Improved Single Keyword Pattern Matching Algorithm for Intrusion Detection System

K. Prabha, Ph.D Research Scholar, Erode Arts and Science College, Erode, Tamil Nadu, India

S. Sukumaran, Ph.D Associate Professor of Computer Science, Erode Arts and Science College, Erode, Tamil Nadu, India

ABSTRACT
With the spreading of the internet and online procedures requesting a secure channel, it has become an inevitable requirement to provide the network security. It is very clear that firewalls are not enough to secure a network completely because the attacks committed from outside of the network are stopped whereas inside attacks are not. This is the situation where intrusions detection systems (IDSs) are in charge. IDSs are used in order to stop attacks, recover from them with the minimum loss or analyze the security problems.

String matching algorithms are essential for IDS that filter packets and flows based on their payload. This work describes the concept of single keyword pattern matching algorithms. A new improved single keyword pattern matching algorithm is proposed. The new method reduces character comparisons, faster and more reliable in network security applications. The experimental results show that the new algorithm is highly efficient. Its search time is cut down significantly compared with other popular existing algorithms and its memory occupation stays at a low level. Moreover, conclusion on results is made and direction for future works is presented.

Keywords
Network Security, Pattern matching, Intrusion Detection

1. INTRODUCTION
Security attacks through internet have proliferated in recent years. Hence, information security is an issue of very serious global concern of the present time. The need for network security and in particular the need for Intrusion Detection Systems (IDS) have been brought out. IDSs, as originally introduced by Anderson [3] in 1980 and later formalized by Denning [1] in 1987, have received increasing attention in the recent years.

IDS are widely used and heavily depended upon. The continued growth in both network traffic and intrusion signature databases makes the performance of these systems increasingly challenging and important. Intrusions refer to the network attacks against vulnerable services, data-driven attacks on applications, host-based attacks like privilege escalation, unauthorized logins and access to sensitive files, or malware like viruses, worms and trojan horses. Intrusion detection means detecting unauthorized use of a system or attacks on a system or network.

IDS are implemented in software or hardware in order to detect these activities. IDSs collect information from a computer or a computer network in order to detect attacks and misuses of the system. Three types of data are used by IDSs. These are network traffic data, system level test data and system status files [2].
hen long patterns are consider as the advantage of this algorithm. But its ]. On the other hand, the beamGs is the basis for 7

{Algorithm brute (text, pattern)
  n = length(text)
  m = length(pattern)
  for i=0 to (n-m)
    { j = 0
      while((j<m) and (text(i+j) = pattern(j))
        j++
      if j = m
        return i //match at i
      return -1 // no match
  
  2.4 Karp-Rabin Algorithm (KR)
  The Karp-Rabin Algorithm was created by Michael Rabin and Richard Karp. They used a completely different approach than the single keyword methods[13]. The main idea is that instead of using comparisons it involves mathematical computations which more specifically extends to the notion of hashing. The application of hashing (converting each string into a numeric value) has always been a useful approach when it comes down to string matching. If both words have different hash values then we conclude they are different. But if their hash values are the same we cannot conclude they are the same string and will have to perform further comparisons.
  
  for i=0 to (n-m)
    { j = 0
      while((j<m) and (text(i+j) = pattern(j))
        j++
      if j = m
        return i //match at i
      return -1 // no match
  
  Karp-Rabin-Matcher(T,P,d,q)
  n = length(T)
  m = length(P)
  h = d^m mod q
  p = 0
  t0 = 0
  for i = 1 to m //preprocessing
    { p = (d*p + P[i]) mod q
      t0 = (d*t0 + T[i]) mod q
    }
  for s = 0 to n-m //matching
    { if p = t0
      if P[1..m] = T[s+1..s+m]
        print "Pattern occurs with shift" s
      if s < n-m
        ts+1 = (d*(ts+T[s+1]*h) + T[s+m+1]) mod q
    }
  
  3. PROPOSED METHOD
  The improved single keyword pattern matching algorithm which is formulated based on the two algorithms Horspool and Karp-Rabin algorithm. Karp-Rabin algorithm is based on hashing approach but not the comparison of characters, which is consider as the advantage of this algorithm. But its weakness is the enormous time needed when long patterns are present [17]. On the other hand, the Horspool algorithm is easy and works in any order. In most situations that it applied on and has a high performance compare to other algorithms. It is easy to implement and has less memory space so, it can be implement in any case that need the exact string matching algorithm for small pattern and large pattern size [14].
  
  3.1 Improved single keyword pattern matching algorithm (ISPMA)
  The two phases of the proposed algorithm are (i) preprocessing phase, (ii) searching phase.
Step 1: In the first phase, the ISPMA performs the same preprocessing phase as in the existing two algorithms. It prepares the hash function used in KR algorithm and the bmBc table used in HP algorithm for the pattern.

Step 2: The process of computing hash functions for the patterns and text window are exactly the same as the process of creating them in the existing KR algorithm. The bmBc table is the same as it was in the existing HP algorithm.

In the searching phase, the ISPMA performs the comparison between the pattern and the text by utilizing the advantages of the KR and HP.

Step 3: After the preprocessing phase has finished, the comparison starts between the text and pattern by comparing the numerical value of pattern hash and window text hash.

Step 4: Whether if the two hash value are not identical then the ISPMA perform the shifting.

Step 5: Next Shift to the right based on the values of right most character for the window text in the bmBc table. This will speed up the algorithm during the comparison process and it reduces the number of character comparison by using the hash function.

For example, Take the pattern (PQRPRPRP) After preprocessing phase,

\[ \text{Hash (x)} = 17597 \]
\[ \text{Hash (y)} = 17533 \]

Fig 3.1 Preprocessing Phase

For the previous pattern (PQRPRPRP) and the text (PQRSQPQRPRPRPSRSRQRPS). The searching phase for the ISPMA is depicted in below Figure.

\[ \text{Fig 3.2 Searching Phase} \]

Here the Hash (x) = 17597 and Hash (y) = 17533 so the hash value are not equals. The ISPMA shift the value of character P in the bmBc table which is 2. Similarly, the process of ISPMA continues until all characters in the text are being compared and whether the mismatching or matching is found.

4. RESULTS

The proposed approach is implemented using MATLAB. The evaluation of the proposed method is performed based on the factors Efficiency, Runtime, Space and Accuracy.

The result of the experiments are presented below:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Character Comparison</th>
<th>Number of Attempts</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>KR</td>
<td>12</td>
<td>8</td>
</tr>
<tr>
<td>HP</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>ISPMA</td>
<td>8</td>
<td>6</td>
</tr>
</tbody>
</table>

Fig 4.1 Efficiency Comparison

The result shows that the ISPMA reduces the number of character comparison to 8 and reduces the number of attempts to 6. This is because of hashing approach of Karp- Rabin algorithm to perform the character comparison and depends on shift table of Horspool algorithm to perform the movement of pattern.

4.1 Time performance

The running-time performance, also referred to as time complexity, is measured in number of machine steps, and in this case we are primarily concerned with character or byte comparisons. To present the results of the running time of algorithms, we vary the input size, where the input is the English words. The number of patterns to be matched remains the same. The running time (in milliseconds) for the algorithms are recorded in the following table:

<table>
<thead>
<tr>
<th>Input Size</th>
<th>BM</th>
<th>KR</th>
<th>HP</th>
<th>ISPMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>20000</td>
<td>15</td>
<td>15</td>
<td>17</td>
<td>13</td>
</tr>
<tr>
<td>60000</td>
<td>40</td>
<td>45</td>
<td>46</td>
<td>38</td>
</tr>
<tr>
<td>100000</td>
<td>68</td>
<td>73</td>
<td>79</td>
<td>65</td>
</tr>
<tr>
<td>140000</td>
<td>102</td>
<td>102</td>
<td>108</td>
<td>101</td>
</tr>
<tr>
<td>180000</td>
<td>119</td>
<td>132</td>
<td>139</td>
<td>115</td>
</tr>
<tr>
<td>200000</td>
<td>133</td>
<td>144</td>
<td>159</td>
<td>123</td>
</tr>
</tbody>
</table>

Fig 3.3 Searching Phase for next window test
4.2 Space performance

The amount of memory consumed while the algorithm runs, is considered only in addition to the necessary space to store the keyword and input. The keyword and the keyword set must always be stored. The space performances of proposed algorithm and the three algorithms were compared using one pattern. The results are shown in Fig.4.3. The results show that ISPMA size is smaller than that of the other algorithms, for the same pattern.

Table 4.3 Space Comparison

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Pattern Length (Byte)</th>
<th>Memory Space (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM</td>
<td>20</td>
<td>180</td>
</tr>
<tr>
<td>KR</td>
<td>20</td>
<td>160</td>
</tr>
<tr>
<td>HP</td>
<td>20</td>
<td>140</td>
</tr>
<tr>
<td>ISPMA</td>
<td>20</td>
<td>100</td>
</tr>
</tbody>
</table>

4.3 Accuracy Performance

The number of patterns are vary, the accuracy for the four algorithms are shown in Figure 4.4. Horspool and Boyer-Moore have the minimum accuracy because the shifted values are affected by increasing signature length. The difference between them is very small. Karp-Rabin and improved single keyword pattern matching algorithms were not affected by the increased signature length because their shifted values are always one byte.

Table 4.4 Accuracy Comparison

<table>
<thead>
<tr>
<th>Number of patterns</th>
<th>Accuracy (%)</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BM</td>
<td>KR</td>
<td>HP</td>
<td>ISPMA</td>
</tr>
<tr>
<td>50</td>
<td>92</td>
<td>95</td>
<td>95</td>
<td>99.5</td>
</tr>
<tr>
<td>100</td>
<td>91.8</td>
<td>94.5</td>
<td>94.5</td>
<td>99</td>
</tr>
<tr>
<td>150</td>
<td>91</td>
<td>94.3</td>
<td>94.3</td>
<td>98.5</td>
</tr>
<tr>
<td>200</td>
<td>90.6</td>
<td>94</td>
<td>93</td>
<td>98.2</td>
</tr>
<tr>
<td>250</td>
<td>90.3</td>
<td>93.7</td>
<td>92.5</td>
<td>98</td>
</tr>
</tbody>
</table>

5. CONCLUSION

This work identifies the number of promising algorithms and provides an overview of recent developments in the single keyword pattern matching for IDS. Boyer-Moore Algorithm uses two tables and matching starts with right to left, but in Horspool uses only one table and the matching is faster than the Boyer-Moore. The Brute force algorithm requires no preprocessing of the pattern. Karp-Rabin algorithm is based on hashing approach. The proposed ISPMA algorithm is compared with the exiting algorithms and the result shows that the algorithm is faster and more reliable in network security applications. The results of algorithm show an improvement in average comparing, faster than the original algorithms, less character comparison and performs less number of attempts compared to the exiting algorithms. In future work, we will enhance the method by moving toward the parallel computing to reduce the workload of system and consequently improve the speed and accuracy of the detection of malicious activities.

6. REFERENCES


7. ABOUT AUTHORS

K. Prabha received B.Sc Computer Science and M.Sc Computer Science Degree from Bharathiar University, Coimbatore and M.Phil in Periyar University, Salem. She is pursuing Ph.D degree in Computer Science at Bharathiar University. She has 7 years of teaching experience. She is working as Assistant Professor of Computer Science in Erode Arts and Science College, Erode, Tamilnadu. Her research interests include Network Security and Data Mining.

Dr. S. Sukumaran graduated in 1985 with a degree in Science. He obtained his Master Degree in Science and M.Phil in Computer Science from the Bharathiar University. He received the Ph.D degree in Computer Science from the Bharathiar University. He has 25 years of teaching experience starting from Lecturer to Associate Professor. At present he is working as Associate Professor of Computer Science in Erode Arts and Science College, Erode, Tamilnadu. He has guided for more than 40 M.Phil research Scholars in various fields and guided one Ph.D Scholar. Currently he is Guiding 5 M.Phil Scholars and 8 Ph.D Scholars. He is member of Board studies of various Autonomous Colleges and Universities. He published around 15 research papers in national and international journals and conferences. His current research interests include Image processing, Network Security and Data Mining.