31
4	29	User 3	1	2020-07-25 15:17:08
5	30	User 4	1	2020-07-25 16:52:53
6	31	User 5	1	2020-07-25 20:13:17
7	32	User 6	1	2020-07-25 21:55:15
8	33	User 7	1	2020-07-25 22:33:34
9	34	User 8	1	2020-07-27 10:37:09
10	35	User 9	1	2020-07-27 11:14:47
11	36	User 10	1	2020-07-27 12:25:47

Fig.12. Keystroke Logins Final Table

The figure 13 demonstrates the final table of the used packets length during the test of each of the ten users.

	user_id	user_name	total_packet_length	type_of_usage
1	1	Legitimate Usr	104	Standard Usage
2	2	User 1	409	Limited Usage
3	3	User 2	194	Limited Usage
4	4	User 3	120	Limited Usage
5	5	User 4	284	Limited Usage
6	6	User 5	235	Limited Usage
7	7	User 6	163	Limited Usage
8	8	User 7	131	Limited Usage
9	9	User 8	192	Limited Usage
10	10	User 9	169	Limited Usage
11	11	User 10	143	Limited Usage

Fig.13. Total Packets Length Usage Final Table

3. RESULTS AND DISCUSSION

The Experiment demonstrated several results. These results were taken after each test, where the anomaly was detected by at least two systems, because when only one of the two first detection line systems (Keystroke and Wireshark) detect the abnormal activity the final analysis is done on Event Viewer to confirm if it is a false alarm or a real intrusion, to put it in a much simpler way. Table 1 shows the type of detection system that detected the anomaly in the usage.

Table 1. Source of Anomaly Detection

Users	Detection Software		
	Keystroke Authentication	Wireshark	Event Viewer
User 1		✓	✓
User 2	✓	✓	
User 3	✓		✓
User 4	✓	✓	
User 5		✓	✓
User 6	✓	✓	
User 7	✓		✓
User 8	✓	✓	✓
User 9	✓	✓	
User 10	✓		✓

From the table it is shown that eight out of the ten users were detected via Keystroke Authentication, that is a detection success rate of 80%, that's because of the low number of similarity average difference between each of the user's graphs and the standard user's graphs (fig. 14).

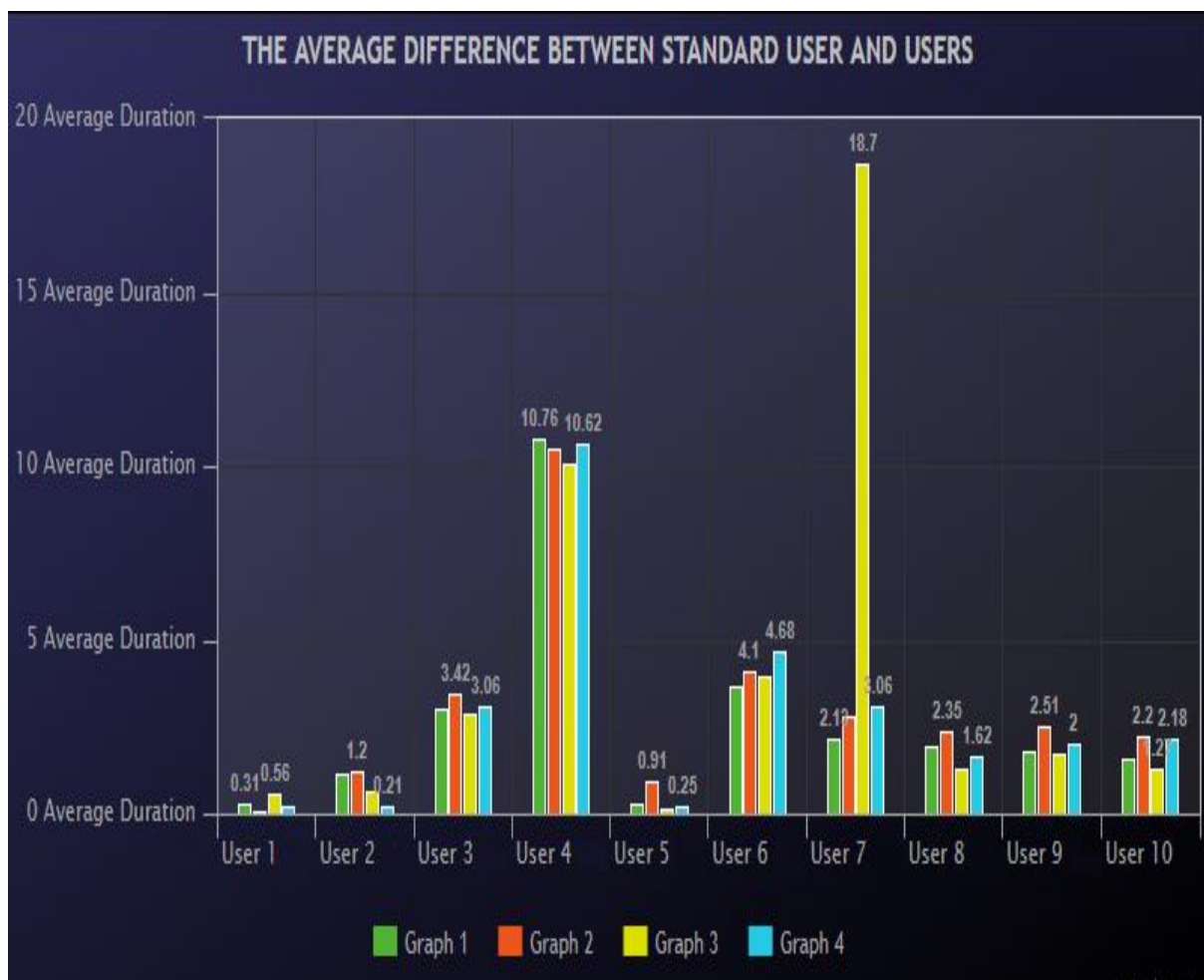


Fig.14. Multi-Bar Chart of Average Duration Difference Between Standard User and Users

Whereas seven email alerts were sent to the administrator's email address due to passing the given limit of usage of internet data which was set between 150 megabytes and 50 megabytes, see figure 15, which means that seven out of ten users were detected via their abnormal internet usage captured via Wireshark which makes a detection success rate of 70%.

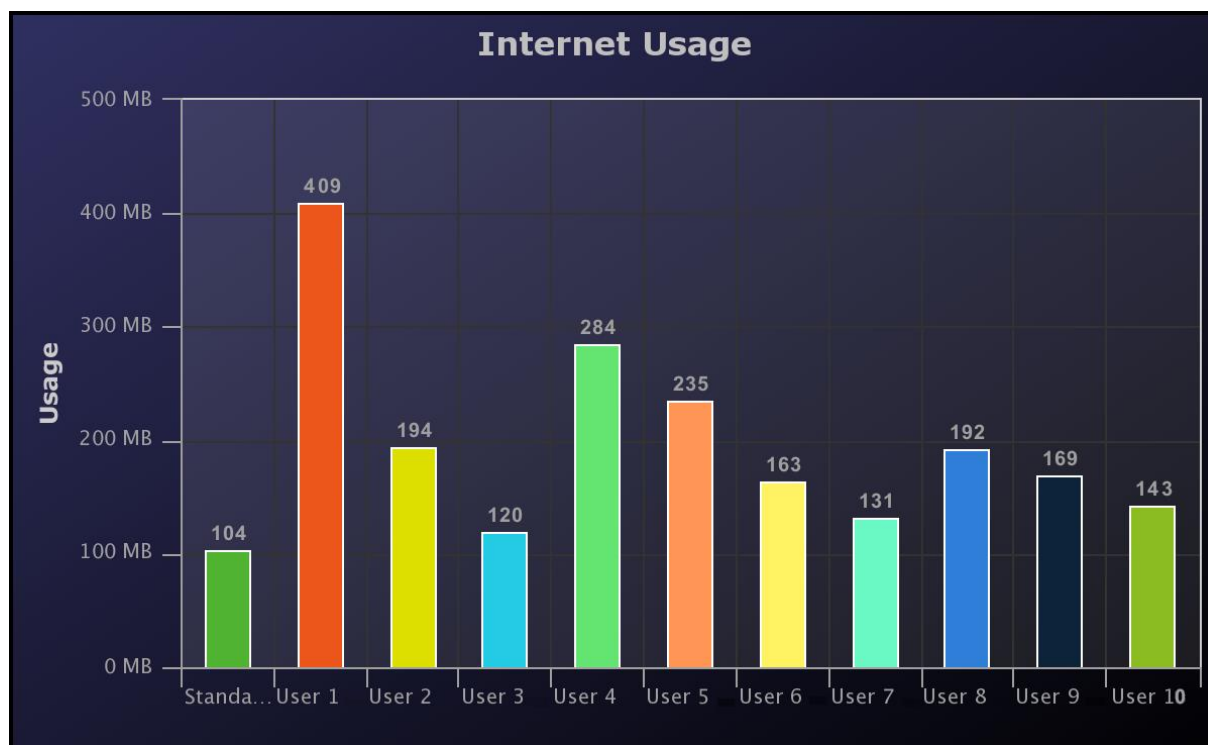


Fig.15. Internet Usage of Each User

Event Viewer was only used as a third detection line, its usage was a necessity to confirm if indeed there was an imposter intrusion or a false alarm through analysing the logs and the used files, all users had problems accessing the exact mentioned files in time, which increased the detection level of host-based detection, the use of Event Viewer helped reduce the false alarm rate to 0%.

The following chart shows the combined contribution percentage of the three detection systems (fig. 16):

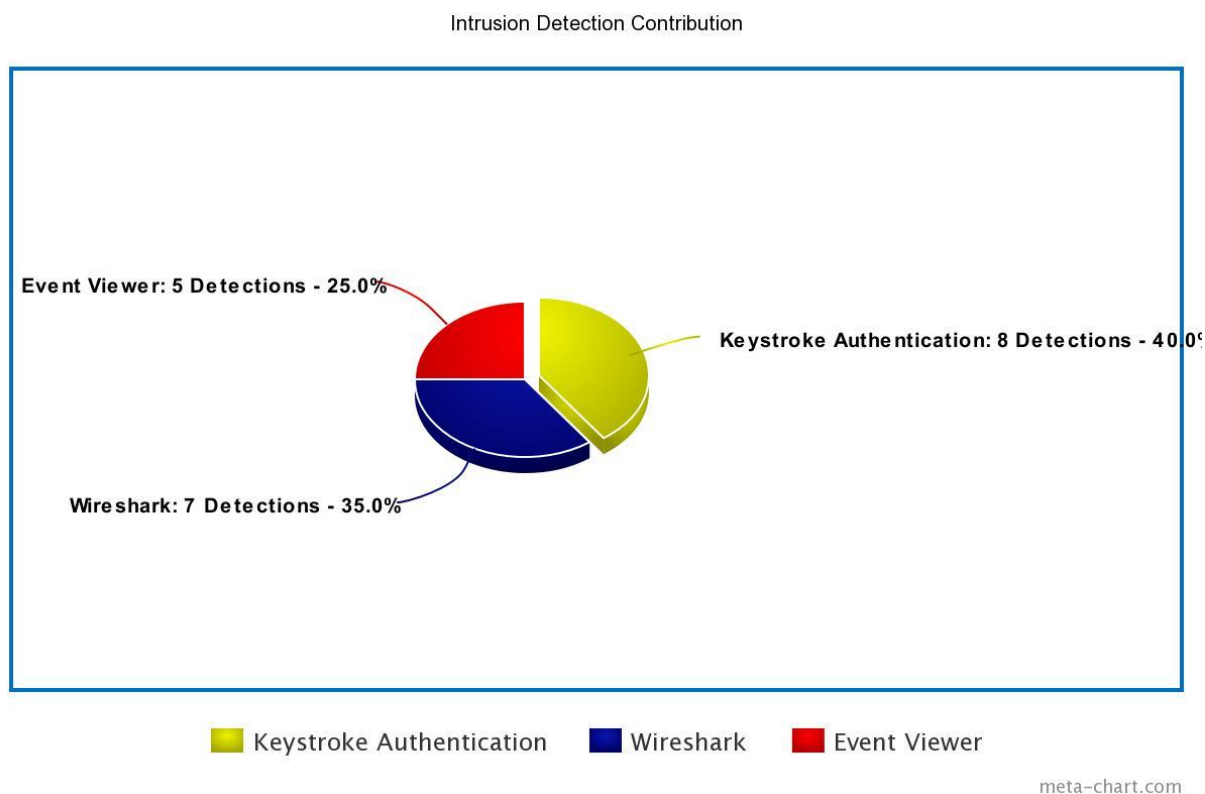


Fig.16. Intrusion Detection Systems Contribution Rate

The pie chart shows that the first detection line which is Keystroke Authentication has the most detection success rate with the most contribution with 40%, followed by the second detection line that is Wireshark with 35% and finally the third and last detection line which is Event Viewer with a contribution of 25% because of necessity, that's due to the detection of the intrusion before it is needed.

The proposed methodology can be very useful for forensics detection and investigation, especially since it extracts the data and can be customized to extract more information from Wireshark that can be decrypted using decryption tools and analyze what the users have been accessing on the internet.

Not only that but with the Keystroke Authentication, it increases the chances of finding the attacker, especially when the fingerprints recognition is not useful.

4. CONCLUSION

This paper discussed the different types of anomaly detection systems, their accuracy, false alarm rate and their vulnerabilities and compared between them, stated the GDPR regulations that needs to be implemented when using the Intrusion detection systems, and finally finished with a software-based methodology that were followed by experiments and analysis that effectively meet the objectives of this study which are improving the accuracy and success rate of imposters detection and reducing the false alarm rate.

Although this methodology turned out to be successful, there are still several open issues that need to be addressed, because only ten users cannot define the actual success rate for this software-based methodology, as for future work this methodology needs to be implemented as a machine learning device with an AI algorithm that connects a new and particular host-based detection with a new Keystroke Authentication developed with a special more accurate artificial intelligence algorithm, that detects imposters intrusion and send real-time alerts based on the data of the three detection lines with a below 5% false alarm rate without having to wait for the administrator to analyze it.

5. ACKNOWLEDGEMENT

This work was supported by Dr Hassan Chizari, Associate Professor in Cyber Security in the School of Computing and Engineering at the University of Gloucestershire, UK, and the IT Help Zone for providing the suitable tools and environment to help this project. Last but not least, we sincerely thank the reviewers of the manuscript for their comments and suggestions.

6. REFERENCES

- [1] Kemmerer R A, Vigna G. Intrusion detection: a brief history and overview. *Computer*, 2002. 35(4), supl27-supl30.
- [2] Li W. Using genetic algorithm for network intrusion detection. *Proceedings of the United States department of energy cyber security group*, 2004. 1, 1-8.
- [3] GrahamR .FAQ: Network intrusion detection systems. <http://www.robertgraham.com/pubs/network-intrusion-detection.html>. 2000.
- [4] Vaccaro H S. *Detection of anomalous computer session activity*. Retrieved from. 1988.

- [5] Tidjon L N, Frappier M, Mammar A. Intrusion Detection Systems: A Cross-Domain Overview. *IEEE Communications Surveys & Tutorials*, 2019. 21(4), 3639-3681.
- [6] Davies R M. Firewalls, Intrusion Detection Systems and Vulnerability Assessment: A Superior Conjunction? *Network security*, 2002. 9(1), 8-11.
- [7] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, (2009). 41(3), 1-58.
- [8]Chapelle O, Scholkopf B, Zien A. Semi-supervised learning (chapelle o. et al. eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 2009. 20(3), 542-542.
- [9]Jackson P. *Introduction to expert systems*: Addison-Wesley Longman Publishing Co., Inc. 1998.
- [10] Dutta S. *Knowledge processing and applied artificial intelligence*: Elsevier. 2014.
- [11]Russell S. ve Norvig, P. *Artificial Intelligence: A Modern Approach*. 1995.
- [12]Stone P, Veloso M. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 2000. 8(3), 345-383.

How to cite this article:

Tazerouti A, Ikram A. Imposters anomaly detection. *J. Fundam. Appl. Sci.*, 2021, 13(1), 243-263.