RIST

Information Processing at the Digital Age

VOL22. N.2(2018) 20-30

Resistance against DoS attacks in VANETs using the IDS Snort

Rahal Rabah[a,]*, Noudjoud Kahya[a], Nacira Ghaoulmi-Zine[a]

[a]*Department of computer science, Badji Mokhtar Annaba University, Annaba, Algeria.*

Submitted 29/12/2018, accepted 25/01/2019

Abstract

The Denial of Service (DoS) attack represents one of the most dangerous problems in the field of computer security. It purpose is to threaten the availability of a service or a system. Its application in a network like VANET (vehicular ad-hoc networks) could even cause fatal accidents. To combat this kind of vulnerability, we propose the use of an intrusion detection system (IDS) known as Snort. This IDS will allow us to detect it and fight it by applying a Tcp reset attack against the attacker, which is a computer attack that can be used for a beneficial reason. In this paper, we will implement Snort in a real vehicular network established at the University of Badji Mokhtar Annaba, Algeria. We will apply a DoS attack on a target vehicle. We will extract information about the throughput problems in order to create an image on the damages brought by the attacker on the target machine. We will study the effectiveness of Snort in fighting against the applied attack. Finally, we will study the impact of speed on Snort's effectiveness. The results shows the efficiency of Snort, were it decreased the rate of throughput problems caused by DoS with 51% at a speed of 20 km/h.

Keywords: VANet; Security; DoS; IDS; Snort; tcp-reset; Test-bed

1. Introduction

The vehicular ad-hoc networks (VANET) are a specific type of wireless ad-hoc networks used to establish a connection between vehicles. This connection can be ensured through three modes (Vehicle-to-vehicle, Vehicle-to-infrastructure and hybrid mode).

The objective of VANETs is to improve the experience of driving for drivers and passengers by offering them an application package. Applications divided into two groups, **safety** applications, like "Alert message broadcast, Intersection Collision Warning, Cooperative Collision Warning, Work Zone Warning and

Approaching Emergency Vehicle". **Comfort** applications, such as "Internet Access, Data Transfer, Parking Lot Payment and Traffic Information".

Benefiting of this kind of networks, will make our roads a safer place, but at the same time it opens the door to a set of computer attacks that can cause fatal accidents, like the Denial of Service (DoS). The DoS is an attack that aims to prevent the operation of a network in order to threaten its availability. It floods a network in order to interrupt access to the various services offered. It can take several forms according to the mode of its operation (like: Ping flood, Smurf, Fraggle, DDoS and Syn-flood), in this paper we will work on the Syn-flood model, which is the DoS attack based on a flaw in the connection establishment principle called three-way handshake. This operation takes place in three steps: The client sends a SYN message to the server to request the establishment of a connection. The server responds with a Syn-ack message (synchronize-acknowledgment). In a normal case, the client responds with an ACK message, and the connection is established. However, in a malicious situation, an attacker removes the last step, which causes the server to wait for a certain time before releasing the resources reserved for the client. By generating enough incomplete connections of this type, it is possible to overload server resources to cause a denial of service.

In order to fight against Syn-flood in VANETs, We propose the implementation of an Intrusion Detection System (IDS) known as Snort and which has never been implemented or studied in a context of vehicular networks. The goal of our implementation is to provide a second line of defense that serves to detect and combat DoS attacks. This fight is done by the application of a counter-attack on the attacker known as Tcp reset attack, which serves to interrupt all connections Tcp of the attacker.

The implementation will be done in a real vehicular ad-hoc network established at the University of Badji Mokhtar Annaba, Algeria. We will apply a DoS attack on a target vehicle. We will extract information about the throughput problems in order to create an image on the damages brought by the attacker on the target machine. We will study the effectiveness of Snort in resistance against the attack applied by extracting information about the rejected packets. Finally, we will study the impact of speed on Snort's effectiveness (at 10 km/h vs 20 km/h).

This paper will be divided as follows, Section 2 State of the Art, Section 3 focuses on the IDS Snort; Section 4 presents the laboratory preparation, Section 5 Results and Discussion, in addition, we conclude with a conclusion and perspectives.

2. State of the art

Many researchers have worked on the Snort Intrusion Detection System. Some of them in an analytical way, like Liao et al, 2013, which they proposed a comprehensive review on the methodologies used by the IDSs (the signature, anomaly and stateful protocol analysis), the approaches (Statistics, Pattern, Rule, State and Heuristic) and the types of technology (HIDS, NIDS, WIDS and NBA). They also presented a study on the two best-known open-source IDS (Snort and ClamAV) and the different solutions that was proposed by the researchers to improve their performances.

In 2013, a state of the art on intrusion detection systems in VANETs was done by Erritali and El Ouahidi. They proposed a classification of IDSs based on architectures (stand-alone, hierarchical or distributed) and based on detection techniques (anomaly detection system, Signature or Specifications based system). Whose Snort was cited as a stand-alone IDS based on the Signature. Moreover, it was the first IDS mentioned in its category that reflects its reputation and popularity.

Others worked on his performances, like In 2010, Salah and Kahtani have studied the performances of Snort under Linux and Windows 2003 server. The study was in terms of throughput and packet loss in different traffic load conditions (low and high, normal and malicious). Moreover, to improve snort performances, they also studied the packet loss at the kernel level and the impact of adjusting the key system

parameters for Linux and Windows. In particular, for Linux, the impact of choosing different NAPI budget values and for Windows, the impact of configuring processor Scheduling option. Experimentation has shown that changing the Windows processor Scheduling option does not have a real impact on Snort's performances. Unlike the value of NAPI budget values under Lunix, which they have shown that a small value (2 for example) can significantly improve the performances of Snort.
In 2018, Shah and Issac compared the performances of two IDSs (SNORT and Suricata) at a network speed of 10 Gbps. They found that Suricata could handle a higher speed of network traffic than Snort with a lower packet drop rate, but that it consumed higher computing resources. In contrast to Snort, which showed higher detection accuracy but also, triggered a high rate of false positive alarms. For this reason, they proposed an adaptive plug-in for Snort and they did an empirical study for the selection of the most efficient algorithm. They found that the best result (in terms of false positive rate and false negative rate) was achieved by using an SVM optimized with the firefly algorithm.

Others worked on its effectiveness against DoS attacks, such as Chakrabarti et al, in 2010. Where they made a comparative study between Snort and EagleX in terms of efficiency against DoS and scanning attacks (SYN stealth, enumeration and TCP scans). The results of simulation showed the higher efficiency of Snort compared to EagleX. However, in this study, the results was presented using Log files, which make them unclear and hard to compare between them. Moreover, a lot of other research has been done too.
However, it has **never** been **studied** in the context of vehicular ad-hoc networks.

3. Snort

Snort is an Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) at the same time. It was developed in 1998 by Martin Roesch (Sazzadul, 2010). It makes it possible to identify abnormal or suspicious activities on the analyzed target (network or host). It allows to know about the attempts of intrusions successful or failed. It is classified as a signature-based detection system (The system has a database behavior of certain attacks with which the collected data are compared). Moreover, its architecture is known as Stand-alone which means that when it is used, it relies only on its local resources and does not exchange information with other network nodes (Erritali and El Ouahidi, 2013) .

3.1. The Operating Principle of Snort

Snort is executed in four steps:

1. The packet decoder: The first process executed when running Snort is called the packet decoder. Its purpose is to make the future use of package content easier. It receives the raw packets, then it starts three decoding operation that runs on three different layers of the TCP/IP model.
2. Pre-processors: In this step, two types of processors will be executed. The first is called the Packet Examiner, who deals only with the examination of packages. The second (packet modifier) aims to put the packets previously treated in the best possible conditions for the next step.
3. The detection engine: This is the most critical and important phase, because it takes care of the detection of the anomalies one is based on a set of rules. The rules are loaded at the launch of the software to build a decision tree used to classify the captured stream.
4. The output plugin: This is the last step of Snort's execution, it serves to inform the user when an anomaly is detected. The malicious behavior can be logger and / or generate as an alert.

3.2. The Operating Principle of Snort rules

Writing snort rules is done using a simple and powerful description language. These rules are composed of two parts, the first one is the header of the rule that determines the action (alert, log, pass, activate, dynamic, drop, reject and sdrop), the protocol (tcp, udp, ...), source and destination IP addresses, network masks and ports. The second part that represents the options of the rule like the messages of alert.

4. Experimentation

In this paper, the experimentation was done in a real vehicular network at the University of Badji Mokhtar Annaba, Algeria. Its goals are:

- The extraction of concrete information about the impact of DoS attack in a vehicular network.
- Obtaining an idea on the relationship between the speed and the impact of the attack on the target.
- Drawing out information about its impact on the target machine in terms of throughput problems.
- Study the effectiveness of the snort against DoS.
- Verify its adaptation in a mobile nodes.

For the establishment of such a network, we used the following equipment's:

Hacker's side:

- Cisco AIR-ANT1949 Antenna installed on the roof of the attacking vehicle (Fig.1.a).
- An AIR-AP1231G-E-K9 access point (Fig.1.b).
- A Dell computer with an Intel (R) Core (TM) CPU i7-3517U CPU @ 1.90GHz 2.40 GHz.
- Kali Lunix operating system.

Target machine side:

- A Sharkee GPSB antenna installed on the roof of the target vehicle (Fig.1.c).
- An AIR-AP1231G-E-K9 access point (Fig.1.b).
- A Dell computer with an Intel (R) Pentium (R) 3558U @ 1.70GHz 1.70GHz processor.
- Windows 8 operating system.

The network was deployed in the 500 meter driving area shown in Fig.2, located at Badji Mokhtar Annaba University, Algeria. A Cisco AIR-ANT1949 antenna is installed on the roof of the hacker's vehicle, which is connected to an AIR-AP1231G-E-K9 AP and a Dell i7 computer running Kali Lunix as the operating system. While driving along the entire path shown in Figure 2, the pirate's vehicle follows a target vehicle. The target vehicle is equipped with a sharkee GPSB antenna connected to an air-ap1231g-e-k9 access point and a Dell computer equipped with an Intel (R) Pentium (R) 3558U processor. On this computer, a Snort intrusion detection system is installed. Both vehicles (Fig.1.d) are driven with the same speed and the same distance between them (10 meters). The experimentation will be made under 4 conditions:

1. At a speed of 10 Km/h and the target machine does not use Snort.
2. At a speed of 10 Km/h and the target machine uses Snort
3. At a speed of 20 Km/h and the target machine does not use Snort.
4. At a speed of 20 Km/h and the target machine uses Snort.

From 1 and 2, we will extract information about the effectiveness of Snort.

From 1 and 3 we will study the impact of the speed on the attack and the damages brought on the target machine.

From 2 and 4 we will talk about the effectiveness of Snort during the increase of the speed.

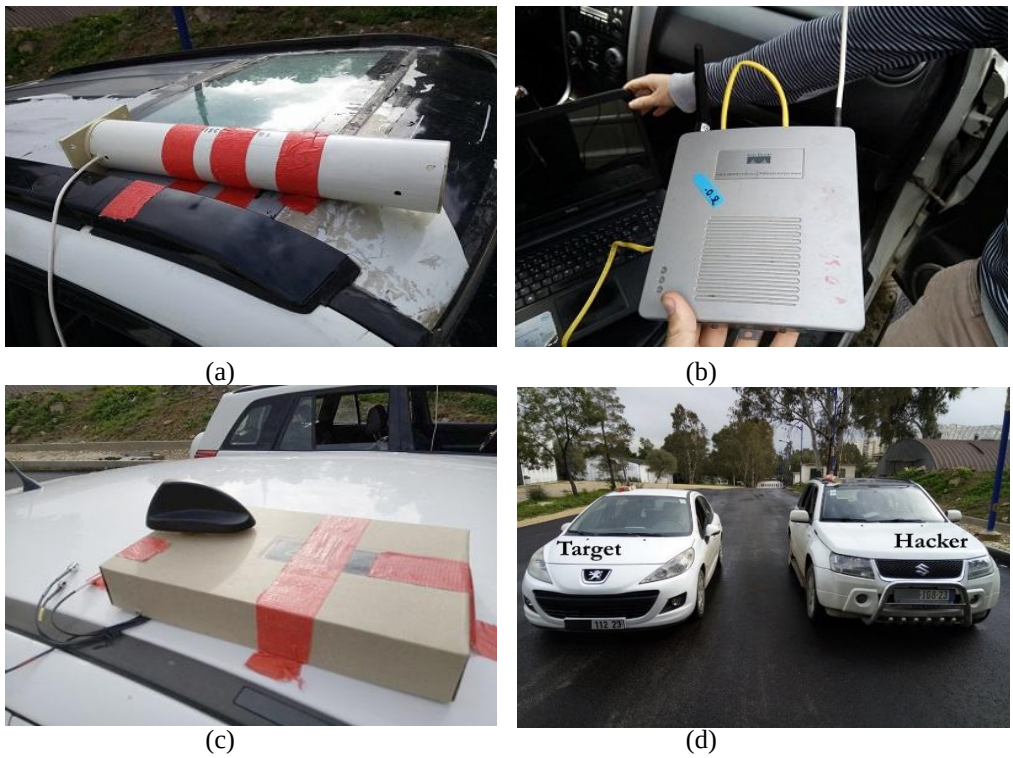

Fig. 1. Equipment used in the testbed

Fig. 2. The area of the deployment of the testbed at UBMA, Algeria

5. Results And Discussion

The results of the application will be divided into two axes, before the use of the Snort and during its use.

5.1. Without Snort

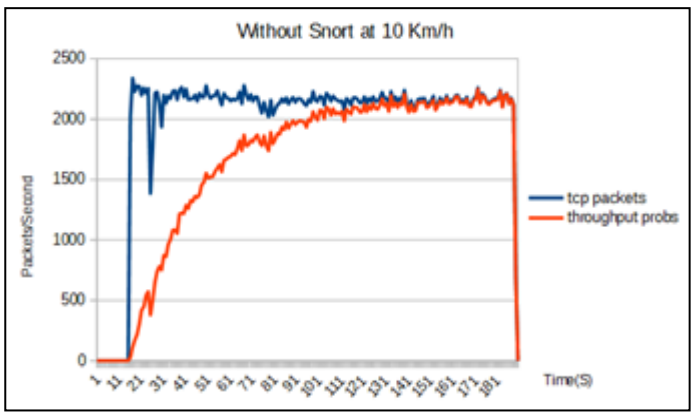

Fig. 3. The impact of the DoS attack on the target machine at 10 Km/h

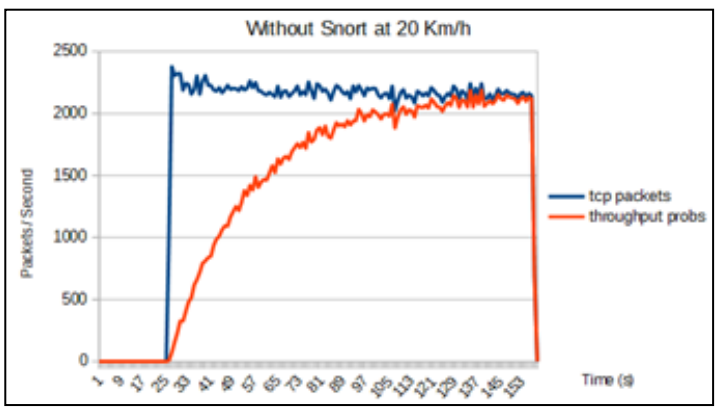

Fig. 4. The impact of the DoS attack on the target machine at 20 Km/h

According to figures Fig.3 and Fig.4, we can clearly see that during our application of a DoS attack in a vehicular environment, the number of tcp packets sent by the attacker arrived at 2000 to 2500 packets per second. This caused the appearance of throughput problems (retransmissions, zero window packets, duplicate acknowledgments, etc.) that appeared in the two figures mentioned earlier with the red line. It can be clearly seen that after 130 to 140 seconds, the system becomes totally unable to receive Tcp traffic, which means the total effectiveness of the attack in the two speeds used in our experimentation.

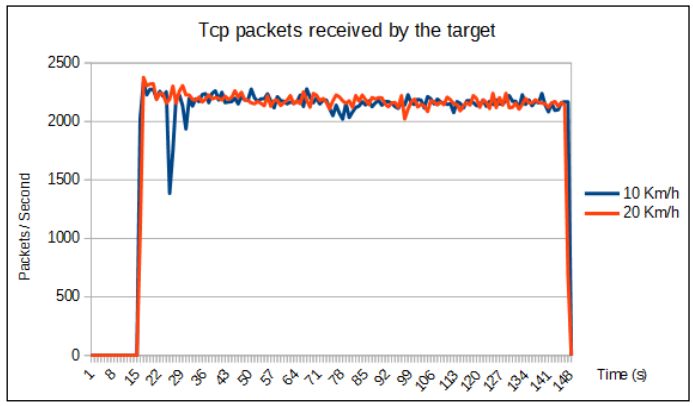

Fig. 5. The tcp packets sended by the attacker to the target machine at 10 Km/h vs 20 Km/h

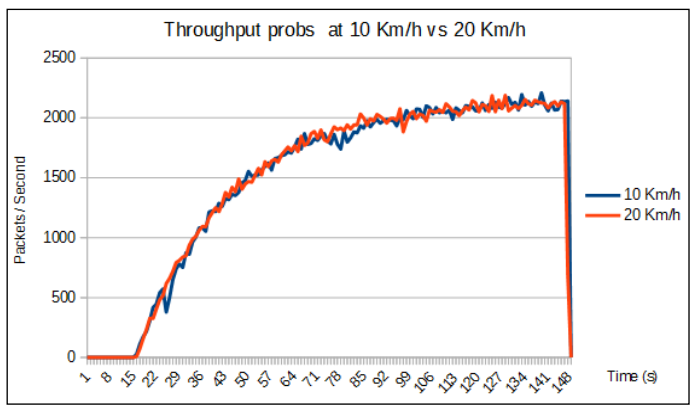

Fig. 6. The throughput probs on the target machine at 10 Km/h and 20 Km/h.

Fig.5 that illustrates the difference between the number of packets received by the target machine in the two speeds 10 Km/h and 20 Km/h, we can clearly see that the number was almost the same in both speeds, with a slight reduction of 0.06% during the increase in speed. Moreover, Fig.6 illustrates the throughput problems brought on the target machine in both speed 10 Km/h and 20 Km/h, which we can clearly see that the number was the same with a small increase of 0.11% at 20 km / h.

5.2. With Snort

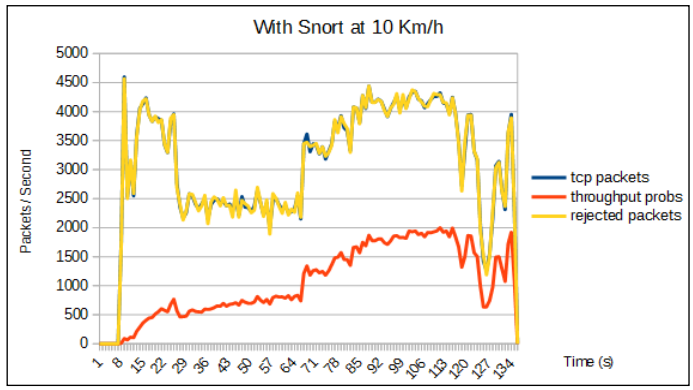

Fig. 7. The effectiveness of Snort at 10 Km/h

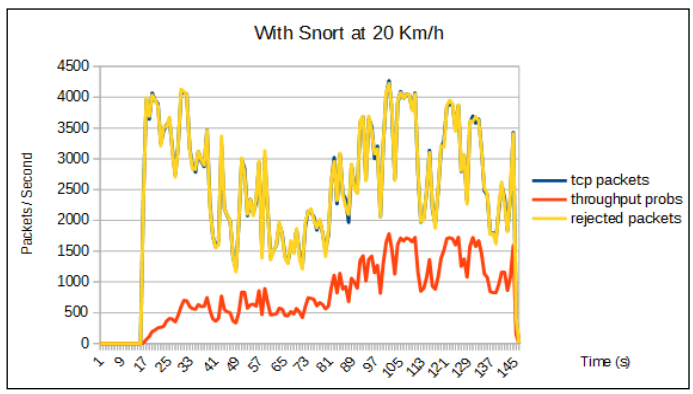

Fig. 8. The effectiveness of Snort at 20 Km/h

According to the two figures Fig.7 and Fig.8, we see the efficiency of snort. This efficiency is represented by the number of packets tcp reset (rejected packets with the yellow color), which are applied by the Snort to cut off the attacker's connection when detecting a DoS attack. We see that the rate of packets tcp reset was almost the same as the packets tcp sent by the attacker. This significantly reduced the rate of throughput problems compared to the case in which the Snort was disabled.

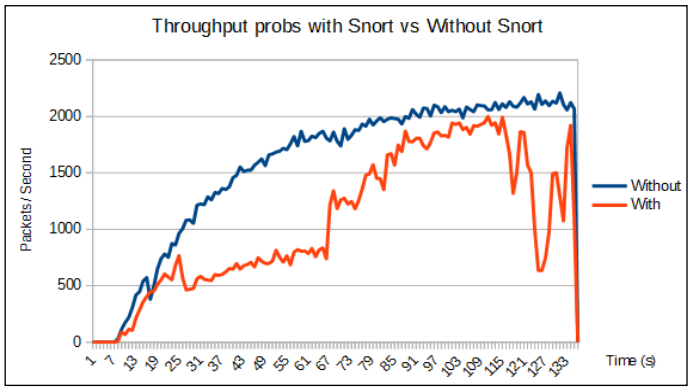

Fig. 9. Throughput probs with and without Snort at 20 Km/h

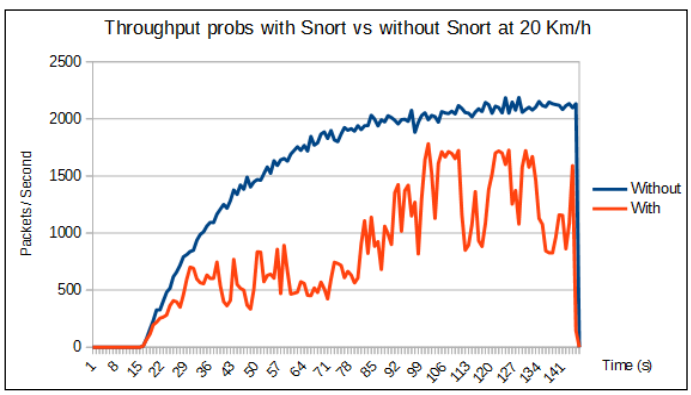

Fig. 10. Throughput probs with and without Snort at 20 Km/h

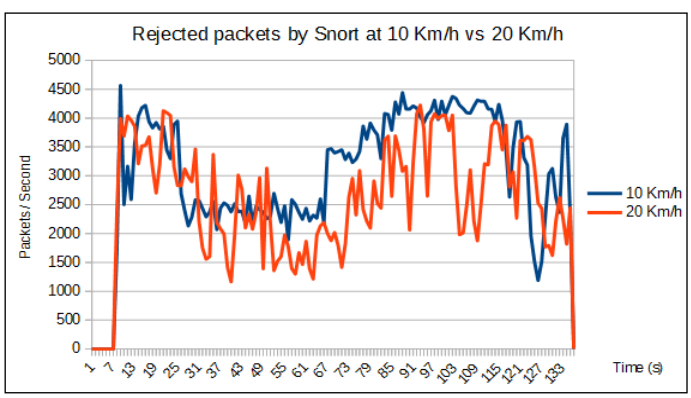

Fig. 11. The impact of the mobility on Snort's performances

This difference in the rate of the throughput probs has been well shown in the two figures Fig.9 and Fig.10, of which they show the rate of the throughput probs before the activation of Snort and after its activation at a speed of 10 Km/h and 20 Km/h successively. In Fig.9 at a speed of 10 Km/h, the rate of throughput probs was reduced by 47% upon activation of Snort. Moreover, it was reduced by 51% at a speed of 20 Km/h according to Fig.10. This reflects its effectiveness against DoS attacks. Fig.11 shows the impact of mobility on Snort performances, which we found that during the speed increase from 10 Km/h to 20 Km/h, the rate of packets rejected (tcp reset packets) was decreased by 18%, but it has seen from the above figures that it is still effective.

5.3. Overall Impact of Snort

We found that a simple DoS attack in a vehicular ad-hoc network with 2,000 to 2,500 packets sent per second could cause real throughput problems after 130 to 140 seconds, whose system will be unable to handle tcp traffic. Moreover, in a vehicular context the speed has an impact on the power of the attack. We found that during the increase of the speed from 10 Km/h to 20 Km/h the numbers of the packets sent was reduced by 0.06% but the rate of throughput problems has been increased by 0.11% (Other throughput problems have appeared because of the increase in speed). When applying Snort, we can clearly see its efficiency of detection and fight against DoS, whose number of packets interrupted (tcp reset) was almost the same as that of the packets received. In addition, the rate of throughput problems was significantly reduced by 51% at a speed of 20 km/h. Regarding the adaptation of Snort to the mobile context. According to our experimentation, we found a reduction of 18% during the increase of the speed of 10 Km/h to 20 Km/h, but its efficiency of interruption was the same. Overall, the Snort represents a reliable and free solution that can be used in vehicular networks to fight DoS attacks.

6. Conclusion And Perspectives

Denial of service attack is one of the most dangerous attacks; it targets the availability of systems. Its application in a vehicular network can cause human losses. In this paper, we propose the use of an intrusion detection system labeled Snort as a countermeasure. The application was made in a real vehicular network at Badji Mokhtar University Annaba, Algeria. Moreover, the results showed its effectiveness of which it has decrease the rates of throughput problems with 51% at a speed of 20 Km/h. In addition, it has shown an adaptation to the mobile context. In our future work, we will carry out a simulation for the proposed contribution, but with larger VANET, many flows, and various trajectories and speeds, in order to get a more convincing conclusion regarding the performance of Snort in VANETs.

7. References

Liao H-J, Richard Lin C-H, Lin Y-C, Tung K-Y. Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications 2013;36:16–24. doi:10.1016/j.jnca.2012.09.004.

Erritali M., El Ouahidi B. A Survey on VANET Intrusion Detection Systems. International Journal of Engineering and Technology 2013;5:1985-1989.

Salah K, Kahtani A. Performance evaluation comparison of Snort NIDS under Linux and Windows Server. Journal of Network and Computer Applications 2010;33:6–15. doi:10.1016/j.jnca.2009.07.005.

Shah SAR, Issac B. Performance comparison of intrusion detection systems and application of machine learning to Snort system. Future Generation Computer Systems 2018;80:157–70. doi:10.1016/j.future.2017.10.016.

Chakrabarti S, Chakraborty M, Mukhopadhyay I. Study of snort-based IDS. Proceedings of the International Conference and Workshop on Emerging Trends in Technology - ICWET '10, ACM Press; 2010. doi:10.1145/1741906.1741914.

Sazzadul Hoque M. An Implementation of Intrusion Detection System Using Genetic Algorithm. International Journal of Network Security & Its Applications 2012;4:109–20. doi:10.5121/ijnsa.2012.4208.