Risk-Driven Aggregation and Transmission Prioritization of Cyber Alerts over Mobile Networks

Hasan Cam and Pierre Mouallem

Network Science Division
Army Research Laboratory
Adelphi, MD 20783

Abstract—Alert Aggregation in mobile networks plays an important role in mitigating the adverse impact of alert generation by reducing the amount of communication and security data to be transmitted. However, it is not guaranteed that the bandwidth necessary to transmit all aggregated alerts is always available, which usually result in the transmission of a portion of the alerts, while others are discarded or queued. The transmission of insufficient alert information hinders making correct decisions about attacks, leading to compromising network security. In order to maximize the benefits of data aggregation while minimizing the impact of partial alerts, this paper presents a risk-driven real-time transmission prioritization technique for implementing lossy and lossless aggregation of cyber alerts. Lossy alert aggregation and transmission are managed adaptively by allowing the prioritization and transmission of aggregated alerts according to the risk assessment of such alerts. This paper also presents a risk-driven utilization model that further adapts the aggregation and prioritization in response to dynamic network conditions. The performance results of the proposed techniques are obtained by running simulations on data collected from a mobile network. Simulation results for the aggregation of raw alerts have shown an average reduction of 51% in data storage space and bandwidth usage.

Keywords—Mobile Networks, Cyber Security, Intrusion Detection System, Alert Aggregation, Transmission Prioritization

I. INTRODUCTION

Network-based and host-based intrusion detection systems (IDS) in wireless mobile networks rely on the ability of mobile nodes to monitor each other’s transmission activities and to analyze packet contents to detect intrusions. These IDSs usually generate high amounts of false alarms [1, 2], leading to poor intrusion detection performance and adversely affecting the already bandwidth-limited communication medium of wireless mobile networks. Alert aggregation is usually used to reduce the amount of alert data without losing important information. However since bandwidth is limited, it is not guaranteed that all alerts can be transmitted.

Risk is defined as “the probability or threat of damage, loss, or any other negative occurrence that is caused by external or internal vulnerabilities, and that may be avoided through preemptive action” [3]. In mobile networks, risk is presented when insufficient alert information is transmitted for analysis, thus hindering the ability to make correct decisions about attacks, leading to compromising network security. To mitigate that adverse effect, aggregation techniques are needed that prioritize and transmit alerts based on their criticalities, so that the most critical alerts are transmitted first, and less critical alerts are queued. By doing so, we minimize the overall risk of the system since the chance of having important alerts disregarded is significantly reduced.

In order to determine the rate of aggregation, this paper presents a utilization model that aims at maximizing the utilization of the available bandwidth, and reducing the risk associated with not transmitting all the generated alerts. To this end, two modes of transmissions are used, namely lossless and lossy. Lossless transmission is used when there is enough bandwidth available to transmit all meta-alerts, including their log data, whereas lossy transmission is used when the available bandwidth is not sufficient. The degree of lossiness in lossless transmission is dynamically calculated using the utilization model presented so that the available bandwidth is fully utilized. This technique differs from related work [4, 5] where they adapt the alert aggregation based solely on bandwidth availability by also taking into consideration the risk factor.

The rest of this paper is organized as follows: Section 2 discusses the related work. Section 3 presents the proposed alert aggregation and transmission prioritization technique. Section 4 discusses the bandwidth utilization analysis. Section 5 describes the experimental setup and results. Section 6 concludes with final remarks and future work.

II. RELATED WORK

Alert aggregation techniques in [6, 7, 8] aim at mitigating the adverse impact of high rates of intrusion detection systems alerts. They simplify the processing of alerts and aid in their analysis. However most of these techniques focus on wired networks where bandwidth and distance is not usually a concern. Fewer research efforts have addressed mobile ad-hoc network based IDS. Some of that research includes consensus voting among mobile nodes, adding additional components to detect routing misbehavior, or a combination of the two [9]. However all those techniques aggregate alerts as they are generated, without any risk-related analysis prior or throughout aggregation and transmission. This paper

U.S. Government work not protected by U.S. copyright
addresses that gap by using a risk analysis to drive aggregation and transmission prioritization.

Data aggregation is also used to enhance bandwidth utilization [10, 11]. But, this type of work treat all alerts equally without considering their criticalities, probably causing adverse impact on the system security. However, this paper incorporates risk assessment into the utilization analysis, to more effectively use the available bandwidth. The goal is to help analysts or systems administrators take appropriate actions based on the criticality of alerts.

III. RISK-DRIVEN ALERT AGGREGATION AND TRANSMISSION PRIORITIZATION

This section presents the alert aggregation and transmission prioritization technique. We begin by describing the network topology in which this technique operates, and then present the proposed technique.

A. Network Model for Hop-based Alert Transmission

The mobile network covered in this paper is comprised of mobile nodes, each running an IDS, and a base station or Forward Operating Base (FOB), also known as base station, that receives the alerts from the nodes and analyzes them. Mobile nodes are dispersed through a site, and only nodes that are within one-hop range of the FOB can directly communicate with it. Figure 1 illustrates a hop-based alert transmission in a mobile network.

B. Risk-Driven Alert Aggregation

The proposed alert aggregation technique aggregates alerts based on their attributes. A number of alert attribute sets are formed such that each set represents one or more alert attributes, where an attribute can be a member of more than one set. Some examples of attribute sets could be: \{signatureID\}, \{signatureID, srcIP\}, or \{signatureID, srcIP, dstIP, portID\}. The meta-alerts described by each set are kept in a separate queue. All parameters not included in these attribute sets are stored as a single list within the meta-alert. Figure 3 shows an example of alert aggregation. It shows the aggregation of three raw alerts generated at a single node into one meta-alert based on two attributes, srcIP and SigID. As mentioned, parameters not included in the attribute sets, such as port ID (e.g., portID1, portID2) and destination IP (e.g., dstIP1, dstIP2) along with the raw log data, are stored in the rightmost field of the meta-alert as shown in figure 3.

Since each node in the network can receive meta-alerts from adjacent nodes that are farther from the FOB, each node is capable of performing two types of alert aggregation: (i) Local Alert Aggregation and (ii) non-local Alert Aggregation.
For the first type, each node aggregates its own alerts, whereas in the second type, each node aggregates the meta-alerts received from other nodes with the locally generated meta-alerts.

Once alerts are aggregated, they are queued pending the availability of a transmission window. Since bandwidth is limited in these types of deployments, two modes of transmissions are used:

a) **Lossless Transmission**, which is used when there is enough bandwidth available to transmit all meta-alerts, including their log data. In this case, the compression rate is relatively low, as raw log data cannot be meaningfully compressed. Based on our experiments, using this method reduces the overall data transmitted by an average of 5%.

b) **Lossy Transmission**, which is used when the available bandwidth is not sufficient. The degree of lossiness is dynamically calculated using a utilization analysis so that the available bandwidth is fully utilized. Since the log data comprises most of the actual alert data (approximately 93%) the degree of lossiness is controlled by varying the amount of log data included in the meta-alerts.

The transmission mode is determined based on the analysis of bandwidth availability and utilization. By default lossless aggregation is chosen and if a pre-determined threshold for bandwidth availability is crossed, the aggregation mode is changed to lossy and the degree of lossiness is dynamically assessed based on the network conditions.

Additionally, meta-alerts are prioritized based on their risk assessment so that the most critical alerts get transmitted first, regardless of the transmission mode used. Alert prioritization is achieved by calculating the probability distribution for each queue. The meta-alerts of the queue with the highest value are chosen to be forwarded to the next node in the network (or to the FOB in the case of the nodes connected to the FOB).

Figure 4 shows an example for a single node of the aggregation of generated alerts, and the aggregation of local and non-local meta-alerts. It also shows the transmission prioritization of queues according to their criticalities, where $Q_j$ represents meta-alerts with the highest criticality and $Q_i$ represents meta-alerts with the lowest criticality.

![Fig. 4. Risk-Driven Aggregation and Prioritization of local and non-local meta-alerts at a single node](image)

To aggregate alerts based on the alert attributes or parameters, our technique assumes: (i) an attack instance is a random process generating alerts, and (ii) attributes of alerts have different probability distributions. We compute categorical attributes which have multinomial distributions.

Note that the multinomial distribution of an alert with a categorical attribute set is equal to the product of probabilities of all event(s) that lead to the alert generation. If an alert has two categorical attributes, say “signature ID” and “source IP”, its multinomial distribution equals the product of their probabilities. In a multinomial distribution, each trial results in one of some fixed finite number $k$ of possible outcomes, with probabilities $p_1, \ldots, p_k$.

The probability mass function of multinomial distribution for meta-alerts in the queue $j$, denoted $Q_j$, can be expressed as:

$$P(Q_j) = \frac{N^j}{(n_1^j!n_2^j!\cdots n_k^j!)} \prod_{i=1}^{k} p_i^{n_i}$$

(1)
where

\( N_j \) is the number of raw alerts assigned to \( Q_j \),

\( k \) is the number meta-alerts in \( Q_j \),

\( n_i \) is the number of alerts aggregated in meta-alert \( i \),

\( p_i \) is the probability of observing meta-alert \( i \).

Note that each attribute has a (relative) weight with respect to the other attributes. Moreover, particular instances of an attribute have varying weights, depending on their criticality. For example, source IP address has a weight based on its importance among all attributes. Some specific source IP addresses may have more weights than others, depending on their occurrence frequency in alerts data. The availability of computing resources dictates how often these weights are updated using a risk analysis, which includes a combination of historical data, analysts feedback and alert criticality. Given all these constraints, \( p_i \), can be calculated as follows:

\[
p_i = W_{Ai} \times W_{Iai} \times \frac{n_i}{N_j} \times r_i
\]

(2)

where

\( p_i \) is the probability of meta-alert \( i \),

\( W_{Ai} \) is the weight of attributes of the queue containing meta-alert \( i \),

\( W_{Iai} \) is the weight of the instances of attributes of meta-alert \( i \),

\( n_i \) is the number of alerts in meta-alert \( i \),

\( N_j \) is the number of raw alerts in queue \( Q_j \),

\( r_i \) is the criticality of meta-alert \( i \), which ranges between 0 and 1, where 0 represents no risk and 1 represents maximum risk.

Since bandwidth is limited in this type of deployments, meta-alerts are only transmitted when bandwidth is available.

IV. BANDWIDTH UTILIZATION ANALYSIS OF NODES

In order to determine the degree of lossliness in alert transmission, a utilization analysis is performed to dynamically adapt the degree of data aggregation while meeting mission deadlines. Since the alerts generated have an expiry time, they have to be delivered to the FOB before their deadline expires.

The bandwidth allocated to a node is usually divided into guaranteed and unguaranteed bandwidths, which are dynamically allocated based on the network state. To calculate the utilization of the overall bandwidth, the utilization of each type of bandwidth is calculated separately then summed up as follows:

\[
U_k = U_g + U_u
\]

(3)

Where \( U_g \) is the utilization of the guaranteed bandwidth, and \( U_u \) is the utilization of the unguaranteed bandwidth.

The authors in [4] present a utilization model as a factor of average transmission time and relative deadlines of packets. We extend that notion by including the criticality of the data and apply it to alert data. To that end, the utilization at node \( k \) can be calculated as follows:

\[
U_k = \sum_{i=1}^{p} \sum_{j=1}^{N} \frac{T_{ij} \times C_{ij}}{D_{ij}}
\]

(4)

where \( k \) is the number of nodes in the network,

\( N \) is the distance between node \( k \) and the FOB in hops, \( P \) is the number of meta-alerts that are transmitted by node \( k \), \( T_{ij} \) is the average transmission time of meta-alert \( i \) at node \( j \), \( C_{ij} \) is the criticality of meta-alert \( i \) at node \( j \), and \( D_{ij} \) is the relative deadline of meta-alert \( i \).

V. EXPERIMENTAL VALIDATION

A. Experimental Setup

In order to validate the aggregation and prioritization techniques presented in this paper, alerts data are collected across twenty nodes of a network, each running an intrusion detection system. Figure 5 illustrates connectivity of these nodes and FOB within a particular time interval, which determines the locations and directions of aggregation process of this experiment. Alerts flow from the outer most nodes towards the FOB. For simplicity we assume that no routing loops exist and that each node transmits to a single node.

To simplify the experiment, we assume that local and non-local alert aggregation is done using the same attribute sets across all nodes, with each attribute set corresponding to a meta-alert Queue. The total number of alerts collected was approximately 1.5 million across the 10 nodes. Table 1 shows the number of alerts collected at each node.

![Node wireless connectivity at a given time.](image)

**Fig. 5.** Node wireless connectivity at a given time.

**TABLE I. RAW ALERT COUNT PER NODE**

<table>
<thead>
<tr>
<th>Node ID</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Alerts (in 1000s)</td>
<td>339</td>
<td>148</td>
<td>217</td>
<td>228</td>
<td>207</td>
<td>34</td>
<td>43</td>
<td>23</td>
<td>112</td>
<td>100</td>
</tr>
<tr>
<td>Raw Alerts (in 1000s)</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>Raw Alerts (in 1000s)</td>
<td>221</td>
<td>101</td>
<td>412</td>
<td>182</td>
<td>220</td>
<td>112</td>
<td>98</td>
<td>75</td>
<td>182</td>
<td>127</td>
</tr>
</tbody>
</table>

B. Experimental Results

We first run aggregate the alerts locally at each node. The raw alerts are aggregated and divided among five queues. The resulting meta-alerts are shown in table 2.

The total number of meta-alert in table 2 represents a worst case scenario, when no non-local meta-alert aggregation
occurs, meaning the FOB will receive all the meta-alerts generated at each node if bandwidth is available. The total number of meta-alerts generated is 107028, which represents a reduction of 96% when compared to the total number of raw alerts (≈3.18 million).

When taking into account local and non-local alert aggregation, a best case scenario would be when each node receives all possible non-local meta-alerts before it transmits its own meta-alerts, resulting in further compression. Table 2 shows the results of simulating this scenario, where nodes N3, N4 and N5 do not transmit until they receive the meta-alerts from nodes N6, N7, N8, N9 and N10, etc. Similarly nodes N1 and N2 do not transmit until they received the meta-alerts from nodes N3, N4 and N5.

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Local MA</th>
<th>Received Non-local MA</th>
<th>Aggregated local and non-local MA</th>
<th>Total MA received at the FOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7353</td>
<td>26547</td>
<td>31047</td>
<td>84196</td>
</tr>
<tr>
<td>2</td>
<td>5394</td>
<td>52274</td>
<td>53149</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2493</td>
<td>25104</td>
<td>26547</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9405</td>
<td>12901</td>
<td>20134</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>14321</td>
<td>24466</td>
<td>32140</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1302</td>
<td>20293</td>
<td>21054</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4050</td>
<td>--</td>
<td>4050</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>1987</td>
<td>12242</td>
<td>12901</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6577</td>
<td>--</td>
<td>6577</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5225</td>
<td>8348</td>
<td>11547</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>2493</td>
<td>5545</td>
<td>6342</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>9102</td>
<td>--</td>
<td>9102</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>2392</td>
<td>--</td>
<td>2392</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>8799</td>
<td>--</td>
<td>8799</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>11213</td>
<td>--</td>
<td>11213</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1029</td>
<td>--</td>
<td>1029</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1125</td>
<td>--</td>
<td>1125</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1486</td>
<td>--</td>
<td>1486</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>5737</td>
<td>--</td>
<td>5737</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>5545</td>
<td>--</td>
<td>5545</td>
<td></td>
</tr>
</tbody>
</table>

The total number of meta-alerts received at the FOB in the best case scenario is 84196 as shown in Table 2, which represents a reduction of 21.3% when compared to the total number of meta alerts where only local aggregation occurs (107028), and a reduction of 96.8% when compared to raw alerts (3.18 million).

Next we compare the size on disk of the raw alerts to the size of meta-alerts. Additionally we compute the size of the logs for each alert (also known as trim file, which represents the raw data stream for the time where the alert occurred). As mentioned earlier, Lossless transmission transmits all the alerts and their log files, whereas lossy transmission transmits all alerts with a subset of their log files, which can vary between 0% and 99%. As shown in Table 3, the reduction in the required bandwidth greatly varies (between 7.96% and 92.57%) as it heavily depends on the aggregation mode and on the degree of lossliness.

**VI. CONCLUSION**

This paper has presented a new risk-driven alert data aggregation and transmission prioritization technique based on alerts attribute sets for alerts generated by intrusion detection systems in mobile networks. We’ve also simulated our aggregation techniques using real alerts captured from a mobile network and the results show that our techniques can achieve a very high percentage of aggregation and adapt to the available bandwidth to maximize its efficiency. This paper does not consider computing power or the energy available at each node. Taking them into consideration might affect how often we can prioritize meta-alerts since the prioritization process is a computing intensive operation. Future work will include further analysis of those additional parameters, mobility and robustness.

**REFERENCES**


