Journal of Theoretical and Applied Information Technology
15th September 2018. Vol.96. No 17

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A FUZZY-BASED BUFFER SPLIT ALGORITHM FOR BUFFER ATTACK DETECTION IN INTERNET OF THINGS

[1]MUSHTAQ MOHAMMED ABDULNABI, [2]ROSILAH HASSAN, [3]NOR EFFENDY OTHMAN, [4]AZIZAH YA'ACOB

[1,2,3]Research Centre for Software Technology and Management (SOFTAM),
Faculty of Information Science and Technology
[4]Pusat CITRA Universiti
Univesiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, MALAYSIA
E-mail: [1]mushtaqalani2@gmail.com, [2]rosilah@ukm.edu.my, [3]effendy@ukm.edu.my, [4]aziey@ukm.edu.my

ABSTRACT

Internet of Things (IoT) process technology is a rapidly emerging technology with great benefits for the pool of individuals, homes, companies and other institutions in general. This technology is used to connect everyday devices to the network to enable additional features such as remote control and access to data, But the network is exposed to several attacks, the most important attacks that cause the problem of overload is event-based attack, Smart stretched attack on the buffer one of the most important attacks on the network, in this research we propose a new algorithm called a Fuzzy Buffer Split (FBS) to defend against these two types of buffer attacks.

Keywords: *IoT, Buffer Reservation Attacks, Overflow, Fuzzy System, Algorithm*

1. INTRODUCTION

Every year there are many new emerging terms, most of which are related to the modern world and technology. In the same way, the term "Internet objects" has recently emerged, which means the new generation of Internet or network that gives the possibility of understanding between interconnected devices through Internet Protocol. These devices include instruments, sensors, and various artificial intelligence devices. Internet of things (IoT) the individual freedom of movement and freedom from the place, that is, the person can control the means without the need to be in a certain place to control a particular device [1]. IoT is basically connected to objects, sensors, other triggers, and many other smart technologies, enabling connections from person and object to object [2]. Internet object is an integral part of the evolving future Internet and can be defined as a dynamic global network infrastructure with high self-configuration based on different protocols. They are interoperable as physical and virtual objects with different identities, physical properties, and virtual characters are innovative and seamlessly integrated into the information network [3]. According to traditional information companies including the telecommunications network, Process techniques that connect ordinary physical objects with specific addresses that provide intelligent services [4], are the latest update of the Internet that has brought about a new change in the world of information and communications technology. Represents a Thread point at a time when most of the things connected to the Internet are with each other with people anytime and anywhere you can select a lot of unique things anywhere. [5], Objects can connect devices to each other to make the digital ecosystem as well as be connected to the Internet [6], as shown in Figure 1.

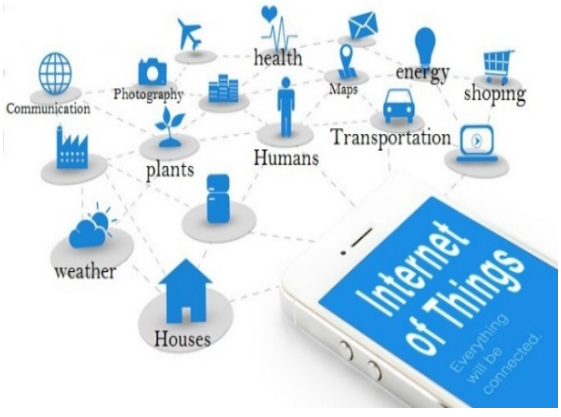

Figure 1: Interconnection of IoT

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Despite the continuous development and updates on the Internet, it is not free from vulnerabilities that may be small but have a direct and severe impact on the network. Such as malicious attacks that attempt to sabotage, break the network, and steal information [7]. There are a lot of malicious attacks on the networks of various kinds, which is the form of legitimate packages enter the system, including legitimate and illegal, harmful, which exploits large areas in the buffer and thus lead to the occurrence of so-called overload. Researchers have studied many solutions to stop this problem and prevent them from entering but most studies were unable to detect [8]. In this paper the goal is to detect malicious attacks that enter the network and work on the impossibility of the largest possible area of the store and the introduction of illegal packets and penetration of the system and thus occur overload.

1.1 Buffer Attack

Is a buffer that collects packets entering the network called a fragment, the buffer acts on the order of incoming fragments in order of arrival time and processing them to enter the next Stage. Attacks that enter the network through the buffer because it is the primary recipient of all packets from other nodes, [9] Therefore the store is exposed to several attacks that work to occupy the largest possible area of the store and thus the occurrence of overload. In this paper we used two types of attacks that often enter the network that lead to damage by reaching the excess load,[10] and are considered to be the most dangerous attacks that are difficult to detect. These attacks are called (event-based attack, Smart stretched short burst attack).

1.2 Buffer Attack In IoT

Buffer overflow and vulnerabilities are caused by many applications and unchecked of the identified by the availability of space before copying unreliable data in a predefined space in the system memory,[11] overwriting the contents of memory out of the buffer. The program looks at the memory space it sees the current data of the bypass instead of the original data. If the program tries to use values from that region [12] it is likely that the program is unable to see the expected consequences and can range from program crashes and other more dangerous procedures such as DOS or worse, and the implementation of new malicious code that is planted by the user [13]. Buffer-based buffer overrun can allow attackers to execute code on a recipient's computer, where it replaces memory addresses that will be used at a later time, while "stack overrun" usually results in DOS, where it tries to write to memory that is available. [14]. Hence when this the timeout for the reassembly node expires, must drop the non-integrated packet from the buffer and reassemble for freeing the memory for other new fragmented packages, to mount an attack that stores the buffer; the attacker generates one fragmentation with the arbitrary load and sends it towards its target. If the buffer does not contain the target after it is by another fragmented package, fragmentation retains that buffer to reassemble the fragmented packet of the attacker, The attacker Now either do not send the remaining vacuums or release them sporadically in order to occupy the buffer resources until the time-out period for the re-assembly has ended. During this time, additional packages cannot be segmented by the target node [15].The node is targeted by several different attacks, sometimes attacks from neighboring nodes or from an external network are sending malicious attacks that attempt to break the safety and infiltration into the network. As mentioned above, there are different types of attacks, such as those that enter the network in a legitimate package so that they are not detected, and others that send multiple attacks at different times to try to put pressure on the network and break the protection.

1.3 Split Buffer Approach

The benchmark proposes mechanisms to increase the costs for an attacker such that it has to continuously send complete fragmented packets in short bursts in order to prevent legitimate packets from being processed at the target node. In this case, the buffer reservation attack resembles flooding attack. If a node stored individual fragments of multiple packets in its reassembly buffer, legitimate and malicious packets would compete for the available buffer resources based on the actually used buffer space. An attacker would then have to follow up on its pretense by transmitting further fragments. To enable direct competition for the buffer resources between legitimate nodes and an attacker [16] the benchmark proposes to split the reassembly buffer into fragment-sized buffer slots. Each slot has the maximum size of a fragment for a given link layer. Buffer slots are filled until either a packet has been fully received or an overload situation is reached. In case of a complete packet, the reassembling node assembles the packet in-order in the buffer and processes the packet normally. In case of a buffer

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

overload situation, the node has to decide which packet to discard. To this end, the reassembling node can base its decision on the observed sending behavior for packets located in the split buffer.

2. BASELINE APPROACH

The baseline approach (Hummen et al., 2013) proposes a discard strategy for packets in the split buffer that is based on per-packet scores, capturing the extent to which a packet is completed along with the continuity in the sending behavior. In case of a buffer overload situation, the node then discards the packet with the lowest score. If two or more packets share the lowest score, the selection is performed randomly between these packets. The baseline approach identified three fundamentally different sending behaviors (attacks) that an attacker may show during the buffer reservation attack:

- FRAG1 attack.
- Short burst attack.
- Stretched attack.

And it was able to detect and avoid them using the following score function:

$$score_{i+1} = \begin{cases} \frac{\text{fragment bytes}}{\text{total bytes}} & \text{if } i = 0 \\ score_i + \frac{\text{fragment bytes}}{\text{total bytes}} & \text{if } a - w < l < a + w \\ \frac{score_i}{2^{\max\{1;\lfloor l/a \rfloor\}}} & \text{else} \end{cases}$$

Where **i** is the elapsed time since the last fragment, **a** is average elapsed time between two consecutive fragments of a packet and **W** is a window around expected value **a**.

3. PROBLEM STATEMENT

There are two types of attack the benchmark isn't able to detect them:

i. Event-based attack
ii. Smart stretched short burst attack

First attack is event-based attack; the malicious node will send its first fragment and wait until then sending the first fragment from a legitimate node.
Here, the malicious node will send its second fragment and wait for the next fragment from the legitimate node to send again [17].
This process will be repeated until an overload situation is reached, there are two possible cases to drop a packet in such situation, the first one happens if the legitimate node has sent its fragment and then the overload occurs. Second attack Smart stretched short burst attack In this attack, the malicious node will send almost but a few fragments in a short burst in order to occupy as many buffer resources at the target node as possible and to get a high score faster, after sending these fragments the malicious node will send each of the remaining fragments at the largest allowed time value within the scope of the window w, in this way the attacker appears like a legitimate node and its packet remains as long as possible in the target's buffer since it has a high percentage of completion and its sending behavior is legal.

4. RELATED WORKS

Internet usage and internet addiction have been studied by many researchers with respect to various perspectives like positive, negative, geographic, social, professional and health and others [18]. The attacks previously identified on the detailed security analysis of the IPv6 Low-power Wireless Personal Networks (6LoWPAN) are based on a fragmentation mechanism where two types of attack have been identified at the specified level that enable the attacker to prevent legitimate packet reassembly and reassemble once again at the target node and at low cost [19]. The magma process can load the attacks identified above by sending one part that is compatible with the correct one. To control these attacks (event-based attack,[20] Smart stretched short burst attack) they proposed defense mechanisms and the Boer content sequence system,[21] but this mechanism enables the right nodes to acquire half the success of 50% so the attacker can exploit at least half and therefore the process of overload [22].

Over the years that have elapsed, buffer overflows has been one of the most common, widespread, and most dangerous loopholes. Although many solutions and high techniques have been proposed to address this problem, they have found loopholes [23] at the same time which, Identify attacks, provide detailed signs and effects to be identified, but this takes time and requires additional resources. So they proposed a solution to control this problem, which integrates the detection of the attack and the generation of a mechanism for defense and abuse which is called Heap Therapy. Which works during the implementation of the program collects the impact of lightweight packages and starts the diagnosis in order to

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

generate a defense in real time. But the regulation of the expected time between sending the package and the other package is not controlled so there is a possibility of entry of the attacker [24]. Stack overruns still pose a security risk and cause multiple errors to be repeated. The embedded systems can be attacked very easily by the proposed heap attacks and based on the embedded security processor's embedded security techniques, the hardware defense mechanism within the network of the Heap Defender, designed to detect the resulting attacks from the excess stack, The unit of these devices located inside the embedded processor, does not affect the program and does not disrupt the work and integrity of the pipeline [25].

The process of synchronizing all instructions and analyzing them in parallel within the Heap Defender is parallel with the central processing pipeline, resulting in the Heap Defender having very little overall performance. Despite the power of the proposed mechanism, however, there are types of attacks that may penetrate this mechanism and the system as legitimate and correct packages and intervention to occupy the large space within the system [26]. The complexities of software systems are beginning to take on an increasing, increasing number of bugs. Many of these errors pose significant security vulnerabilities. The most common of these errors is double buffer overrun. A test process implemented through 20 different attacks exceeded the buffer and used it to compare four publicly available tools to prevent dynamic infiltration and to stop all buffer overruns [27]. Then the process of comparing the tools was carried out empirically and theoretical, It is the most effective tool against 50% of the various attacks. There are six types of attacks that none of the tools can handle. However, there are various sites and situations that allow more than supposed attacks to enter randomly, leading to the accumulation of sent packets and the formation of malicious attacks [28].

5. PROPOSED SOLUTION

Internet networks and systems used in them need several defenses against attacks. We have also mentioned that there are many attacks that enter the network and spy on them and steal their information, leading to the destruction of confidential data. In this paper, we have highlighted two types of network attacks that enter the buffer, represented by the problem of legitimate packages that are officially inserted into the network. Where it works to reduce the largest possible area of the store and exploit for the benefit of harmful packages and concentration until the transition to the second stage and enter the system. The presence of these harmful packets leads to the occurrence of overload, which is a problem experienced by many systems and networks. Much of the previous research has been done to get rid of them but it is still undetectable. In this paper a new algorithm has been proposed to detect these attacks and prevent them from accessing the network. The FBS algorithm is a new algorithm that detects attacks based on mathematical rules that you execute on packets entering the buffer before entering the network.

5.1 Fuzzy Based Buffer Split- FBS

The problem of overload due to malicious attacks on the network and studying previous proposals, we proposed a new algorithm that detects harmful attacks and controls the prevention of overload is the FBS algorithm. We detect attacks through score, and also calculate the arrival time and the expected time between sending the packet and the other packet and then deleting packets that have a low score and drop it, and send the Legitimate to buffer mange. The proposed algorithm works to prevent attacks that enter the network that provides this secret system to ensure the complete transfer of information and the safety of access. This system is designed and aimed at integrated and effective networks in the online environment and the implementation of transport and communication process with the Internet of things. The benchmark proposes mechanisms to increase the costs for an attacker such that it has to continuously send complete fragmented packets in short bursts in order to prevent legitimate packets from being processed at the target node. In this case, the buffer reservation attack resembles a flooding attack. There is also a lot of systems that have already been implemented for this purpose, but has not been proven highly efficient, as is the case in the system FBS. The paper proposed two types of scenarios that are event based attack and attack smart stretched burst. Figure 2, shows a conceptual diagram of the developed system. Firstly, the buffer and the clock are used as input to a block called input-variable preparation, which is supposed to calculate the input variables of the FBS. The output of the FBS (the score) is fed into a block called buffer manger. The buffer manager is responsible to drop the fragment of the lowest score from the buffer. This process occurs within each node in the network.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

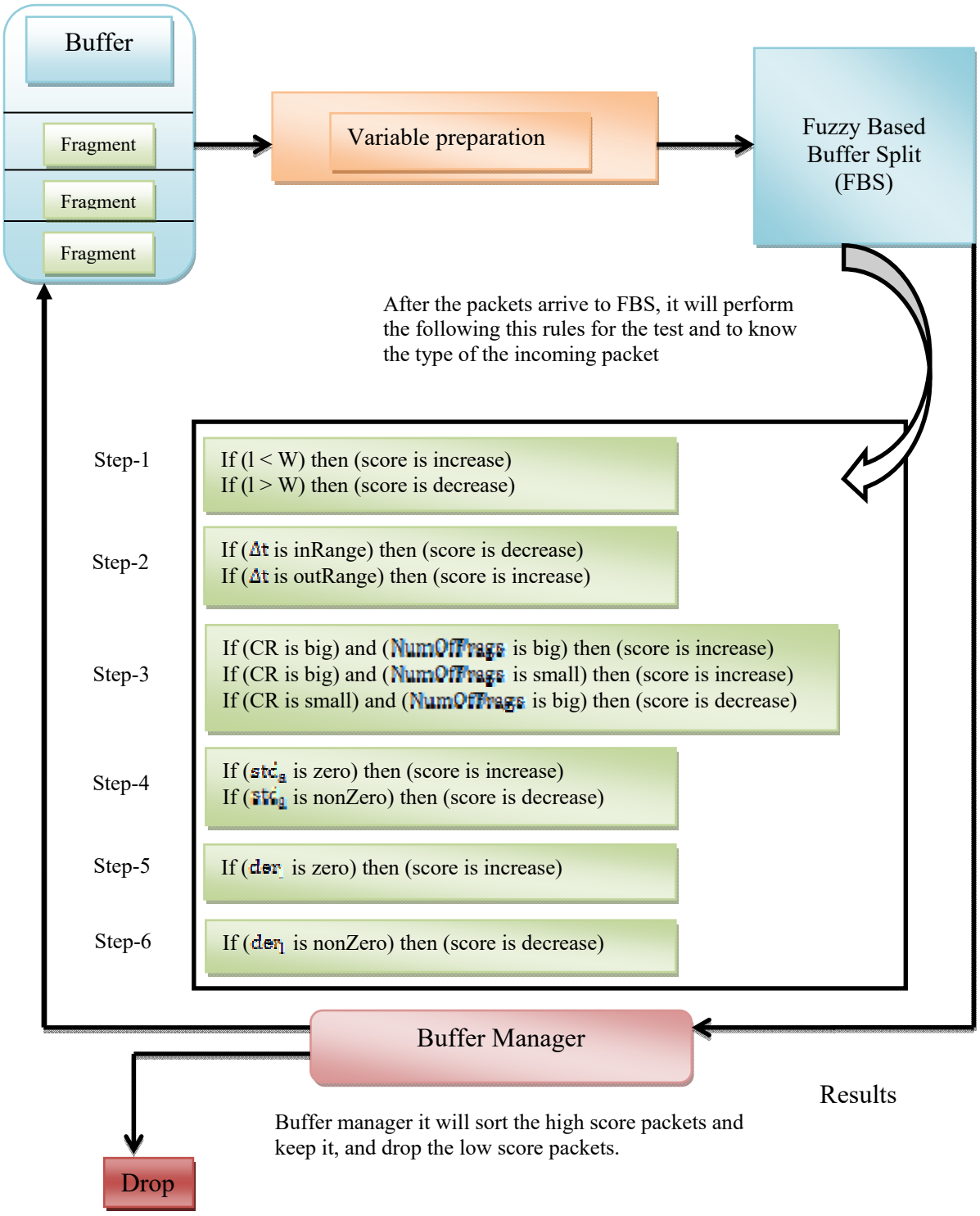

Figure 2: Overview for FBS Design

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5.2 Fuzzy Sugeno

The developed mathematical tool to build fuzzy model of **a** system, they assumed that the membership function set **A** as **A(x), x ∈X**. The membership function is characterized by two parameters providing a range of the membership functions values between zero and one. (Takagi & Sugeno, 1985) The fuzzy interpretation of the proposition: **x** is **(a)** and y is **(B)** is **A(x) ^ B(y)**. The particular modification of Sugeno type of fuzzy on the original Mamdani fuzzy is the format of the implications **R** in the form of **a** mathematical function as:

If (f(x_1 is A_1 and x_2 is A_2.. and x_n is A_n)then y=g(x_1,x_2,…x_n)

6. IMPROVEMENT

In order to detect the previous two attacks, some parameters have been taken into consideration and these parameters are as follow:

1- Δt The association time with the latest fragment entered the target's buffer (the time difference between the receiving time of the current fragment and the receiving time of the latest fragment entered the target's buffer).
2- CR: The completeness ratio of packet.
3- $NumOfFrags$: The number of the fragment in the packet.
4- std_a: The standard deviation of **a** s values.
5- der_l: The absolute value of derivative of **i**.

The previous five parameters beside **i** were used as inputs of Sugeno fuzzy inference system to compute the score of a packet.
The output of the system takes two constants as membership functions: increase and decrease. The first constant (increase) is equal to:

$$old\ score + {}^{1}/_{number\ of\ fragments}$$

And the second constant (decrease) is equal to:

$$\frac{old\ score}{2^{max(floor(\frac{1}{a}),1)}}$$

6.1 Numerical Example Explains The First Attack

If a malicious node has sent its FRAG1 in the time unit 40, and there is a legitimate node is sending its FRAG1 in the time unit 50, the malicious node then will send its FRAG2 in the time unit 51 after 1 time unit of sending the first legitimate fragment, the malicious node will repeatedly send its FRAGN after sending the legitimate FRAGN, so the Δt value (the association time value) will be too small because the sending time is too close and that Indicates to event-based attack and the benchmark will not be able to detect it, because the l values are legal.

6.2 Numerical Example Explains The Second Attack

i. Legitimate packet:
Let's assume that a packet consists of 8 fragments and its fragments arrived in these time units:

50	60	70	80	90	100	110	120

This packet will be treated as a legitimate packet by the benchmark, because the value of l for each fragment is within the correct range (w=25). And also, it will be treated as a legitimate packet by the fuzzy scoring system because the values of ⟦std⟧ _a and ⟦der⟧ _l are zeros.

ISSN: 1992-8645 E-ISSN: 1817-3195

TABLE 2: THE VALUE OF LEGITIMATE

Variable	Value
(a) Time between sending a packet and the other	10, 10, 10, 10, 10, 10, 10
(I) Real arrival time of packet	10, 10, 10, 10, 10, 10, 10
(std) Denotes the standard deviation of α's values, mean	0, 0, 0, 0, 0
(der) Denotes the absolute value of derivative of l	0

For a random variable vector A made up of N scalar observations, the standard deviation is defined as:

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}|A_i - \mu|^2}$$

Where μ is the mean of A:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} A_i$$

ii. Malicious Packet

The second type is malicious attacks that threaten the security of the network and exploit the largest possible area of the network, and are irregular and have random time as shown in the example:

Example:

Let's assume that a packet consists of 8 fragments and its fragments arrived in these time units:

1	2	3	29	63	104	150	200

This packet will be treated as a legitimate packet by the benchmark, because the value of **l** for each fragment is within the correct range **(w=25)**. In contrast, this packet will be treated as **a** malicious packet by the fuzzy scoring system because the values of **std_a** and **der_l** are non-zero or near zero, so its score will be reduced and the attack will fail. As show the value in Table 3.

TABLE 3: THE VALUE OF MALICIOUS

Variable	Value
(a) Time between sending a packet and the other	1, 1, 9, 34, 15, 5, 20, 6, 24, 83, 28,43
(I) Real arrival time of packet	1, 1, 26, 34, 10, 41, 46, 50
(std) Denotes the standard deviation of α's values, mean	0, 25, 8, 7, 5, 4
(der) Denotes the absolute value of derivative of l	11.301

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

8. SIMULATION

In simulation, MATLAB environment has been used to simulate the grid. For the first and second attacks, data packet paths are pre-defined because the issue of routing is beyond the scope of this search, and only some nodes can create data packets as all data packets generated from the same node go the same way (the elimination of a packet occurs only between the legitimate and malicious packet). When the malicious nodes are waiting for the legitimate contract in the transmission process where they are fully prepared, the legitimate nodes are sent at the same time the malicious nodes are sent to occupy space and take space to send their nodes. Again, the malicious nodes are sent multiple times to reach the overload. If send the malicious nodes as much as the legitimate contract, the success rate is half and may fail. Therefore 11 malicious attack probability values (0% to 100%, 10%), are applied to all nodes and confirmation of the contract received if it is malicious or legitimate. The following parameters in Table 3: are the most important parameters used in simulation. The Performance measures were used to evaluate the performance of Fuzzy system. Typically, other network parameters are used to assess the extent to which the routing protocols are Poorly understood or implemented. The performance measures listed below were used to evaluate the performance of this research study. The MATLAB environment was used in network simulation. First and second attacks, data packet paths are pre-defined because the routing issue is beyond the scope of this research, and only some nodes can create data Packets Generated from the same node go the same way, the performance measures listed below were used to evaluate the performance of this research study.

The entry of such attacks reduces the space available incoming packets. These attacks are in the form of legitimate, but illegal, attacks, exploiting the waiting time given to legitimate packages to enter and occupy a space of the network. The main objective of this research is to detect and expel these malicious attacks and prevent them from entering. A new FBS algorithm has been used to detect malicious attacks by calculating their values in terms of the number of points, the time of entry and the expected time between each packet. This algorithm has been successful in detecting attacks and preventing them from entering and maintaining the network from getting into overload.

TABLE 3: SIMULATION PARAMETERS

	First attack	*Second attack*
Number of nodes	50 nodes	50 nodes
Number of fragments in Packet	From 10 to 20 fragments	From 2 to 20 fragments
Time period between two fragments	0.1 sec	0.05 sec
Packet lifetime	35 sec	10 sec
Timeout period	3.5 sec	2 sec
Window (w)	0.25 sec	0.25 sec

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

9. COMPARISON

In each new study it is necessary to mention the difference between them and previous studies to see the extent of their impact and success. This paper has highlighted the discovery and entry of harmful network attacks, although many previous studies have conducted many attempts to solve the problem. The following differences illustrate the current work on the previous work:

i. The new FBS algorithm analyzes incoming packets and calculates the number of score contained in them and the data. Where the previous study counted only time.

ii. FBS arranges packets according to the time of arrival from the oldest to the most recent in the buffer before sending them, where other studies send packets at random.

iii. New rules are used to test the packet and determine its reliability.

iv. Using FBS does not expose the network to the overload status of not allowing fake packets to be entered after testing, because FBS algorithm detects attacks and prevents them from entering. Unlike other studies where only the presence of attacks and transmissions of non-test and the success is half or less.

v. There is a warehouse management where the results are received from FBS after analyzing, sorting and inserting legitimate packages, and deleting illegal packets. Unlike other studies where everyone is redirected to the network.

10. CONCLUSION

Despite recent studies by many researchers and scientists, development is taking place on Internet networks from time to time and a lot of new proposals and solutions are being put forward. However, there are a lot of changes to appear on the networks, disrupting the functioning of the system and disrupting its work. In this research, one of the problems facing the network from the outside and inside, which is a major problem, must be solved in order to eliminate the damage caused by the network. One of the important problems is the problem of overload that gets on the network; one of the causes of the occurrence of overload is the vulnerability of malicious attacks by users and other networks.

The entry of such attacks reduces the space available for incoming packets. These attacks are in the form of legitimate, but illegal, attacks, exploiting the waiting time given to legitimate packages to enter and occupy a space of the network.

The main objective of this research is to detect and expel these malicious attacks and prevent them from entering. A new FBS algorithm has been used to detect malicious attacks by calculating their values in terms of the number of points, the time of entry and the expected time between each packet.

11. ACKNOWLEDGEMENT

The authors would like to acknowledge the assistance provided by the Network and Communication Technology Research Group, FTSM, UKM in providing facilities throughout the research. The support given by Research Project Code No: DCP-2017-020/2 towards the writing of this paper.

REFERENCES

[1] Kuyoro, S., Et Al. (2015). Internet of Things (Iot): An Overview. 3rd International *Conference on Advances in Engineering Sciences & Applied Mathematics.*

[2] Ma, H.-D. (2011). "Internet of Things: Objectives and Scientific Challenges." *Journal of Computer Science and Technology 26(6): 919-924.*

[3] Ismail, N. H. A. B., Hassan, R., & Othman, N. E. (2014, August). LABC: Local Route Repair using Artificial Bee Colony algorithm in 6LoWPAN network. *In Computational Science and Technology (ICCST), 2014 International Conference on(pp. 1-5). IEEE.*

[4] Coetzee, L. And J. Eksteen (2011). The Internet of Things-Promise for the Future an Introduction. *IST-Africa Conference Proceedings, 2011, IEEE*

[5] Hassan, R., Jubair, A. M., Azmi, K., & Bakar, A. (2016, December). Adaptive congestion control mechanism in CoAP Application Protocol for Internet of Things (IoT). *In Signal Processing and Communication (ICSC), 2016 International Conference on (pp. 121-125). IEEE.*

[6] Ismail, N. H. A., Hassan, R., & Ghazali, K. W. (2012). A study on protocol stack in 6lowpan model. *Journal of Theoretical and Applied Information Technology, 41(2), 220-229.*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[7] Z. H. Hu, "The research of several key question of internet of things," in Proc. of 2011 Int. *Conf. on Intelligence Science and Information Engineering, pp. 362-365.*

[8] L. Atzori, A. Iera, and G. Morabito, "The internet of things: *A survey," Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010*

[9] R. Hummen, J. H. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle. Towards Viable *Certificate-based Authentication for the Internet of Things. In Proc. Of ACM HotWiSec, 20*

[10] T. Heer, O. Garcia-Morchon, R. Hummen, S. Keoh, S. Kumar, and K. Wehrle. Security Challenges in theIP-based Internet of Things. *Springer WirelessPersonal Communications Journal, 2011.*

[11] D. Tian, Q. Zeng, D. Wu, P. L. 0005, and C. Hu. Kruiser: Semisynchronized non-blocking concurrent kernel heap buffer overflow monitoring. *In Proceedings of the 19th Annual Network and Distributed System Security Symposium, 2012*

[12] Q. Zeng, J. Rhee, H. Zhang, N. Arora, G. Jiang, and P. Liu. DeltaPath: Precise and Scalable Calling Context Encoding. *In Symposium on Code Generation and Optimization, 2014.*

[13] I. Simon. A comparative analysis of methods of defense against buffer overflow attacks. *security/boflo.html, January 2001.*

[14] Bugzilla. Bug 2451 - CVE-2013-4243 libtiff (gif2tiff): possible heapbased buffer overflow in readgifimage(). *bugzilla.maptools.org/ show bug.cgi?id=2451.*

[15] D. Larochelle and D. Evans. Statically detecting likely buffer overflow vulnerabilities. *In Proceedings of the 2001 USENIX Security Symposium, Washington DC, USA, August, 2001*

[16] Zeng, Q., Zhao, M., & Liu, P. (2015, June). Heaptherapy: An efficient end-to-end solution against heap buffer overflows. *In Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on (pp. 485-496). IEEE.*

[17] Hummen, R., Hiller, J., Wirtz, H., Henze, M., Shafagh, H., & Wehrle, K. (2013, April). 6LoWPAN fragmentation attacks and mitigation mechanisms. In Proceedings of the *sixth ACM conference on Security and privacy in wireless and mobile networks (pp. 55-66). ACM.*

[18] Kim, E., & Kaspar, D. (2012). Design and application spaces *for IPv6 over low-power wireless personal area networks (6LoWPANs).*

[19] Gilad, Y., & Herzberg, A. (2011, August). Fragmentation considered vulnerable: blindly intercepting and discarding fragments. *In Proceedings of the 5th USENIX conference on Offensive technologies (pp. 2-2). USENIX Association.*

[20] Zeng, Q., Zhao, M., & Liu, P. (2015, June). Heaptherapy: An efficient end-to-end solution against heap buffer overflows. *In Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on (pp. 485-496). IEEE.*

[21] Wang, J., Zhao, M., Zeng, Q., Wu, D., & Liu, P. (2015, June). Risk assessment of buffer" Heartbleed" over-read vulnerabilities. *In Dependable Systems and Networks (DSN), 2015 45th Annual IEEE/IFIP International Conference on (pp. 555-562). IEEE.*

[22] Zeng, Q., Rhee, J., Zhang, H., Arora, N., Jiang, G., & Liu, P. (2014, February). Deltapath: Precise and scalable calling context encoding. *In Proceedings of Annual IEEE/ACM International Symposium on Code Generation and Optimization (p. 109). ACM.*

[23] Qiang Zeng*, Mingyi Zhao*, Peng Liu The Pennsylvania State University Park, PA, 16802, USAEmail: {quz105, muz127,.2015

[24] *Garg, V.K. and Rappaport, T.S., 2001. Wireless network evolution: 2G to 3G. Prentice Hall PTR.*

[25] Dongfang Li Zhenglin Liu Yizhi Zhao Department of electronic Science and Technology, Huazhong

[26] D Li, X Zhan, Q Tong, X Zou, Z Liu - Chinese *Journal of Electronics, 2016 - IET*

[27] Li, D., Liu, Z., & Zhao, Y. (2012, September). HeapDefender: A mechanism of defending embedded systems against heap overflow via hardware. In Ubiquitous Intelligence & Computing and 9th International *Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on(pp. 851-856). IEEE.*

[28] John Wilander and Mariam Kamkar Dept. *of Computer and Information Science, 2003*