Distributed IDS for Efficient Resource Management in Wireless Sensor Network

Eung Jun Cho, Choong Seon Hong
Department of Computer Engineering
Kyung Hee University
Yongin, Republic of Korea
ejcho@networking.khu.ac.kr, cshong@khu.ac.kr

Deokjai Choi
Department of Computer Engineering
Chonnam National University
Gwangju, Republic of Korea
dchoi@jnu.ac.kr

Abstract—Intrusion detection system (IDS) is one of the basic components of network security. IDS can use signatures or abnormal behaviors to detect attacks. When IDS uses signatures, in order to detect more attacks, we have to implement more attack signatures on IDS. However, in resource-constrained wireless sensor networks, it is not efficient to implement all of attack signatures. And it is hard to maintain attack signature database because it should be updated frequently when new kind of attacks appear. In this paper, we propose a distributed IDS mechanism which reduces the usage of memory and overhead. Analysis of our proposal shows that our proposal requires less memory.

Keywords—Intrusion Detection System; Wireless Sensor Network; security; resource efficient; Bloom filter

I. INTRODUCTION

Wireless Sensor Networks (WSN) are composed of several tiny sensor nodes. These sensor nodes have constrained resources, such as, low memory and low computational power. In spite of these limitations, wireless sensor networks can be used in various fields. Generally sensor nodes use IEEE 802.15.4 PHY/MAC [1] and Zigbee [2] protocol stack. In a special purpose, to support IPv6, 6LoWPAN (IPv6 Low Power Wireless Personal Area Network) [3] can be used instead of Zigbee. These kinds of mechanism can be used to build an Internet of Things (IoT). However, wireless sensor networks are vulnerable against many attacks, such as sinkhole [4], wormhole, etc. 6LoWPAN also introduces IP based attacks in WSN. Moreover, WSN can be deployed in various environments, such as, nuclear plants, power station, hospital, etc. Thus security issues in WSN are becoming so important nowadays. Because sensor nodes have limited battery power, low memory capacity, and low computational power, we have to design a resource-efficient mechanism which is implementable on sensor nodes.

An IDS (Intrusion Detection System) is a basic component of network security to detect attacks. The IDS can be categorized into two classes; one uses attack signatures and the other uses abnormal states to detect attacks. Because the IDS based on abnormal states requires high memory usage for storing the state of traffic and high computation power for the traffic analysis, it is not suitable for sensor nodes. The IDS based on attack signatures requires high memory for storing attack signatures. Moreover, the performance of the IDS depends upon the number of attack signatures. If we can obtain more attack signatures, the IDS can detect more attacks. However, in a resource-constrained sensor node, it is not feasible to implement complete attack signatures. To solve this problem, [5] proposes the mechanism to reduce the attack signatures significantly. In this paper, we overview the problem of the [5] and we propose a distributed IDS scheme.

This paper is organized as follow. Section II provides the background about Bloom filters [5], [6]. Section III shows the problem statement of [5] and the proposed mechanism. In section IV, we analyze the performance of our proposal and conclude this paper in section V.

II. BACKGROUNDS

A. Bloom Filter

A Bloom filter is a simple space-efficient randomized data structure which uses several hash functions [6]. Bloom filter has a false positive rate but the space reduction allows it to overwhelm when its false positive rate is made extremely low. Fig.1 shows a simple example of a Bloom filter operation process. As we can see in Fig. 1, all array spaces are filled with 0, initially. In Insert Phase, X1 and X2 are passed through a hash function which makes its result values. According to these values, corresponding bits are marked as 1. When membership queries are performed, Y1 and Y2 are passed through hash functions also. If the result values are perfectly matched, then we can say that Y1 and Y2 are the same values as X1 and X1.

This research was partially supported by the MKE, Korea, under the ITRC support program supervised by the NIPA*(NIPA-2011-C1090-1131-0005), and by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation. Dr. CS Hong is the corresponding author.
In this case, we can make the same length of corresponding bit sets for various lengths of the input data. And we can make a specific false positive rate by adjusting array size. This can be formulated as (1), when the number of input data is \( n \), array size for Bloom filter is \( m \), and number of hash functions is \( k \) [6].

\[
fpr = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx \left(1 - e^{-kn/m}\right)^k \tag{1}
\]

Equation (1) can be simplified as follows when \( k = 2 \) [5].

\[
m \geq n \log_2 e \log_2 \left(\frac{1}{fpr}\right) \approx 1.44n \log_2 \left(\frac{1}{fpr}\right) \tag{2}
\]

If we specify a fixed \( fpr \) (false positive rate) and the number of input data is confirmed, we can calculate minimum value of \( m \) which satisfies specific \( fpr \).

B. IDS mechanism using Bloom Filter

The work in [5] proposes the IDS for WSN using Bloom filter. In WSN, mechanisms which are implemented on sensor nodes should be resource-efficient because of resource-constrained characteristics of sensor nodes. When a sensor node sends and receives data, it uses much more energy than performing complex calculations. Thus it is important to reduce the messaging overhead when designing the mechanism for WSN. In [5], to reduce the memory usage of IDS based on signatures, authors utilized Bloom filter. To apply complete SNORT [7] rule set, including 13,339 number of attack patterns; it requires approximately 252KB of memory usage. However, when they use Bloom filter in IDS, they can minimize the memory usage by 13KB whereas the number of hash functions are 2 and \( fpr \) is 0.024.

When the specific attack is detected, we need to take an action against attack. To perform this action, we have to know that which kind of the attack is detected. But with Bloom filter array, we cannot identify the attack. To solve this problem, in [5] a signature-code is generated, and then the signature-code is sent to the gateway (or sink) where signature database resides.

Fig. 2 shows this operation. In signature database, there are corresponding rules which have information for the action against specific attacks.

III. PROBLEM STATEMENT AND PROPOSAL

In this section, we show problem statement of [5] and provide our proposal.

A. Problem Statement

In [5], authors utilize the Bloom filter to reduce the signature size. And when they detect the attack, sensor node sends signature-code to gateway to identify the attack. However, attacker can make sensor nodes to send useless message to the gateway because every Bloom filter has \( fpr \). Because sensor nodes are deployed in open area, an attacker can have access to sensor node and extracts the Bloom filter array information, easily. If an attacker can know messages which make false positive to Bloom filter array in sensor nodes, the attacker can send messages to several sensor nodes many times. Sensor nodes which receive the messages from the attacker will check them to find the malicious ones. After that, sensor nodes will send the signature-code to gateway. In aspect of gateway, a lot of sensor nodes start to send the signature-code, but it is not an attack signature-code. However this will be like a DoS (Denial of Service) attack on the gateway. And sensor nodes would waste their limited battery power.

B. Distributed IDS Mechanism

To solve the problem mentioned in previous section and to enhance the resource efficiency, we propose distributed IDS using multiple Bloom filter arrays. The followings are assumptions of our proposal.

- Wireless sensor network working in Ad-hoc manner.
- Multiple Bloom filter arrays are randomly deployed.

1) Selective and Feed-backed Distribution Phase

To make more efficient Distributed IDS, we propose a classification method to distribute the attack signatures for multiple Bloom filter arrays. In SNORT rules, there are priorities, 1, 2, and 3, which corresponds to the severity of attack signatures. 1 is for the most dangerous and harmful attacks while 3 is for the unusual traffic not identified as dangerous [7] [8]. In general, the 3rd priority is less important and even though we do not detect it, it does not cause serious problems. So we can distribute the 3rd priority attack signature at rate interval. On the other hand, we can locate 1st priority attack signature on distributed Bloom filter arrays more frequently. With this mechanism, we can enhance the security robustness of distributed IDS. Fig 3 shows an example of selective distribution. In Fig. 3, S1 has the highest priority and S3 has the lowest priority. As we can see, according to its priority, S1 is located on every Bloom filter arrays and S3 is located on only one Bloom filter array. In this distributed manner, we can detect most of the S1, which is the most dangerous type of attacks.

However with pre-mentioned mechanism, we cannot enhance the detection rate of distributed IDS. To accomplish higher detection rate, we propose feed-backed distribution mechanism.
IDS can be used to detect the attack and also can analyze the attack signature statistically. Whenever IDS detects attacks, it logs attacks on its database. With this information, we can classify the attack signatures into several levels by its frequency of detection. In other words, we can distribute frequent signatures more densely on distributed IDS. And we can distribute seldom ones less densely. With this scheme, we can distribute the attacks which might appear more frequently.

Table 1 shows a classification rule of selective and feedbacked distribution. Row means frequency of attack signatures which is reported by each IDS. And column means priority of the attack signature which is defined in SNORT rules. With proposed distribution mechanism we can enhance both security level and detection rates while reducing the resource usage. The value between 1 and 10 means relative weight of distribution. If the value is high, it should be located in more Bloom filter arrays than lower one.

2) Operation of Distributed IDS

Due to the absence of infrastructure, sensor nodes communicate in multi-hop Ad-hoc manner. It means that a sensor node is not only the end point of communication, but also the relay node to forward the data. In this situation, many nodes participate in data delivery from source to destination. This means that one data packet passes through many sensor nodes. Even though we distribute the attack signatures in many sensor nodes, we can maintain the specific fpr when each sensor nodes lower their Bloom filter array size. Fig. 4 shows how relay node acts as a part of IDS. When the sender sends M1 to the receiver, two relay nodes support to transmit the M1. When M1 passes through relay nodes, each relay node checks the M1 using its own Bloom filter array. As we discussed previously in part A of section II, Bloom filter requires large array size to lower the false positive rate. According to (2) in part A of section II, we need more array space when the number of input data increased and fpr is fixed.

<table>
<thead>
<tr>
<th>Frequency &amp; Priority</th>
<th>1 (High rate)</th>
<th>2</th>
<th>3 (Low rate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

In other words, if we can reduce the number of input data, we can save the size of Bloom filter array. Thus, if we use multiple Bloom filter arrays, we can reduce the number of input data for each Bloom filter array. This means that we can accomplish the specific fpr value, reducing the Bloom filter array size. However data has independent false positive rates at every different Bloom filter array, we have to adjust fpr value to accomplish the same fpr level. In example, if we set the fpr value as 0.024 and make 2 Bloom filter arrays, total fpr value will be 0.048 for one input data. To maintain the fpr value in same level, we have to divide the fpr by the number of Bloom filter array when we make multiple Bloom filter arrays. We can derive the equation (3) from (2) to calculate minimum size of Bloom filter arrays m, when j is the number of Bloom filter arrays:

\[ m \geq 1.44 \frac{n}{j} \log_2 \left( \frac{j}{fpr} \right) \]  

As we can see in (3), m is proportional to \( \frac{\log_2 j}{j} \) and as \( j > \log_2 j, m \) is inversely proportional to j.

To solve the problem statement we mentioned in part A in this section, we implement signature database into sensor node with Bloom filter array. The size of signature-code generated by Bloom filter is 4 bytes; when we use 13kB Bloom filter array and each hash function have to generate the number between 0 and 13,312. The number 13,312 is less than 65,536 which can be express in \( 2^{16} \), the same as 2 bytes (= 16 bits). Because [5] uses 2 hash functions, signature-code size is 4 bytes. To assign a space for corresponding rules; we can use 1 byte to express 256 kinds of rules. This means that the signature database requires 5 bytes per every single attack signature. To store 13,339 number of attack signatures into the signature database, it requires about 67kB size of memory.

IV. PERFORMANCE ANALYSIS

In this section, we analyze the detection rate according to the number of Bloom filter arrays j, and average number of hops h. In this analysis, we don’t consider network topology. And we also analyze the memory usage according to the number of Bloom filter arrays j.
Fig. 5 shows detection rates according to the number of Bloom filter arrays $j$, and average number of hops $h$, when attack signatures are randomly distributed. To achieve over 95% detection rate, the average number of hops should be 4 to 10. In terms of practical viewpoint, it is the most efficient when using 2 number of Bloom filter arrays.

Fig. 6 shows detection rates according to the number of Bloom filter arrays $j$, and average number of hops $h$, when attack signatures are selectively distributed. We set $a$, $b$, and $c$ as 0.3, 0.48 and 0.22, respectively, according to classification.conf file which is included in SNORT. As we can see, when we applied proposed selective distribution mechanism, we can enhance the detection rate significantly. In Fig. 5, to accomplish 95% detection rate, we need to reduce the number of Bloom filter arrays to 2 or have more than 10 hops number. But when we apply our selective distribution, we just require the hop number as 6.

In [5], they require 13kB to apply rule set of SNORT 2.8 for their proposed IDS. And we already discuss that signature database requires 5 bytes per every single attack signature. Based on this information, we can calculate the minimum requirement of memory on each sensor node as (6) when the number of Bloom filter arrays is $j$.

$$m \geq 1.44 \frac{n}{j} \log_2 \left( \frac{j}{j_{pr}} \right) + \frac{n}{j} \times 40 \quad (6)$$

And we can derive the equation (7) which applies the proposed selective distribution mechanism as follows, when $a$, $b$, and $c$ are 0.3, 0.48 and 0.22, respectively.

$$m \geq 1.44 \log_2 \left( \frac{j}{j_{pr}} \right) \left( an + \frac{bn}{2} + \frac{cn}{j} \right) + \frac{40}{j} \left( an + \frac{bn}{2} + \frac{cn}{j} \right) \quad (7)$$

Fig. 7 shows the change of minimum memory requirement $m$ according to the number of Bloom filter arrays $j$, when input data $n$ is 13,312.
And we can compare the memory usage of proposed selective distribution mechanism to decide whether it should be applied or not. As we can see, selective distribution mechanism requires more memory but this is the tradeoff between detection rates and memory usage. And we can see that when we implement signature database on each sensor nodes, it requires more memory usage than [5]. However, as we distribute the Bloom filter arrays, minimum memory requirement is also decreased. If we compare the memory requirement for Bloom filter array itself, we can see that our proposal requires less memory usage than that of [5].

Our proposal can eliminate the messaging overheads which send the signature-code to gateway. And even though false positives occur in Bloom filter, sensor nodes can handle them using signature database. This will reduce the energy consumption for sensor nodes to send signature-code query messages to the gateway. The gateway can also reduce the overhead in response to the signature-code query.

V. CONCLUSION AND FUTURE WORKS

In this paper, we proposed resource-efficient distributed IDS using multiple Bloom filter arrays to eliminate additional messages. We analyzed our proposal mathematically. Our proposal requires less memory usage than previous works. Also, we enhanced our proposal using selective distribution of attack signatures. In the future, we will improve our proposal by performing the simulation on real network topology environment. And we can take advantage from the feed-backed distribution mechanism.

REFERENCES