Utilising fuzzy logic and trend analysis for effective intrusion detection

Abstract

Computer security, and intrusion detection in particular, has become increasingly important in today's business environment, to ensure safe and trusted commerce between business partners as well as effective organizational functioning. Various approaches to intrusion detection are currently being utilized, but unfortunately in practice these approaches are relatively ineffective and inefficient. New means and ways that will minimize these shortcomings must, therefore, continuously be researched and defined. This paper will propose a proactive and dynamic approach, based on trend analysis and fuzzy logic that could be utilized to minimize and control intrusion in an organization's computer system.

Keywords: Computer misuse, Intrusion detection, Fuzzy logic, Pattern recognition.

1 Introduction

The introduction of technologies such as e-commerce will not only increase the value of information, but will also increase security requirements of those organizations that are intending to utilize such technologies. Evidence of these requirements can be seen in the 2001 CSI/FBI Computer Crime and Security Survey (Power, 2001). According to this source the annual financial losses caused through security breaches in 2001 have increased by 277% when compared to the results from 1997. The 2002 Computer Crime and Security Survey confirms this by stating that the threat from computer crime and other information security breaches continues unabated and that the financial toll is mounting (Richardson, 2002).

Information is currently protected through a process of identifying, implementing, managing and maintaining a set of information security controls or countermeasures (GMITS, 1998). Continuous monitoring and detection are a crucial part of this process and it will not only identify the most effective security controls, but it can also be used to implement and manage the identified security controls (Brace, 2000).

Intrusion monitoring and detection can be implemented by an intrusion detection system and today there are many commercial intrusion detection systems available. These commercial implementations are generally restricted in their monitoring functionality (Dowland, 2000) and more research is currently being conducted to improve this functionality.

The focus of this paper is to define and describe a proactive and dynamic fuzzy methodology that can be utilized in an intrusion detection system. The fuzzy methodology is based on the assumption that the intruder's behavior can be grouped into common generic intrusion phases and that all users' actions on the system can be monitored in terms of these phases.

The first part of the paper will provide an overview on intrusion detection systems (IDS) and fuzzy logic. The overview will commence with a discussion on the most important shortcomings of the current intrusion detection systems as well as an intrusion detection strategy that can be used to overcome these problems. The second part of the paper will spell out how fuzzy logic can be utilized to implement the associated intrusion detection strategy. The final part will describe a working prototype for the intrusion detection system.
Martin Botha

Martin Botha is a system engineer in computer based radio networks at the South African Police Service. His research interests include Information Security in general, Intrusion Detection Systems, hacking and GPS networks. He received his B-COM in Information Technology from the University of South Africa, his B-Tech and M-Tech from the Port Elizabeth Technikon in Information Technology and is currently working towards a D-Tech in Information Technology at Port Elizabeth Technikon. Contact him at the SAPS, Radio Technical, Lennox street 1-7, Port Elizabeth, South Africa; bothamar@saps.org.za.

Rossouw von Solms

Rossouw von Solms is a professor in Information Technology at the Port Elizabeth Technikon. He obtained a PhD from the Rand Afrikaanse University in Johannesburg and has been involved with research in information security for more than 10 years. He has published and presented numerous papers internationally on information security. He served for six years as the chairman of IFIP Working Group 11.1, dealing with information security management issues. He can be contacted at: rossouw@petech.ac.za.

strategy and will also discuss the experimental results.

2 Overview of current Intrusion Detection Systems and fuzzy logic

Current Intrusion Detection Systems are based on two major intrusion detection approaches namely, misuse and anomaly intrusion detection (Seleznyov, 2000). Misuse intrusion detection systems detect intrusions that follow well-known patterns of attack (or signatures) that exploit known software vulnerabilities (Kumar and Spafford, 1995; Ilgun and Kemmerer, 1995). These misuse intrusion detection systems include encoded knowledge about poor or unacceptable behavior and directly search for it (Smaha, 1993). The primary limitation of this approach is that it looks only for known weaknesses, and may not be of much use in detecting unknown future intrusions (Seleznyov, 2000). Most commercial intrusion detection systems utilize this approach.

The second approach, anomaly intrusion detection systems, for example IDES (Lunt et al., 1992), is detecting abnormal behavior and reports other irregular behavior as potentially intrusive (Seleznyov, 2001). Thus, anomaly intrusion detection systems are based on the detection of the anomalous behavior or the abnormal use of the computer resources. The main problems with anomaly intrusion detection systems are that they tend to be computationally expensive because several metrics are often maintained that need to be updated against every system activity and they may be gradually trained incorrectly to recognize an intrusive behavior as normal due to insufficient data (Seleznyov, 2000).

Currently, research programs are conducted worldwide with the aim to overcome these shortcomings. One example of such a research program is where a group of researchers developed an immunology approach for Intrusion Detection Systems. This approach combines both the anomaly and the misuse detection approaches and treats a wide range of computer security problems as instances of a problem solved by immunology; that of distinguishing ‘self’ (the body’s own cells and molecules) from ‘other’ (everything else).

Under this analogy, ‘self’ represents the stable operations of a computer, and ‘other’ represents the intrusive (or otherwise anomalous) behavior that must be prevented (Forrest).

Although this approach seems to be an improvement on current approaches, it will still not detect all attacks and it can sometimes identify legitimate behavior as intrusive. Its principle virtues are its low computational overhead and scalability in distributed environments. Thus, it is a simple solution that does not catch everything, but is reasonably low cost to execute (Forrest). The rest of this paper will describe an alternative approach that will aim to be more precise.

2.1 A simple approach

In this section it will be highlighted how one of the shortcomings, that is the lack of precise data, can possibly be overcome. Precise data, in this context, refers to data that can be used accurately to identify an intruder’s behavior such as illegal accessing of a program. Although such data is normally very scarce, some precise data can be found on a computer system and must therefore be used optimally.

This can be achieved by firstly grouping the typical behavior of intruders into a set of generic intrusion phases and secondly to use the available data in terms of every user on the system to (i) predict the possibility that the user is in a process of performing an intrusion attack and (ii) to determine which behavior phase was reached by the user/intruder. Through previous research conducted on generic intrusion phases (Nomad, 1999; Cooper, 1995), it was possible to define the follows phases:
Fuzzy logic and trend analysis for effective intrusion detection

Martin Botha and Rossouw von Solms

- Probing phase
- Initial access phase
- Super-user access phase
- Hacking phase
- Covering phase
- Backdoor phase

Figure 1 is a graphic representation of the simple intrusion detection approach. The figure provides a simplified explanation of the approach and it shows that the approach is based on a linear relationship between the number of events observed for each user on the system and the confidence level that these events might correspond with that of an intruder. In real-life other techniques such as statistical or mathematical methods might be used, but because of the lack of precise data, it was decided to utilize the linear method.

The simple approach consists of six generic intrusion phases and a simple algorithm (precise method). This approach will gather precise data from the firewall and operating system audit logs as well as the various user profiles. For example, if there is evidence in the firewall audit logs that indicates that a user probed for available services, tried to access the firewall illegally and that all these activities were performed after hours, one can predict (according to the graph) that the user is in the process of performing an intrusion attack and that one can couple an intrusion probability value of 33.3% to that intruder. This means that the system administrator can be 33.3% certain that the user is performing an intrusion activity.

2.2 A precise method (algorithm)

The second shortcoming of current anomaly intrusion detection systems is that there is no precise method that can combine the available precise data with other less precise data (pieces of evidence) in a rigorous and consistent manner. This characteristic is vital for developing solutions for real-life problems such as the intrusion detection problem.

The strategy that will be discussed in the rest of this paper does have this vital characteristic. The strategy is based on fuzzy logic. Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth, thus truth values which are between completely true and completely false (Kosko, 1993). The objective of the strategy is to compare (i) the generic intrusion phases (identified in Section 2.1) to (ii) the actions of a user or intruder (in the rest of the paper the person performing the intrusion will be referred to as an intruder). Each of these will be represented by a graph using fuzzy logic. These graphs will then be compared using pattern recognition techniques. The first graph will be called the template and will represent the six generic intrusion phases, and the second graph will represent the actual actions of the user/intruder and will be called the user action graph.
In the next section, it will be explained how real-life input information (precise and less precise data) can be converted into a geometrically shaped graph by using the fuzzification, inference and composition steps of the fuzzy logic method. It will also be explained how this geometrical graph can be used to construct both the user action graph and template, as well as mapping them in order to determine the probability that a user/intruder is carrying out an intrusion activity.

3 The fuzzy methodology

Having defined an intrusion detection fuzzy strategy in the previous section, this part of the paper will spell out in detail how fuzzy logic can be utilized to implement the strategy. In practice, it is not so easy to construct the graphs and to map them onto each other as explained in the previous section. Fortunately, fuzzy logic provides a comprehensive approach that can be used to construct the user action graph and the template, as well as mapping them. Fuzzy logic uses multi-valued logic to model problems that deal with ambiguous data. This section of the paper will only describe the fuzzy methodology. For more information on the various fuzzy concepts and formulas, refer to the reference section.

The approach is based on four steps. The four steps are:

- Fuzzification step
- Inference step
- Composition step
- Defuzzification step (Mapping strategy) (Berkan, 1997)

3.1 Fuzzification step

Fuzzification is normally the first step to be performed when employing fuzzy logic. The objective of this step is to define input variables as well as input membership functions for each input variable (Berkan, 1997). Input variables are mixed data and refer to the mixture of numerical and symbolic (linguistic) data (Berkan, 1997). Suitable variable inputs for the intrusion detection problem can be found by analyzing the generic intrusion phases. Table 1 shows possible input variables (sensors that can measure input data) for each step of the generic intrusion phases. In practice, an extensive set of input variables will be defined and the implementation of the variables can differ from organization to organization.

Eleven input variables are defined in this case that will provide information on the intruder’s actions on the system. As mentioned earlier, an extensive set of input variables will be used in practice. The information gained from the input variables represents real-world values and must be converted to truth-values in order to accurately map the intruder’s actions onto the user action graph. The truth-value indicates the certainty factor that an input variable is true. The conversion from real-life values (precise or less precise data) to truth-values is done by using membership functions. A membership function is a graphic representation of the degree to which each precise (real-world) input variable belongs to a fuzzy set (Kosko, 1993; Kaehler, 1999). The fuzzy set is a set of all the truth-values for that given variable. The degree

<table>
<thead>
<tr>
<th>Generic intrusion phases</th>
<th>Input variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 1: Probing available services</td>
<td>1) Probing command; and 2) Illegal firewall access.</td>
</tr>
<tr>
<td>Phase 2: Gaining initial access</td>
<td>3) Invalid password attempt; 4) User terminal (network address); and 5) User working hours.</td>
</tr>
<tr>
<td>Phase 3: Gaining full system access</td>
<td>6) Illegal password file access attempt; 7) Illegal file/directory access attempt; and 8) Illegal application access.</td>
</tr>
<tr>
<td>Phase 4: Performing the hacking attempt</td>
<td>9) Intruder’s action.</td>
</tr>
<tr>
<td>Phase 5: Covering hacking tracks</td>
<td>10) Audit log access.</td>
</tr>
<tr>
<td>Phase 6: Modifying utilities to ensure future access</td>
<td>11) Creating user account.</td>
</tr>
</tbody>
</table>
of belonging is computed by the membership expression. A membership expression is a mathematical expression that translates the precise input variable values to truth-values (linguistic values) (Kooks, 1993).

The fuzzy set will be used to perform mathematical calculations during the inference, composition and defuzzification steps. The standard notation for expressing a fuzzy set is as follows:

\[ A = \{(x, \mu_A(x))\}, \quad x \in X \]  
1

For input variable 2 (Illegal firewall access) one can define the following membership expression for this input:

\[
\text{Illegal firewall access (x)} = \\
0, \text{if number of attempts } < 3 \\
0.33\%, \text{if number of attempts } = 3 \\
0.66\%, \text{if number of attempts } = 4 \\
1, \text{if number of attempts } > 4
\]

This membership expression can be interpreted as follows:

- One can be 0% certain, if the user or the intruder types his/her password incorrectly zero to two times, that this behavior corresponds to that of a potential intruder.
- One can be 33% certain, if the user or the intruder types his/her password incorrectly three times, that this behavior corresponds to that of a potential intruder.
- One can be 66% certain, if the user or the intruder types his/her password incorrectly four time, that this behavior corresponds to that of a potential intruder.
- One can be 100% certain, if the user or the intruder types his/her password incorrectly five or more times, that this behavior corresponds to that of a potential intruder.

The membership function for illegal firewall access input can be represented by the graph in Figure 2(a. A). This figure shows the fuzzification process where real-world values are translated into truth values. Figure 2(a. B) shows the categorizing process, which transforms these truth-values into standard fuzzy values, which will be discussed in Section 3.2.

The fuzzy set for the membership expression for illegal firewall access is as follows:

\[
A(\text{Illegal firewall access}) = \\
0/0 \cup 0.33/1 \cup 0.66/2 \cup 1/3 \\
\text{(Where 1 = R/W, 2 = F/FTP and 3 = N/A)}
\]

The membership function and fuzzy set of the rest of the input variables are shown in Table 2 and Figure 2 respectively.

\begin{table}[h]
\centering
\begin{tabular}{ |c|c|c| } 
\hline
\textbf{Fuzzy set} & \textbf{Mathematical expression} \\
\hline
A (Probing command) & 0/0 \cup 0.33/1 \cup 0.66/2 \cup 1/3 \\
& (Where 1 = R/W, 2 = F/FTP and 3 = N/A) \\
\hline
A (Invalid password attempt) & 0/3 \cup 0.33/4 \cup 0.66/5 \cup 1/6 \\
\hline
A (User terminal) & 0/0 \cup 0.33/1 \cup 0.66/2 \cup 1/3 \\
\hline
A (User working hours) & 0/7 \cup 0/19 \cup 0.33/6 \cup 0.33/20 \cup 0.66/5 \\
& \cup 0.66/21 \cup 1/4 \cup 1/22 \\
\hline
A (Illegal password file access) & 0/0 \cup 0.33/1 \cup 0.66/2 \cup 1/3 \\
\hline
A (Illegal file/directory access) & 0/0 \cup 0.33/1 \cup 0.66/2 \cup 1/3 \\
\hline
A (Illegal application access) & 0/0 \cup 0.33/1 \cup 0.66/2 \cup 1/3 \\
\hline
A (Hacker’s action) & 0/0 \cup 0.33/1 \cup 0.66/2 \cup 1/3 \\
\hline
A (Audit log access) & 0/0 \cup 0.33/1 \cup 0.66/2 \cup 1/3 \\
\hline
A (Creating user account) & 0/0 \cup 1/1 \\
\hline
\end{tabular}
\caption{Fuzzy sets for the input membership expression.}
\end{table}

3.2 Inference step

The inference step is the second step in the fuzzy logic process. The purpose of the inference process is to categorize each input variable according to standard fuzzy values, such as: low, medium or high. The inference process is based on fuzzy rules. The membership expression for each input variable will be used to define these fuzzy rules. The result of the inference step is known as the 'output' membership function (Kahler, 1999).
Fuzzy logic and trend analysis for effective intrusion detection

Figure 2: The fuzzification step.
The fuzzy rules, output membership function and fuzzy set for the output membership function for the input variable: Illegal firewall access is as follows:

The fuzzy rules for illegal firewall access input variable are as follows:

- **Rule 1**: If the user types his/her password incorrectly zero to two times, then the contribution of this input should be zero.
- **Rule 2**: If the user types his/her password incorrectly three times, then the contribution of this input should be low.
- **Rule 3**: If the user types his/her password incorrectly four times, then the contribution of this input should be medium.
- **Rule 4**: If the user types his/her password incorrectly five or more times, then the contribution of this input should be high.

The output membership function for illegal firewall access input variable can be represented by the categorization graph in Figure 2(a, B). The fuzzy set for the output membership function for illegal firewall access is as follows (see Berkan for more detail):

A (Illegal firewall access) = 0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (13)

The output membership function and fuzzy set for the rest of the input variables are shown in Figure 2 and Table 3.

### 3.3 Composition step

The next step in the fuzzy logic process is the composition step. During the composition step, all 11-output membership functions will be combined. This binding process is performed by truncating the categorization triangles at each fuzzy level and then superimposing the resultant trapezoids over each other to create a new geometrical shape graph (McNeil, 1994).

### 3.4 Defuzzification step

The previous step concluded with the construction of a geometrical graph in terms of the 11 input variables. This step will explain how this geometrical graph can be used to map the user’s/ intruder’s actions onto the six generic intrusion phases.

The mapping strategy consists of three phases, namely:

- **Construction of template graph**
- **Construction of user action graph**
- **Mapping the two graphs**

#### 3.4.1 Construction of the template graph

The template represents an intruder’s typical actions when progressing through all six phases of the generic intrusion phases. Therefore, the template graph can be constructed by maximizing the various output membership functions and then combining them. The output membership functions are maximized when all output membership functions have a

### Table 3: Fuzzy sets for the output membership expression.

<table>
<thead>
<tr>
<th>Fuzzy set</th>
<th>Mathematical expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (Probing command) =</td>
<td>0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (14)</td>
</tr>
<tr>
<td>A (Invalid password attempt) =</td>
<td>0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (15)</td>
</tr>
<tr>
<td>A (User terminal) =</td>
<td>0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (16)</td>
</tr>
<tr>
<td>A (User working hours) =</td>
<td>0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (17)</td>
</tr>
<tr>
<td>A (Illegal password file access) =</td>
<td>0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (18)</td>
</tr>
<tr>
<td>A (Illegal file/directory access) =</td>
<td>0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (19)</td>
</tr>
<tr>
<td>A (Illegal application access) =</td>
<td>0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (20)</td>
</tr>
<tr>
<td>A (Hacker’s action) =</td>
<td>0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (21)</td>
</tr>
<tr>
<td>A (Audit log access) =</td>
<td>0/0 ∪ 0.33/2.75 ∪ 0.66/5.5 ∪ 1/8.34 ∪ 0.66/11.09 ∪ 0.33/13.84 ∪ 0/16.67 ... (22)</td>
</tr>
<tr>
<td>A (Creating user account) =</td>
<td>0/0 ∪ 0.5/4.17 ∪ 1/8.34 ∪ 0.5/12.51 ∪ 0/16.67 ... (23)</td>
</tr>
</tbody>
</table>
The various output membership functions can mathematically be maximized and combined by employing the following expression:

\[ \mu_{\bigcup}(x) = \mu_1(x) \land \mu_2(x) \land \cdots \land \mu_j(x), \quad x \in X \]

\[ \therefore \mu_{\bigcup} (\text{Template}) = \frac{\sum x' A_i}{\sum A_i} \]

Where \( x \) is the global centre of gravity and \( x' \) is the local centre of gravity for the consequent fuzzy set \( S_i \) and can be calculated by the following formula:

\[ x^{'}_i = \frac{\int x f(x)dx}{\int f(x)dx} \]

And \( A_i \) is the local area of the shape.

The COG for the template graph computed by expression 26 and 27 is as follows:

\[ \text{Template graph} = 50 \]

This template graph has a COG value of 50. Since the COG value is the centre of the graph and the probability universe is equal to 100, one needs to multiply the COG value by two in order to determine the probability value. Thus, the COG value of 50 means that one can be 100% (50 x 2 to convert the COG value into probability value) certain that the intruder performed an intrusion attack. To map the intruder’s actions onto the template graph, one merely needs to calculate the COG value of the user action graph and compare it to the
variable value of 50. If the user action graph has a COG value of 50, it means that one can be 100% certain that the intruder has completed all six generic phases and can, according to the methodology, be classified as an intruder. Figure 3 shows a typical example of how the mapping process is conducted. From this figure it is evident that user X only completed certain phases of the set of generic intrusion phases and therefore a COG value of 25.05 was calculated for the user. A template was also constructed for user X and this template has a COG value of 50. Since the two COG values are not the same, it means that the two graphs do not match, therefore, one cannot be 100% certain that the intruder moved through all six phases.

The fuzzy methodology that will implement the algorithm defined in Section 2.2 has been discussed in this section. The fuzzy methodology is based on a mapping strategy that will be used to map the user’s actions onto the generic intrusion phases. The strategy consists of three phases and was discussed briefly in this section. In the next section, a working prototype implementation of the fuzzy methodology will be discussed.

4 HIDS

An important step in creating any new methodology is proper testing. To test the fuzzy methodology thoroughly, it was important to implement a working prototype in a typical real-life environment. A working prototype called Hybrid Intrusion Detection System, also known as HIDS, was developed for this purpose.

HIDS is a software suite written in Visual Basic and Visual C programming languages, which can be installed on either a host computer or a server. The prototype allows for two types of testing environments, that is experimental testing and real-time testing. The experimental testing environment is used by the system administrator for testing historical audit logs and user profile data. HIDS also allows viewing of the different audit logs located on both the client and the server in real-time for investigation purposes.
The system administrator will typically implement HIDS through the real-time testing environment. When selecting this environment, HIDS will implement the fuzzy methodology by running in the background and when identifying an intrusion attack, it will notify the system administrator in real-time.

The testing was conducted by utilizing the real-time test environment. For the initial test, 12 users were asked to participate. Ten of these users were asked to do legal and illegal actions on the system, and two users (User 1 and User 12) were asked to do only normal authorized actions. For each user a unique user profile was set up. Figures 4 and 5 show the result for the actions performed by User 11 (an illegal user). Figure 4 shows the results of the fuzzy engine and it can be interpreted that there is 48% probability that User 11 is performing an intrusion attack. Figure 5 shows the overall results for User 11.

Table 4 shows the results for all 12 users for the fuzzy engine.

In examining the test results for the fuzzy engine in Table 4, it would seem that the fuzzy engine component of HIDS operates relatively well since it correctly identifies the 10 users who performed illegal actions on the system. Perhaps, the most telling result is that for User 11, if taken into account that all possible security controls were set up to prevent users performing intrusive actions. The system calculates a probability value of 48% (24 x 2), which means that HIDS is 48% certain that the user is performing an intrusion attack.

5 Conclusion

A novice fuzzy methodology that will identify the different levels of an intrusion attack has been proposed in this paper. The model will identify the intrusion attack, by reading audit log files and user profiles on the operating system and then by constructing the user graphs according to the information. The methodology will also construct a typical intrusion graph.
(template graph) and it will then map the user graph onto this template graph. If the two graphs match, the methodology will then alert the security officer that someone is carrying out an intrusion attack. If not, the methodology will then compute which phase the intruder reached. As soon as the methodology computes the phase, it will also alert the security officer and notify him/her that someone is performing an intrusion attack and which intrusion phase was reached. Fuzzy logic will be used in both the mapping and phase determining processes.

A prototype called HIDS was developed which implements the fuzzy methodology. Several tests have already been performed which suggested that the methodology is feasible. The final phase will be to test the prototype in a live environment.

Finally, the fuzzy methodology, as outlined in this paper, is a promising area of research that might provide a better quality and more effective intrusion detection. The research will be extended so that it can identify other security aspects, such as corruption of information, disclosure of information, theft of information and denial of services, thus making a definite contribution to the security of information in the 21st century.

<table>
<thead>
<tr>
<th>USER NAME</th>
<th>COG VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>0.00</td>
</tr>
<tr>
<td>Test 2</td>
<td>8</td>
</tr>
<tr>
<td>Test 3</td>
<td>6</td>
</tr>
<tr>
<td>Test 4</td>
<td>6</td>
</tr>
<tr>
<td>Test 5</td>
<td>16</td>
</tr>
<tr>
<td>Test 6</td>
<td>15.03</td>
</tr>
<tr>
<td>Test 7</td>
<td>12.66</td>
</tr>
<tr>
<td>Test 8</td>
<td>15.23</td>
</tr>
<tr>
<td>Test 9</td>
<td>12</td>
</tr>
<tr>
<td>Test 10</td>
<td>14</td>
</tr>
<tr>
<td>Test 11</td>
<td>24</td>
</tr>
<tr>
<td>Test 12</td>
<td>0.00</td>
</tr>
</tbody>
</table>
References


