CHAPTER – FIVE

Prioritization of Detected Intrusion In Biometric Template Storage For Prevention Using Neuro-Fuzzy Approach

Title

5.1 Introduction
5.2 Neuro-Fuzzy concepts
   5.2.1. Concept of Artificial Neural Network
   5.2.2. Concept of Fuzzy Logic
   5.2.3. Overview of Neuro-Fuzzy Logic
5.3 Fuzzy inference engine
5.4 Proposed system
   5.4.1. Architecture of Neuro-Fuzzy design
   5.4.2. Logic used for Fuzzification
   5.4.3. FuzzyJess used for Logic development
   5.4.4. Encoding of the Rules used for Prioritization
5.5 Findings of this Approach
   Concluding Remark
PREVIEW

This Chapter is divided into five primary sections. The first section provides an overview of biometric template storage. The second section describes concept of Artificial neural network, Fuzzy logic and Neuro-Fuzzy approach. The third section provides principle of Fuzzy Inference engine with the FuzzyJess. Fourth section explains the architecture of proposed Neuro-Fuzzy approach for prioritization of detected intrusion at biometric template storage, logic used for Fuzzification, how FuzzyJess used for Logic development and rules used to set priorities. Fifth section explains output of this proposed Neuro-Fuzzy approach which helps security administrator to decide priorities of detected intrusion to take preventive action.

5.1 Introduction

Biometric Templates contain very sensitive information used to identify people which are bound to them. It is the template that is used to determine the user’s rights and privileges to access that resource. The biometric template is stored in smart card, central repository, biometrics device. Attacks on the biometric template storage can lead to the vulnerabilities like insertion of a fake template, modification of an existing template, removal of an existing template, and replicate the template which can be replayed to the matcher to gain unauthorized access. A security administrator requires assistance to prevent those vulnerabilities. In this chapter, authors proposed an intelligent agent which assists to decide the priority for prevention of intrusion in the biometric template storage using Neuro-Fuzzy approach. Authors used FuzzyJess and Java to achieve this prioritization. Priority table is produced as output.
which is useful for security administrator to implement preventive actions for detected intrusion in biometric template storage.

5.2 Neuro-Fuzzy concepts

5.2.1. Concept of Artificial Neural Network

Nowadays there is a new field of computational science that integrates the different methods of problem solving that can not be so easily described without an algorithmic traditional focus. These methods, in one way or another, have their origin in the emulation, more or less intelligent, of the behaviour of the biological system. It is new way of computing denominated Artificial Intelligence, which through different methods is capable of managing the ambiguities and uncertainties that appear when trying to solve problems related to the real world, offering strong solution and of easy implementation. One of these techniques is known as Artificial Neural Networks (ANN). It is like people, learn by example.

Artificial Neural network commonly referred to as neural networks is an adaptive system that changes its structure based on internal and external information that flows through the network. It is an interconnected group of artificial neurons that uses mathematical model or computational model for information processing based on a connectionist approach to computation. It can learn from data but cannot interpret; it is black box to the user.

The ANN has the capability to acquire knowledge from its surroundings by the adaptation of its internal parameters, which is produced as a response to the presence of an external stimulus. The network learns from the examples which are presented to it, and generalizes knowledge from them. The generalization can be interpreted as the property of artificial neural networks to produce an adequate response to unknown stimulus which are related to the acquired knowledge.

Learning or training of an ANN basically consists in the modification of its weight through the application of learning algorithm when a group of pattern is presented.

The leaning algorithm for the design of the neural network is of two types: supervised learning and unsupervised learning.

Supervised learning is a machine learning technique that sets parameters of an artificial neural network from training data. The task of the learning artificial neural
network is to set the value of its parameters to any valid input value, after having seen output value. In supervised training, both the inputs and the outputs are provided. The network then processes the inputs and compares its resulting outputs against the desired outputs, because in fact we will have two of the output of one of the actual and one is required (desired). Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process occurs over and over as the weights are continually tweaked. The set of data which enables the training is called the "training set." During the training of a network the same set of data is processed many times as the connection weights are ever refined.

The other type of training is called unsupervised training. In unsupervised training, the network is provided with inputs not with desired output. The system itself must then decide what features it will use to group the input data. This is also referred as self organization and adoption.

- **Advantages of Neural Networks**
  - A neural network can perform tasks that a linear program can not.
  - When an element of the neural network fails, it can continue without any problem because of its parallel nature.
  - A neural network learns and does not need to be reprogrammed.
  - It can be implemented in any application without any problem.

- **Disadvantages of Neural Networks**
  - A major problem with neural nets is the "Black Box" nature, or rather, the relationships of the weight changes with the input-output behavior during training and use of trained system to generate correct outputs using the weights.
  - The neural network needs training to operate.
  - Requires high processing time for large neural networks.
  - It is difficult, if not impossible, to determine the proper size and structure of a neural net to solve a given problem.

5.2.2. **Concept of Fuzzy Logic**

Fuzzy Logic was initiated in 1965 by Lotfi A. Zadeh, professor for computer science at the University of California in Berkeley. Fuzzy logic creates the ability to mimic the human mind to effectively employ modes of reasoning that are approximate rather
than exact. In traditional hard computing, decisions or actions are based on precision and certainty. It is a multi-valued logic, that allows intermediate values to be defined between conventional evaluations (crisp values) like true/false, yes/no, high/low etc.

In fuzzy logic, knowledge can be captured in terms of rules and linguistic variables. When the rule set adequately process the fuzzy inputs to produce an adequate response, the rule set and the associated definition of the linguistic variables are said to have modelled the underlying structure of the system. Fuzzy logic systems address the imprecision of the input and output variables by defining fuzzy numbers and fuzzy sets that can be expressed in linguistic variables such as 'VERY HIGH', 'HIGH', 'MEDIUM', 'LOW', 'VERY LOW'. A fuzzy system consists of interpretable linguistic rules but they cannot learn.

Fuzzy rules have been advocated as a key tool for expressing pieces of knowledge in "fuzzy logic". A Fuzzy Rule can be defined as a conditional statement of the form:

\[ \text{IF } x \text{ is } A \ldots \text{THEN } y \text{ is } B, \]

where x and y are linguistic variables; and A and B are linguistic values, determined by fuzzy sets on the universe of discourses X and Y, respectively.

Membership function definitions have been developed for fuzzy variables representing each of the features of the network being monitored. Determining or finding input/output membership functions is the first step of the fuzzy logic control process where a fuzzy algorithm categorises the information entering a system and assigns values that represent the degree of membership in those categories. The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response.

The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system.

There are different membership functions associated with each input and output response. Input membership functions themselves can take any form the designer of the system requires triangles, trapezoids, bell curves or any other shape (shown in...
figure 5.2.2.1.) as long as those shapes accurately represent the distribution of information within the system, and as long as a region of transition exists between adjacent membership functions.

When using fuzzy logic, it is often difficult for an expert to provide “good” definitions for the membership functions for the fuzzy variables.

![Fuzzy Membership function shapes.](image)

**Figure 5.2.2.1: Fuzzy Membership function shapes.**

Fuzzy logic is so tolerant that the system will probably work the first time without any twist.

Fuzzy logic works as follows:

- Define control objectives and criteria.
- Determine the input and output relationships and choose a minimum number of variables for input to the fuzzy logic engine.
- Using the rule based structure of fuzzy logic, break the control problem down into a series of IF X AND /OR Y Then Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that are to be processed and the number fuzzy variables associated with each parameter.
• Create fuzzy logic membership functions that defining the meaning (values) of
  Input/output terms used in the rules.
• Create the necessary pre-processing and post-processing fuzzy logic
  routines.
• Test the system, evaluate the results, tune the rules and membership
  functions, and reset until satisfactory results are obtained.

Here, using fuzzy logic authors define some rules that help the security administrator
or system administrator to select the priority-wise detected intrusion to take some
preventive actions.

• Advantages of Fuzzy Logic
  ▪ Fuzzy logic converts complex problems into simpler problems using
    approximate reasoning. The system is described by fuzzy rules and
    membership functions using human type language and linguistic variables.
    Thus, one can effectively use his/her knowledge to describe the system’s
    behavior.
  ▪ A fuzzy logic description can effectively model the uncertainty and
    nonlinearity of a system. Fuzzy logic avoids the complex mathematical
    modeling.
  ▪ Fuzzy logic is easy to implement.
  ▪ Fuzzy logic based solutions are cost effective for a wide range of
    applications when compared to traditional methods.

• Disadvantages of Fuzzy Logic
  ▪ As the system complexity increases, it becomes more challenging to
    determine the correct set of rules and membership functions to describe
    system behavior.
  ▪ In addition, the use of fixed geometric-shaped membership functions in
    fuzzy logic limits system knowledge more in the rule base than in the
    membership function base.
  ▪ Fuzzy logic uses heuristic algorithms for defuzzification, rule evaluation, and
    antecedent processing. Heuristic algorithms can cause problems mainly
    because heuristics do not guarantee satisfactory solutions that operate
    under all possible conditions.
  ▪ Once the rules are determined, they remain fixed in the fuzzy logic
    controller, which is unable to learn Conventional fuzzy logic cannot generate
rules (users cannot write rules) that will meet a pre-specified accuracy. Accuracy is improved only by trial and error.

5.2.3. Overview of Neuro-Fuzzy Logic
Artificial neural network and fuzzy logic are two key technologies that are used to solve various real world problems intelligently. Intelligent combination of these two approaches can exploit their advantages while eliminating their limitations. The first studies of the neuro-fuzzy systems date of the beginning of the 90’s decade, with Jang, Lin and Lee in 1991, Berenji in 1992 and Nauck from 1993, etc. The majority of the first applications were in process control. Gradually, its application spread for all the areas of the knowledge like, data analysis, data classification, imperfections detection and support to decision-making, etc.

While neural networks have strong learning capabilities at the numerical level, it is difficult for the users to understand them at the logic level. Fuzzy logic, on the other hand, has a good capability of interpretability and can also integrate expert’s knowledge. The hybridization of both the paradigms yields the capabilities of learning, good interpretation and incorporating prior knowledge. The combination can be in different forms. Thus, the disadvantages of the fuzzy systems are compensated by the capacities of the neural networks. These techniques are complementary, which justifies its use together.

The simplest form may be the concurrent neuro-fuzzy model, where a fuzzy system and a neural network work separately. The output of one system can be fed as the input of the other system. The cooperative neuro-fuzzy model corresponds to the case that one system is used to adapt the parameters of the other system.

A Neuro-Fuzzy system is a fuzzy system that uses a learning algorithm derived from or inspired by neural network theory to determine its parameters (fuzzy sets and fuzzy rules) by processing data samples. A Neuro-Fuzzy system can be viewed as a 3-layer feed forward neural network. The learning algorithms can learn both fuzzy sets, and fuzzy rules, and can also use prior knowledge.

A neuro-fuzzy system is a fuzzy system, whose parameters are learned by a learning algorithm. It has a neural network architecture constructed from fuzzy reasoning, and can always be interpreted as a system of fuzzy rules. Learning is used to adaptively
adjust the rules in the rule base, and to produce or optimize the Membership functions of a fuzzy system. Structured knowledge is codified as fuzzy rules.

The fuzzy inference system would then be based on the outputs of neuro-fuzzy classifiers, making final decision in decision making system.

5.3 Fuzzy inference engine

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be made, or patterns discerned. Fuzzy inference engine is the brain of fuzzy system. Fuzzy inference systems have been successfully applied in the field such as automatic control, decision analysis, expert systems etc. Because of its multidisciplinary nature, fuzzy inference systems are associated with number of titles, such as fuzzy rule based systems, fuzzy expert systems, fuzzy modelling, fuzzy logic controller and simply fuzzy systems. Neuro-Fuzzy computing optimizes the standard and consequent parameters of fuzzy inference engine using available data. You can implement two types fuzzy inference system namely Mamdani-type and Sugeno-type.

Mamdani’s fuzzy inference method is the most commonly seen fuzzy methodology. Mamdani’s method was among the first control systems built using fuzzy set theory. After the aggregation process, there is a fuzzy set for each output variable that needs defuzzification. It is possible, and in many cases much more efficient, to use a single spike as the output membership functions rather than a distributed fuzzy set. This type of output is sometimes known as a singleton output membership function, and it can be thought of as a pre-defuzzified fuzzy set. It enhances the efficiency of the defuzzification process because it greatly simplifies the computation required by the more general Mamdani method, which finds the centroid of a two-dimensional function.

Sugeno, or Takagi-Sugeno-Kang, method of fuzzy inference is introduced in 1985. It is similar to the Mamdani method in many respects. The first two parts of the fuzzy inference process, fuzzifying the inputs and applying the fuzzy operator, are exactly the same. The main difference between Mamdani and Sugeno is that the Sugeno output membership functions are either linear or constant.
The basic structure of the two-input, one-output, and three-rule example is shown in the following diagram:

Figure 5.2.3.1: Fuzzy inference process

Information flows from left to right, from two inputs to a single output. The parallel nature of the rules is one of the more important aspects of fuzzy logic systems. Instead of sharp switching between modes based on breakpoints, logic flows smoothly from regions where the system's behavior is dominated by either one rule or another.

Fuzzy inference process comprises of five parts:

- **Fuzzification of the input variables**
  This is the process of generating membership values for a fuzzy variable using membership functions. The first step is to take the inputs which is crisp input and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. This crisp input is always a numeric value limited to the universe of discourse. Once the crisp inputs are obtained, they are fuzzified against the appropriate linguistic fuzzy sets.

- **Application of the fuzzy operator (AND or OR) in the antecedent**
  This is the second step where the fuzzified inputs are applied to the antecedents of the fuzzy rules. After the inputs are fuzzified, you know the degree to which each part of the antecedent is satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator i.e. AND, OR is applied to obtain one number that represents the result of the antecedent for that rule. This number is then applied to the output function.
The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth value.

- **Implication from the antecedent to the consequent**
  Before applying the implication method, you must determine the rule's weight. After proper weighting has been assigned to each rule, the implication method is implemented. The consequent is reshaped using a function associated with the antecedent. The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication is implemented for each rule.

- **Aggregation of the consequents across the rules**
  This is the process of unification of the outputs of all rules. Because decisions are based on the testing of all of the rules in a Fuzzy Inference System, the rules must be combined in some manner in order to make a decision. Thus, input of the aggregation process is the list of scaled consequent membership functions, and the output is one fuzzy set for each output variable. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. Aggregation only occurs once for each output variable.

- **Defuzzification**
  The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set.

A fuzzy inference Figure 5.2.3.1 displays all parts of the fuzzy inference process; from fuzzification through defuzzification.

To develop this model, the inference engine makes use of FuzzyJess to evaluate fuzzy logic rules. The inputs to the Fuzzy Inference Engine are Fuzzification of the input Variables i.e. FuzzyVariable in FuzzyJess, The fuzzy rules fired within the FuzzyJess environment and the records, which are asserted as facts in FuzzyJess. FuzzyJess can be configured to use Mamdani or Larsen inference mechanisms to compute the firing strength of each rule applied to each fact. The evaluation of rules begins with the analysis of the antecedent. Rules fire until no more rules match the facts in working memory. Only one rule fires per cycle. The inference engine will
match the facts against fuzzy rules, fire rules and execute the associated actions (Badiru).

5.4 Proposed system

5.4.1. Architecture of Neuro-Fuzzy design

Using this intelligent assistant tool, we got user role (either DBA or normal user), suspicious user name and number of times that user tried for intrusion, suspicious host machine name and number of times that host machine was used for intrusion and data about how many times any user tried transactions like modify existing biometric template, Insert a fake biometric template, delete existing biometric template and copy the biometric template for another use. All these values are already stored in facts and retrieve these values from fact to decide priorities of detected intrusions in biometric template storage for preventive actions.

Prevention decision occur whenever there are more detected intrusions; the exact priority assigned to those detected intrusions is unimportant only relative priority order is important because it determines which actions are the highest priority ones. With preventive actions based on priorities; it helps security administrator or database administrator to secure system from system damage.

i. Identify the four parameters like type of user (DBA or other normal user), Suspicious Host frequency (number of times intrusion made from suspicious host machine), Suspicious User frequency (number of times intrusion made by suspicious user), Type of transaction (intrusion made by using Update, Delete, Insert or Copy).

ii. Classify the parameters USERTYPE and TRANSACTION, both are crisp variables because values are of crisp nature and SUSPICIOUS HOST FREQ and SUSPICIOUS USER FREQ are the fuzzy variables because of uncertainty.

iii. Once the parameters are classified, use fuzzy logic for modelling the uncertain parameters referred as fuzzification. Classify SUSPICIOUS HOST FREQ and SUSPICIOUS USER FREQ fuzzy variables in VeryLow, Low, High, VeryHigh fuzzy values as linguistic expressions. The ranges are decided by automated learning method with the help of algorithm authors design. Authors use RFuzzySet for VeryLow two TriangularFuzzySet for

Architecture of proposed approach is shown in the figure 5.4.1.1. As per literature survey, authors developed more than 128 fuzzy rules to decide priorities for preventive actions.

Figure 5.4.1.1: Proposed neuro-fuzzy approach
5.4.2. Logic used for Fuzzification

Logic used for fuzzification of the input variables. Changing a crisp value to a fuzzy value is often referred to as \textit{fuzzification}.

1. Collect all suspicious frequencies into array.
   
   SUSPICIOUS HOST FREQ is nothing but number of times host machine is used for suspicious activities and suspicious SUSPICIOUS USER FREQ is number of times suspicious user perform suspicious activities.

2. Find out minimum number (min) and maximum number (max) of array.

3. Assume X1 as 0.0.

4. Calculate difference between max and min.

5. Store X2 as difference between max and min.

6. Store X3 as twice the difference between max and min.

7. Store X4 as thrice the difference between max and min.

8. Store X5 as max.

9. Calculate X23 as \((X2+X3)/2\) and X34 as \((X3+X4)/2\)

![Membership functions and linguistic expression](image)

Figure 5.4.2.1: Membership functions and linguistic expression

5.4.3. FuzzyJess used for Logic development

The NRC FuzzyJ Toolkit can be used to create Java programs that encode fuzzy operations and fuzzy reasoning. However, a rule based expert system shell (Jess) provides a convenient and suitable way to encode many types of applications. Fuzzy logic programs fit nicely into the rule based paradigm. An integration of the FuzzyJ Toolkit and Jess is FuzzyJess. As identical fuzzy facts are asserted from different rules the contribution from each rule is accumulated. A fuzzy rule fires in Jess when the fuzzy (and crisp) patterns on the left hand side of the rule match. The fuzzy matching is controlled by the use of the fuzzy-match function.
The fuzzymatch function takes two arguments: either both FuzzyValue objects or a FuzzyValue object and a string that represents a valid fuzzy expression. If one of the arguments is a string then it will be converted to a FuzzyValue using the FuzzyVariable associated with the other FuzzyValue argument. FuzzyVariable objects must both be associated with the same FuzzyVariable so that they can be compared. If there is some degree of match between the two, then the fuzzy-match function returns true, otherwise it returns false.

Authors store fuzzy variables in global variables; the code will look much complicated if we use the import function of Jess since we could just use FuzzyVariable and FuzzyValue etc..

Table 5.4.3.1: JESS implementation of Fuzzy Variable

<table>
<thead>
<tr>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>(defglobal ?<em>shostfrqFvar</em> = (new nrc.fuzzy.FuzzyVariable &quot;shostfrq&quot; 0.0 …))</td>
</tr>
<tr>
<td>(?<em>shostfrqFvar</em> addTerm &quot;VeryLow&quot; (new nrc.fuzzy.RFuzzySet …)</td>
</tr>
<tr>
<td>(?<em>shostfrqFvar</em> addTerm &quot;Low&quot;(new nrc.fuzzy.TriangleFuzzySet…</td>
</tr>
<tr>
<td>(?<em>shostfrqFvar</em> addTerm &quot;High&quot;(new nrc.fuzzy.TriangleFuzzySet…</td>
</tr>
<tr>
<td>(?<em>shostfrqFvar</em> addTerm &quot;VeryHigh&quot; (new nrc.fuzzy.LFuzzySet …</td>
</tr>
</tbody>
</table>

Observe that each rule that contributes to the same fuzzy output does fire independently. Because of this it is necessary to allow all of these rules to fire before the final global conclusion is used. This is done using the salience property in the control rule. By setting it to a value of –100, it will fire only after the other rules have fired. Some have argued that this global assimilation of results should be done automatically by the expert system shell, however, it is not possible to identify the rules that will generate the same fuzzy outputs and have them bumped in priority to execute together.

There are a couple of other things that a user must know to use the FuzzyJess extension. You need to have access to the FuzzyJ Toolkit and FuzzyJess packages (nrc.fuzzy and nrc.fuzzy.jess). Normally these will be in a Java jar file for easy inclusion in the classpath variable. The only other thing that is required is that instead of using a Rete object in programs, you must use a FuzzyRete object.

However when the right hand side of the rule is executed, it is often necessary to know what fuzzy values matched the fuzzy patterns specified in the fuzzy match function calls. In particular, this information is required when a fuzzy fact is being
asserted since the shape of the fuzzy value being asserted depends on the degree of matching of the fuzzy patterns on the right hand side.

### 5.4.4. Encoding of the Rules used for Prioritization

Samples of rule and fuzzy rule are as follows:

If type of user is DBA and suspicious host frequency is in range of very high and suspicious user frequency is in range of very high and transaction is modification then priority of intrusion is very high.

The above rule converted in Jess is

```
(defrule pr1
 ?a1<-((crispval2 ?ut &: (eq ?ut "DBA")))
 ?b1<-((crispval3 ?an &: (eq ?an "UPDATE")))
 ?c1<-((shostf ?t &: (fuzzy-match ?t "VeryHigh")))
 ?d1<-((suserf ?t1 &: (fuzzy-match ?t1 "VeryHigh")))
 =>
 (modify ?"pl"(priority "VeryHigh"))
 (retract ?a1 ?b1 ?c1 ?d1))
```

**Table 5.4.4.1: JESS implementation of Fuzzy Rule for DBA UPDATE action with High Priority**

If type of user is DBA and suspicious host frequency is in range of high and suspicious user frequency is in range of high and transaction is insertion then priority of intrusion is high.

The above rule converted in Jess is

```
(defrule pr15
 ?a15<-((crispval2 ?ut &: (eq ?ut "DBA")))
 ?b15<-((crispval3 ?an &: (eq ?an "INSERT")))
 ?c15<-((shostf ?t &: (fuzzy-match ?t "High")))
 ?d15<-((suserf ?t1 &: (fuzzy-match ?t1 "High")))
 =>
 (modify ?"pl"(priority "High"))
 (retract ?a15 ?b15 ?c15 ?d15));
```

**Table 5.4.4.2: JESS implementation of Fuzzy Rule for DBA INSERT action with High priority**

If type of user is DBA and suspicious host frequency is in range of very high and suspicious user frequency is in range of low and transaction is insertion then priority of intrusion is medium.

The above rule converted in Jess is

```
```
If type of user is Normal User and suspicious host frequency is in range of low and suspicious user frequency is in range of high and transaction is selection then priority of intrusion is low.

The above rule converted in Jess is

```
(defrule pr104
  ?a104<-((crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b104<-((crispval3 ?an &: (eq ?an "SELECT"))
  ?c104<-((shostf ?t &: (fuzzy-match ?t "Low"))
  ?d104<-((suserf ?t1 &: (fuzzy-match ?t1 "High"))
 =>
  (modify ?*pl*(priority "Low"))
  (retract ?a104 ?b104 ?c104 ?d104)));
```

If type of user is Normal User and suspicious host frequency is in range of very low and suspicious user frequency is in range of low and transaction is modification then priority of intrusion is very low.

The above rule converted in Jess is

```
(defrule pr117
  ?a117<-((crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b117<-((crispval3 ?an &:(eq ?an "UPDATE"))
  ?c117<-((shostf ?t &:(fuzzy-match ?t "VeryLow"))
  ?d117<-((suserf ?t1 &:(fuzzy-match ?t1 "Low"))
 =>
  (modify ?*pl*(priority "VeryLow"))
```

If type of user is Normal User and suspicious host frequency is in range of low and suspicious user frequency is in range of high and transaction is selection then priority of intrusion is low.

The above rule converted in Jess is

```
(defrule pr25
  ?a25<-((crispval2 ?ut &:(eq ?ut "DBA"))
  ?b25<-((crispval3 ?an &:(eq ?an "INSERT"))
  ?c25<-((shostf ?t &: (fuzzy-match ?t "VeryHigh"))
  ?d25<-((suserf ?t1 &: (fuzzy-match ?t1 "Low"))
 =>
  (modify ?*pl*(priority "Medium"))
```
Authors have designed 128 rules to decide priorities of detected intrusion at biometric template storage. All rules are mentioned in the last sections of this chapter.

5.5 Findings of this Approach

The output screen shows table which contains column like Priority, type of User, Username, Suspicious User Frequency, Host Name, Suspicious Host Frequency and Transaction type. This table will display as intelligent agent which can be notified by security administrator to implement preventive actions. The priority column shows values like VeryLow, Low, Medium High and VeryHigh. The User Type column shows user is DBA or other normal user. User Name column shows user name of both type of user. The suspicious User Frequency column shows number of times that user performed suspicious transaction; name of suspicious transaction is also displayed in the column User Action. Similarly Suspicious User frequency shows number of times host machine used for suspicious activity; machine-id is also displayed in the column Host Name. Table can be sorted on any column. Resultant screen is shown in figure 5.5.1. As per organization policy, security administrator can implement preventive action using triggers for transactions to block suspicious user or suspicious host.
Figure 5.5.1: Output screen
### Priority Rules and Their JESS implementation

<table>
<thead>
<tr>
<th>Rule</th>
<th>Condition</th>
<th>JESS Implementation</th>
</tr>
</thead>
</table>

Table 5.4.6 Rule 1 - 4 with JESS Implementation
Rule 5

IF USERTYPE IS ‘DBA’ AND
SUSPICIOUS HOST FREQ IS High AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS VeryHigh

JESS
(defrule pr5
?a5<-(crispval2 ?ut &:(eq ?ut "DBA\")
?b5<-(crispval3 ?an&:(eq ?an "UPDATE\")
?c5<-(shostf ?t&(fuzzy-match ?t "High")
?d5<-(suserf ?t1&(fuzzy-match ?t1 "VeryHigh")
=> (modify ?"pl"(priority "VeryHigh") (retract ?a5 ?b5 ?c5 ?d5))

Rule 6

IF USERTYPE IS ‘DBA’ AND
SUSPICIOUS HOST FREQ IS High AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS VeryHigh

JESS
(defrule pr6
?a6<-(crispval2 ?ut &:(eq ?ut "DBA\")
?b6<-(crispval3 ?an&:(eq ?an "DELETE\")
?c6<-(shostf ?t&(fuzzy-match ?t "High")
?d6<-(suserf ?t1&(fuzzy-match ?t1 "VeryHigh")
=> (modify ?"pl"(priority "VeryHigh") (retract ?a6 ?b6 ?c6 ?d6))

Rule 7

IF USERTYPE IS ‘DBA’ AND
SUSPICIOUS HOST FREQ IS High AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS VeryHigh

JESS
(defrule pr7
?a7<-(crispval2 ?ut &:(eq ?ut "DBA\")
?b7<-(crispval3 ?an&:(eq ?an "UPDATE\")
?c7<-(shostf ?t&(fuzzy-match ?t "High")
?d7<-(suserf ?t1&(fuzzy-match ?t1 "High")
=> (modify ?"pl"(priority "VeryHigh") (retract ?a7 ?b7 ?c7 ?d7))

Rule 8

IF USERTYPE IS ‘DBA’ AND
SUSPICIOUS HOST FREQ IS High AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS VeryHigh

JESS
(defrule pr8
?a8<-(crispval2 ?ut &:(eq ?ut "DBA\")
?b8<-(crispval3 ?an&:(eq ?an "DELETE\")
?c8<-(shostf ?t&(fuzzy-match ?t "High")
?d8<-(suserf ?t1&(fuzzy-match ?t1 "High")
=> (modify ?"pl"(priority "VeryHigh") (retract ?a8 ?b8 ?c8 ?d8))

Table 5.4.7 Rule 5 - 8 with JESS Implementation
Rule 9

IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryHigh AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS 'Insertion'
THEN PRIORITY IS High

JESS
(derule pr9
  ?a9<- (crispval2 ?ut & (eql ?ut "DBA"))
  ?b9<- (crispval3 ?an & (eql ?an "INSERT"))
  ?c9<- (shostf ?t & (fuzzy-match ?t "VeryHigh"))
  ?d9<- (suserf ?t1 & (fuzzy-match ?t1 "VeryHigh"))
=> (modify ?*pl*(priority "High")) (retract ?a9 ?b9 ?c9 ?d9))

Rule 10

IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryHigh AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS 'Copy'
THEN PRIORITY IS High

JESS
(derule pr10
  ?a10<- (crispval2 ?ut & (eql ?ut "DBA"))
  ?b10<- (crispval3 ?an & (eql ?an "SELECT"))
  ?c10<- (shostf ?t & (fuzzy-match ?t "VeryHigh"))
  ?d10<- (suserf ?t1 & (fuzzy-match ?t1 "VeryHigh"))
=> (modify ?*pl*(priority "High")) (retract ?a10 ?b10 ?c10 ?d10))

Rule 11

IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryHigh AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS 'Insertion'
THEN PRIORITY IS High

JESS
(derule pr11
  ?a11<- (crispval2 ?ut & (eql ?ut "DBA"))
  ?b11<- (crispval3 ?an & (eql ?an "INSERT"))
  ?c11<- (shostf ?t & (fuzzy-match ?t "VeryHigh"))
  ?d11<- (suserf ?t1 & (fuzzy-match ?t1 "High"))
=> (modify ?*pl*(priority "High")) (retract ?a11 ?b11 ?c11 ?d11))

Rule 12

IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryHigh AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS 'Copy'
THEN PRIORITY IS High

JESS
(derule pr12
  ?a12<- (crispval2 ?ut & (eql ?ut "DBA"))
  ?b12<- (crispval3 ?an & (eql ?an "SELECT"))
  ?c12<- (shostf ?t & (fuzzy-match ?t "VeryHigh"))
  ?d12<- (suserf ?t1 & (fuzzy-match ?t1 "High"))
=> (modify ?*pl*(priority "High")) (retract ?a12 ?b12 ?c12 ?d12))

Table 5.4.8 Rule 9 - 12 with JESS Implementation
<table>
<thead>
<tr>
<th>Rule</th>
<th>Condition</th>
<th>JESS Code</th>
</tr>
</thead>
</table>

Table 5.4.9 Rule 13-16 with JESS Implementation
Rule 17  
**IF** USERTYPE IS ‘**DBA**’ **AND**  
SUSPICIOUS HOST FREQ IS *VeryHigh* **AND**  
SUSPICIOUS USER FREQ IS *Low* **AND**  
TRANSCATION IS ‘**Modification**’  
**THEN** PRIORITY IS *High*

**JESS**
(defrule pr17
 ?a13<-\(\text{crispval2} \ ?\text{ut} \ \&: \ (\text{eq} \ ?\text{ut} \ "DBA")\))
 ?b13<-\(\text{crispval3} \ ?\text{an} \ \&: \ (\text{eq} \ ?\text{an} \ "UPDATE")\))
 ?c13<-\(\text{shostf} \ ?\text{t} \ \&: \ (\text{fuzzy-match} \ ?\text{t} \ "VeryHigh")\))
 ?d13<-\(\text{suserf} \ ?\text{t} \ \&: \ (\text{fuzzy-match} \ ?\text{t} \ "Low")\))
 => (\text{modify} \ ?\text{pl} \ (\text{priority} \ "High") \ \text{retract} \ ?a17 \ ?b17 \ ?c17 \ ?d17))

Rule 18  
**IF** USERTYPE IS ‘**DBA**’ **AND**  
SUSPICIOUS HOST FREQ IS *VeryHigh* **AND**  
SUSPICIOUS USER FREQ IS *Low* **AND**  
TRANSCATION IS ‘**Deletion**’  
**THEN** PRIORITY IS *High*

**JESS**
(defrule pr18
 ?a18<-\(\text{crispval2} \ ?\text{ut} \ \&: \ (\text{eq} \ ?\text{ut} \ "DBA")\))
 ?b18<-\(\text{crispval3} \ ?\text{an} \ \&: \ (\text{eq} \ ?\text{an} \ "DELETE")\))
 ?c18<-\(\text{shostf} \ ?\text{t} \ \&: \ (\text{fuzzy-match} \ ?\text{t} \ "VeryHigh")\))
 ?d18<-\(\text{suserf} \ ?\text{t} \ \&: \ (\text{fuzzy-match} \ ?\text{t} \ "Low")\))
 => (\text{modify} \ ?\text{pl} \ (\text{priority} \ "High") \ \text{retract} \ ?a18 \ ?b18 \ ?c18 \ ?d18))

Rule 19  
**IF** USERTYPE IS ‘**DBA**’ **AND**  
SUSPICIOUS HOST FREQ IS *VeryHigh* **AND**  
SUSPICIOUS USER FREQ IS *VeryLow* **AND**  
TRANSCATION IS ‘**Modification**’  
**THEN** PRIORITY IS *High*

**JESS**
(defrule pr19
 ?a19<-\(\text{crispval2} \ ?\text{ut} \ \&: \ (\text{eq} \ ?\text{ut} \ "DBA")\))
 ?b19<-\(\text{crispval3} \ ?\text{an} \ \&: \ (\text{eq} \ ?\text{an} \ "UPDATE")\))
 ?c19<-\(\text{shostf} \ ?\text{t} \ \&: \ (\text{fuzzy-match} \ ?\text{t} \ "VeryHigh")\))
 ?d19<-\(\text{suserf} \ ?\text{t} \ \&: \ (\text{fuzzy-match} \ ?\text{t} \ "VeryLow")\))
 => (\text{modify} \ ?\text{pl} \ (\text{priority} \ "High") \ \text{retract} \ ?a19 \ ?b19 \ ?c19 \ ?d19))

Rule 20  
**IF** USERTYPE IS ‘**DBA**’ **AND**  
SUSPICIOUS HOST FREQ IS *VeryHigh* **AND**  
SUSPICIOUS USER FREQ IS *VeryLow* **AND**  
TRANSCATION IS ‘**Deletion**’  
**THEN** PRIORITY IS *High*

**JESS**
(defrule pr20
 ?a20<-\(\text{crispval2} \ ?\text{ut} \ \&: \ (\text{eq} \ ?\text{ut} \ "DBA")\))
 ?b20<-\(\text{crispval3} \ ?\text{an} \ \&: \ (\text{eq} \ ?\text{an} \ "DELETE")\))
 ?c20<-\(\text{shostf} \ ?\text{t} \ \&: \ (\text{fuzzy-match} \ ?\text{t} \ "VeryHigh")\))
 ?d20<-\(\text{suserf} \ ?\text{t} \ \&: \ (\text{fuzzy-match} \ ?\text{t} \ "VeryLow")\))
 => (\text{modify} \ ?\text{pl} \ (\text{priority} \ "High") \ \text{retract} \ ?a20 \ ?b20 \ ?c20 \ ?d20))

Table 5.4.10 Rule 17-20 with JESS Implementation
Rule 21
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS High AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS 'Modification'
THEN PRIORITY IS High

JESS
(defrule pr21
?a21<- (crispval2 ?ut & (eq ?ut "DBA"))
?b21<- (crispval3 ?an & (eq ?an "UPDATE"))
?c21<- (shostf ?t & (fuzzy-match ?t "High"))
?d21<- (suserf ?t1 & (fuzzy-match ?t1 "Low"))
=> (modify ?*pl*(priority "High") (retract ?a21 ?b21 ?c21 ?d21))

Rule 22
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS High AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS 'Deletion'
THEN PRIORITY IS High

JESS
(defrule pr22
?a22<- (crispval2 ?ut & (eq ?ut "DBA"))
?b22<- (crispval3 ?an & (eq ?an "DELETE"))
?c22<- (shostf ?t & (fuzzy-match ?t "High"))
?d22<- (suserf ?t1 & (fuzzy-match ?t1 "Low"))
=> (modify ?*pl*(priority "High") (retract ?a22 ?b22 ?c22 ?d22))

Rule 23
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS High AND
SUSPICIOUS USER FREQ IS VeryLow AND
TRANSCATION IS 'Modification'
THEN PRIORITY IS Medium

JESS
(defrule pr23
?a23<- (crispval2 ?ut & (eq ?ut "DBA"))
?b23<- (crispval3 ?an & (eq ?an "UPDATE"))
?c23<- (shostf ?t & (fuzzy-match ?t "High"))
?d23<- (suserf ?t1 & (fuzzy-match ?t1 "VeryLow"))
=> (modify ?*pl*(priority "Medium") (retract ?a23 ?b23 ?c23 ?d23))

Rule 24
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS High AND
SUSPICIOUS USER FREQ IS VeryLow AND
TRANSCATION IS 'Deletion'
THEN PRIORITY IS Medium

JESS
(defrule pr24
?a24<- (crispval2 ?ut & (eq ?ut "DBA"))
?b24<- (crispval3 ?an & (eq ?an "DELETE"))
?c24<- (shostf ?t & (fuzzy-match ?t "High"))
?d24<- (suserf ?t1 & (fuzzy-match ?t1 "VeryLow"))

Table 5.4.4.11 Rule 21 - 24 with JESS Implementation
Rule 25
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryHigh AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS 'Insertion'
THEN PRIORITY IS Medium

JESS
(defrule pr25
  (?a25<-(!crispval2 ?ut &:eq ?ut "$DBA")
  ?b25<-(!crispval3 ?an &:eq ?an "$INSERT")
  ?c25<-(!shostf ?t &:fuzzy-match ?t "$VeryHigh")
  ?d25<-(!suserf ?t1 &:fuzzy-match ?t1 "$Low")
)

Rule 26
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryHigh AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS 'Copy'
THEN PRIORITY IS Medium

JESS
defrule pr26
  (?a26<-(!crispval2 ?ut &:eq ?ut "$DBA")
  ?b26<-(!crispval3 ?an &:eq ?an "$SELECT")
  ?c26<-(!shostf ?t &:fuzzy-match ?t "$VeryHigh")
  ?d26<-(!suserf ?t1 &:fuzzy-match ?t1 "$Low")
  => (modify ?"pl"(priority "$Medium") (retract ?a26 ?b26 ?c26 ?d26))

Rule 27
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryHigh AND
SUSPICIOUS USER FREQ IS VeryLow AND
TRANSCATION IS 'Insertion'
THEN PRIORITY IS Medium

JESS
(defrule pr27
  (?a27<-(!crispval2 ?ut &:eq ?ut "$DBA")
  ?b27<-(!crispval3 ?an &:eq ?an "$INSERT")
  ?c27<-(!shostf ?t &:fuzzy-match ?t "$VeryHigh")
  ?d27<-(!suserf ?t1 &:fuzzy-match ?t1 "$VeryLow")
  => (modify ?"pl"(priority "$Medium") (retract ?a27 ?b27 ?c27 ?d27))
)

Rule 28
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryHigh AND
SUSPICIOUS USER FREQ IS VeryLow AND
TRANSCATION IS 'Copy'
THEN PRIORITY IS Medium

JESS
(defrule pr28
  (?a28<-(!crispval2 ?ut &:eq ?ut "$DBA")
  ?b28<-(!crispval3 ?an &:eq ?an "$SELECT")
  ?c28<-(!shostf ?t &:fuzzy-match ?t "$VeryHigh")
  ?d28<-(!suserf ?t1 &:fuzzy-match ?t1 "$VeryLow")
)

Table 5.4.12 Rule 25 - 28 with JESS Implementation
Rule 29
IF USERTYPE IS 'DBA' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS 'Insertion'
THEN PRIORITY IS Medium

JESS
(defrule pr29
?a29<-\(crispval2 \?ut &:(eq \?ut "DBA")\)
?b29<-\(crispval3 \?an&:(eq \?an "INSERT")\)
?c29<-\(shostf \?t&:(fuzzy-match \?t "High")\)
?d29<-\(suserf \?t1&:(fuzzy-match \?t1 "Low")\)
=> (modify \?"pl"(priority "Medium")) (retract \?a29 \?b29 \?c29 \?d29))

Rule 30
IF USERTYPE IS 'DBA' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS 'Copy'
THEN PRIORITY IS Medium

JESS
(defrule pr30
?a30<-\(crispval2 \?ut &:(eq \?ut "DBA")\)
?b30<-\(crispval3 \?an&:(eq \?an "SELECT")\)
?c30<-\(shostf \?t&:(fuzzy-match \?t "High")\)
?d30<-\(suserf \?t1&:(fuzzy-match \?t1 "Low")\)
=> (modify \?"pl"(priority "Medium")) (retract \?a30 \?b30 \?c30 \?d30))

Rule 31
IF USERTYPE IS 'DBA' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS 'Insertion'
THEN PRIORITY IS Medium

JESS
defrule pr31
?a31<-\(crispval2 \?ut &:(eq \?ut "DBA")\)
?b31<-\(crispval3 \?an&:(eq \?an "INSERT")\)
?c31<-\(shostf \?t&:(fuzzy-match \?t "High")\)
?d31<-\(suserf \?t1&:(fuzzy-match \?t1 "VeryLow")\)
=> (modify \?"pl"(priority "Medium")) (retract \?a31 \?b31 \?c31 \?d31))

Rule 32
IF USERTYPE IS 'DBA' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS 'Copy'
THEN PRIORITY IS Medium

JESS
(defrule pr32
?a32<-\(crispval2 \?ut &:(eq \?ut "DBA")\)
?b32<-\(crispval3 \?an&:(eq \?an "SELECT")\)
?c32<-\(shostf \?t&:(fuzzy-match \?t "High")\)
?d32<-\(suserf \?t1&:(fuzzy-match \?t1 "VeryLow")\)
=> (modify \?"pl"(priority "Medium")) (retract \?a32 \?b32 \?c32 \?d32))

Table 5.4.4.13 Rule 29 - 32 with JESS Implementation
<table>
<thead>
<tr>
<th>Rule 33</th>
<th>IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryHigh AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Modification' THEN PRIORITY IS VeryHigh</th>
</tr>
</thead>
</table>
| JESS   | (deffrule pr33
|        | ?a33<- (crispval2 ?ut & : (eq ?ut "NormalUser")
|        | ?b33<- (crispval3 ?an & : (eq ?an "UPDATE")
|        | ?c33<- (shostf ?t & : (fuzzy-match ?t "VeryHigh")
|        | ?d33<- (suserf ?t1 & : (fuzzy-match ?t1 "VeryHigh")
|        | => (modify **p**l *(priority "VeryHigh") (retract ?a33 ?b33 ?c33 ?d33))
| Rule 34| IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryHigh AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Deletion' THEN PRIORITY IS VeryHigh |
| JESS   | (deffrule pr34
|        | ?a34<- (crispval2 ?ut & : (eq ?ut "NormalUser")
|        | ?b34<- (crispval3 ?an & : (eq ?an "DELETE")
|        | ?c34<- (shostf ?t & : (fuzzy-match ?t "VeryHigh")
|        | ?d34<- (suserf ?t1 & : (fuzzy-match ?t1 "VeryHigh")
|        | => (modify **p**l *(priority "VeryHigh") (retract ?a34 ?b34 ?c34 ?d34))
| Rule 35| IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryHigh AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Modification' THEN PRIORITY IS VeryHigh |
| JESS   | (deffrule pr35
|        | ?a35<- (crispval2 ?ut & : (eq ?ut "NormalUser")
|        | ?b35<- (crispval3 ?an & : (eq ?an "UPDATE")
|        | ?c35<- (shostf ?t & : (fuzzy-match ?t "VeryHigh")
|        | ?d35<- (suserf ?t1 & : (fuzzy-match ?t1 "High")
|        | => (modify **p**l *(priority "VeryHigh") (retract ?a35 ?b35 ?c35 ?d35))
| Rule 36| IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryHigh AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Deletion' THEN PRIORITY IS VeryHigh |
| JESS   | (deffrule pr36
|        | ?a36<- (crispval2 ?ut & : (eq ?ut "NormalUser")
|        | ?b36<- (crispval3 ?an & : (eq ?an "DELETE")
|        | ?c36<- (shostf ?t & : (fuzzy-match ?t "VeryHigh")
|        | ?d36<- (suserf ?t1 & : (fuzzy-match ?t1 "High")
|        | => (modify **p**l *(priority "VeryHigh") (retract ?a36 ?b36 ?c36 ?d36))

Table 5.4.14 Rule 33 - 36 with JESS Implementation
Rule 37  IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Modification' THEN PRIORITY IS VeryHigh

JESS (defrule pr37
?a37<- (crispval2 ?ut &:(eq ?ut \"NormalUser\")
?b37<- (crispval3 ?an &:(eq ?an \"UPDATE\")
?c37<- (shostf ?t&(fuzzy-match ?t \"High\")
?d37<- (suserf ?t1&(fuzzy-match ?t1 \"VeryHigh\")
=> (modify ?*pl*(priority \"VeryHigh\") (retract ?a37 ?b37 ?c37 ?d37))

Rule 38  IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Deletion' THEN PRIORITY IS VeryHigh

JESS (defrule pr38
?a38<- (crispval2 ?ut &:(eq ?ut \"NormalUser\")
?b38<- (crispval3 ?an &:(eq ?an \"DELETE\")
?c38<- (shostf ?t&(fuzzy-match ?t \"High\")
?d38<- (suserf ?t1&(fuzzy-match ?t1 \"VeryHigh\")
=> (modify ?*pl*(priority \"VeryHigh\") (retract ?a38 ?b38 ?c38 ?d38))

Rule 39  IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Modification' THEN PRIORITY IS High

JESS (defrule pr39
?a39<- (crispval2 ?ut &:(eq ?ut \"NormalUser\")
?b39<- (crispval3 ?an &:(eq ?an \"UPDATE\")
?c39<- (shostf ?t&(fuzzy-match ?t \"High\")
?d39<- (suserf ?t1&(fuzzy-match ?t1 \"High\")

Rule 40  IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Deletion' THEN PRIORITY IS High

JESS (defrule pr40
?a40<- (crispval2 ?ut &:(eq ?ut \"NormalUser\")
?b40<- (crispval3 ?an &:(eq ?an \"DELETE\")
?c40<- (shostf ?t&(fuzzy-match ?t \"High\")
?d40<- (suserf ?t1&(fuzzy-match ?t1 \"High\")
=> (modify ?*pl*(priority \"High\") (retract ?a40 ?b40 ?c40 ?d40))

Table 5.4.15 Rule 37 - 40 with JESS Implementation
Rule 41  
IF USERTYPE IS 'NormalUser' AND  
SUSPICIOUS HOST FREQ IS VeryHigh AND  
SUSPICIOUS USER FREQ IS VeryHigh AND  
TRANSCATION IS 'Insertion'  
THEN PRIORITY IS High

JESS
(defrule pr41
  (?a41<-crispval2 ?ut &:eq ?ut "NormalUser")  
  (?b41<-crispval3 ?an &:eq ?an "INSERT")  
  (?c41<-shostf ?t &:fuzzy-match ?t "VeryHigh")  
  (?d41<-suserf ?t1 &:fuzzy-match ?t1 "VeryHigh")  
=> (modify ?*pl*(priority "High"))  
(retract ?a41 ?b41 ?c41 ?d41))

Rule 42  
IF USERTYPE IS 'NormalUser' AND  
SUSPICIOUS HOST FREQ IS VeryHigh AND  
SUSPICIOUS USER FREQ IS VeryHigh AND  
TRANSCATION IS 'Copy'  
THEN PRIORITY IS High

JESS
(defrule pr42
  (?a42<-crispval2 ?ut &:eq ?ut "NormalUser")  
  (?b42<-crispval3 ?an &:eq ?an "SELECT")  
  (?c42<-shostf ?t &:fuzzy-match ?t "VeryHigh")  
  (?d42<-suserf ?t1 &:fuzzy-match ?t1 "VeryHigh")  
=> (modify ?*pl*(priority "High"))  
(retract ?a42 ?b42 ?c42 ?d42))

Rule 43  
IF USERTYPE IS 'NormalUser' AND  
SUSPICIOUS HOST FREQ IS VeryHigh AND  
SUSPICIOUS USER FREQ IS High AND  
TRANSCATION IS 'Insertion'  
THEN PRIORITY IS High

JESS
(defrule pr43
  (?a43<-crispval2 ?ut &:eq ?ut "NormalUser")  
  (?b43<-crispval3 ?an &:eq ?an "INSERT")  
  (?c43<-shostf ?t &:fuzzy-match ?t "VeryHigh")  
  (?d43<-suserf ?t1 &:fuzzy-match ?t1 "High")  
=> (modify ?*pl*(priority "High"))  
(retract ?a43 ?b43 ?c43 ?d43))

Rule 44  
IF USERTYPE IS 'NormalUser' AND  
SUSPICIOUS HOST FREQ IS VeryHigh AND  
SUSPICIOUS USER FREQ IS High AND  
TRANSCATION IS 'Copy'  
THEN PRIORITY IS High

JESS
(defrule pr44
  (?a44<-crispval2 ?ut &:eq ?ut "NormalUser")  
  (?b44<-crispval3 ?an &:eq ?an "SELECT")  
  (?c44<-shostf ?t &:fuzzy-match ?t "VeryHigh")  
  (?d44<-suserf ?t1 &:fuzzy-match ?t1 "High")  
=> (modify ?*pl*(priority "High"))  
(retract ?a44 ?b44 ?c44 ?d44))

Table 5.4.16 Rule 41 - 44 with JESS Implementation
<table>
<thead>
<tr>
<th>Rule</th>
<th>IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Insertion' THEN PRIORITY IS High</th>
</tr>
</thead>
</table>
| JESS | (defrule pr45  
?a45<-((crispval2 ?ut &:(eq ?ut "NormalUser"))  
?b45<-((crispval3 ?an&:(eq ?an "INSERT")))  
?c45<-((shostf ?t&(fuzzy-match ?t "High"))  
?d45<-((suserf ?t1&(fuzzy-match ?t1 "VeryHigh")))  
=> (modify ?"p"*(priority "High"))  (retract ?a45 ?b45 ?c45 ?d45))  |
| Rule 46 | IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Copy' THEN PRIORITY IS High |
| JESS | (defrule pr46  
?a46<-((crispval2 ?ut &:(eq ?ut "NormalUser"))  
?b46<-((crispval3 ?an&:(eq ?an "SELECT")))  
?c46<-((shostf ?t&(fuzzy-match ?t "High"))  
?d46<-((suserf ?t1&(fuzzy-match ?t1 "VeryHigh")))  
=> (modify ?"p"*(priority "High"))  (retract ?a46 ?b46 ?c46 ?d46))  |
| Rule 47 | IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Insertion' THEN PRIORITY IS High |
| JESS | (defrule pr47  
?a47<-((crispval2 ?ut &:(eq ?ut "NormalUser"))  
?b47<-((crispval3 ?an&:(eq ?an "INSERT")))  
?c47<-((shostf ?t&(fuzzy-match ?t "High"))  
?d47<-((suserf ?t1&(fuzzy-match ?t1 "High")))  
=> (modify ?"p"*(priority "High"))  (retract ?a47 ?b47 ?c47 ?d47))  |
| Rule 48 | IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Copy' THEN PRIORITY IS High |
| JESS | (defrule pr48  
?a48<-((crispval2 ?ut &:(eq ?ut "NormalUser"))  
?b48<-((crispval3 ?an&:(eq ?an "SELECT")))  
?c48<-((shostf ?t&(fuzzy-match ?t "High"))  
?d48<-((suserf ?t1&(fuzzy-match ?t1 "High")))  

Table 5.4.17 Rule 45 - 48 with JESS Implementation
Rule 49
IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryHigh AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS 'Modification'
THEN PRIORITY IS Medium

JESS
defrule pr49
?a49<-crispval2 ?ut &:eq ?ut "NormalUser")
?b49<-crispval3 ?an &:eq ?an "UPDATE")
?c49<-shostf ?t&:(fuzzy-match ?t "VeryHigh")
?d49<-suserf ?t1&:(fuzzy-match ?t1 "Low")
=> (modify ?"pl"(priority "Medium")) (retract ?a49 ?b49 ?c49 ?d49))

Rule 50
IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryHigh AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS 'Deletion'
THEN PRIORITY IS Medium

JESS
(defrule pr50
?a50<-crispval2 ?ut &:eq ?ut "NormalUser")
?b50<-crispval3 ?an &:eq ?an "DELETE")
?c50<-shostf ?t&:(fuzzy-match ?t "VeryHigh")
?d50<-suserf ?t1&:(fuzzy-match ?t1 "Low")
=> (modify ?"pl"(priority "Medium")) (retract ?a50 ?b50 ?c50 ?d50))

Rule 51
IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryHigh AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS 'Modification'
THEN PRIORITY IS Medium

JESS
(defrule pr51
?a51<-crispval2 ?ut &:eq ?ut "NormalUser")
?b51<-crispval3 ?an &:eq ?an "UPDATE")
?c51<-shostf ?t&:(fuzzy-match ?t "VeryHigh")
?d51<-suserf ?t1&:(fuzzy-match ?t1 "VeryLow")
=> (modify ?"pl"(priority "Medium")) (retract ?a51 ?b51 ?c51 ?d51))

Rule 52
IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryHigh AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS 'Deletion'
THEN PRIORITY IS Medium

JESS
(defrule pr52
?a52<-crispval2 ?ut &:eq ?ut "NormalUser")
?b52<-crispval3 ?an &:eq ?an "DELETE")
?c52<-shostf ?t&:(fuzzy-match ?t "VeryHigh")
?d52<-suserf ?t1&:(fuzzy-match ?t1 "VeryLow")
" => (modify ?"pl"(priority "Medium")) (retract ?a52 ?b52 ?c52 ?d52 ))

Table 5.4.18 Rule 49 - 52 with JESS Implementation
| Rule 53 | IF USERTYPE IS ‘NormalUser’ AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS ‘Modification’ THEN PRIORITY IS Medium |
| Rule 54 | IF USERTYPE IS ‘NormalUser’ AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS ‘Deletion’ THEN PRIORITY IS Medium |
| Rule 55 | IF USERTYPE IS ‘NormalUser’ AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS ‘Modification’ THEN PRIORITY IS Medium |
| Rule 56 | IF USERTYPE IS ‘NormalUser’ AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS ‘Deletion’ THEN PRIORITY IS Medium |

Table 5.4.4.19 Rule 53 - 56 with JESS Implementation
Rule 57  
IF USERTYPE IS 'NormalUser' AND 
SUSPICIOUS HOST FREQ IS VeryHigh AND  
SUSPICIOUS USER FREQ IS Low AND  
TRANSCATION IS 'Insertion' 
THEN PRIORITY IS Low

JESS  
(defrule pr57  
?a57<- (crispval2 ?ut &: (eq ?ut "NormalUser"))  
?b57<- (crispval3 ?an &: (eq ?an "INSERT"))  
?c57<- (shostf ?t &: (fuzzy-match ?t "VeryHigh"))  
?d57<- (suserf ?t1 &: (fuzzy-match ?t1 "Low"))  
=> (modify ?pl* (priority "Low")) (retract ?a57 ?b57 ?c57 ?d57))

Rule 58  
IF USERTYPE IS 'NormalUser' AND 
SUSPICIOUS HOST FREQ IS VeryHigh AND  
SUSPICIOUS USER FREQ IS Low AND  
TRANSCATION IS 'Copy' 
THEN PRIORITY IS Low

JESS  
(defrule pr58  
?a58<- (crispval2 ?ut &: (eq ?ut "NormalUser"))  
?b58<- (crispval3 ?an &: (eq ?an "SELECT"))  
?c58<- (shostf ?t &: (fuzzy-match ?t "VeryHigh"))  
?d58<- (suserf ?t1 &: (fuzzy-match ?t1 "Low"))  
=> (modify ?pl* (priority "Low")) (retract ?a58 ?b58 ?c58 ?d58))

Rule 59  
IF USERTYPE IS 'NormalUser' AND 
SUSPICIOUS HOST FREQ IS VeryHigh AND  
SUSPICIOUS USER FREQ IS VeryLow AND  
TRANSCATION IS 'Insertion' 
THEN PRIORITY IS Low

JESS  
(defrule pr59  
?a59<- (crispval2 ?ut &: (eq ?ut "NormalUser"))  
?b59<- (crispval3 ?an &: (eq ?an "INSERT"))  
?c59<- (shostf ?t &: (fuzzy-match ?t "VeryHigh"))  
?d59<- (suserf ?t1 &: (fuzzy-match ?t1 "VeryLow"))  

Rule 60  
IF USERTYPE IS 'NormalUser' AND 
SUSPICIOUS HOST FREQ IS VeryHigh AND  
SUSPICIOUS USER FREQ IS VeryLow AND  
TRANSCATION IS 'Copy' 
THEN PRIORITY IS Low

JESS  
(defrule pr60  
?a60<- (crispval2 ?ut &: (eq ?ut "NormalUser"))  
?b60<- (crispval3 ?an &: (eq ?an "SELECT"))  
?c60<- (shostf ?t &: (fuzzy-match ?t "VeryHigh"))  
?d60<- (suserf ?t1 &: (fuzzy-match ?t1 "VeryLow"))  
=> (modify ?pl* (priority "Low")) (retract ?a60 ?b60 ?c60 ?d60))

Table 5.4.4.20 Rule 57 - 60 with JESS Implementation
Rule 61

IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS 'Insertion'
THEN PRIORITY IS Low

JESS
(defrule pr61
?a61<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
?b61<- (crispval3 ?an&:(eq ?an "INSERT"))
?c61<- (shostf ?t&(fuzzy-match ?t "High"))
?d61<- (suserf ?t1&(fuzzy-match ?t1 "Low"))
=> (modify ?"pl"(priority "Low")) (retract ?a61 ?b61 ?c61 ?d61))

Rule 62

IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS 'Copy'
THEN PRIORITY IS Low

JESS
(defrule pr62
?a62<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
?b62<- (crispval3 ?an&:(eq ?an "SELECT"))
?c62<- (shostf ?t&(fuzzy-match ?t "High"))
?d62<- (suserf ?t1&(fuzzy-match ?t1 "Low"))

Rule 63

IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS 'Insertion'
THEN PRIORITY IS VeryLow

JESS
(defrule pr63
?a63<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
?b63<- (crispval3 ?an&:(eq ?an "INSERT"))
?c63<- (shostf ?t&(fuzzy-match ?t "High"))
?d63<- (suserf ?t1&(fuzzy-match ?t1 "VeryLow"))
=> (modify ?"pl"(priority "VeryLow")) (retract ?a63 ?b63 ?c63 ?d63))

Rule 64

IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS High AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS 'Copy'
THEN PRIORITY IS VERY Low

JESS
(defrule pr64
?a64<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
?b64<- (crispval3 ?an&:(eq ?an "SELECT"))
?c64<- (shostf ?t&(fuzzy-match ?t "High"))
?d64<- (suserf ?t1&(fuzzy-match ?t1 "VeryLow"))
=> (modify ?"pl"(priority "VeryLow")) (retract ?a64 ?b64 ?c64 ?d64))

Table 5.4.4.21 Rule 61 - 64 with JESS Implementation
Table 5.4.22 Rule 65 - 68 with JESS Implementation
Rule 69  
**IF USER TYPE IS 'DBA AND SUSPICIOUS HOST FREQ IS **Low** AND SUSPICIOUS USER FREQ IS **Low** AND TRANSACTION IS 'Modification' THEN PRIORITY IS Low**

**JESS**

eval("(defrule pr69
?a69<-crispval2 ?ut &:((eq ?ut "DBA\"))
?b69<-crispval3 ?an &:((eq ?an "UPDATE\"))
?c69<-shostf ?t &((fuzzy-match ?t \"Low\"))
?d69<-suserf ?t1 &((fuzzy-match ?t1 \"Low\"))
=> (modify ?*pl*(priority \"Low\")) (retract ?a69 ?b69 ?c69 ?d69))")

Rule 70  
**IF USER TYPE IS 'DBA AND SUSPICIOUS HOST FREQ IS **Low** AND SUSPICIOUS USER FREQ IS **Low** AND TRANSACTION IS 'Deletion' THEN PRIORITY IS Low**

**JESS**

eval("(defrule pr70
?a70<-crispval2 ?ut &:((eq ?ut "DBA\"))
?b70<-crispval3 ?an &:((eq ?an "DELETE\"))
?c70<-shostf ?t &((fuzzy-match ?t \"Low\"))
?d70<-suserf ?t1 &((fuzzy-match ?t1 \"Low\"))
=> (modify ?*pl*(priority \"Low\")) (retract ?a70 ?b70 ?c70 ?d70))")

Rule 71  
**IF USER TYPE IS 'DBA AND SUSPICIOUS HOST FREQ IS **Low** AND SUSPICIOUS USER FREQ IS **VeryLow** AND TRANSACTION IS 'Modification' THEN PRIORITY IS Low**

**JESS**

(defrule pr71
?a71<-crispval2 ?ut &:((eq ?ut "DBA\"))
?b71<-crispval3 ?an &:((eq ?an "UPDATE\"))
?c71<-shostf ?t &((fuzzy-match ?t \"Low\"))
?d71<-suserf ?t1 &((fuzzy-match ?t1 \"VeryLow\"))
=> (modify ?*pl*(priority \"Low\")) (retract ?a71 ?b71 ?c71 ?d71))")

Rule 72  
**IF USER TYPE IS 'DBA AND SUSPICIOUS HOST FREQ IS **Low** AND SUSPICIOUS USER FREQ IS **VeryLow** AND TRANSACTION IS 'Deletion' THEN PRIORITY IS Low**

**JESS**

defrule pr72
?a72<-crispval2 ?ut &:((eq ?ut "DBA\"))
?b72<-crispval3 ?an &:((eq ?an "DELETE\"))
?c72<-shostf ?t &((fuzzy-match ?t \"Low\"))
?d72<-suserf ?t1 &((fuzzy-match ?t1 \"VeryLow\"))
=> (modify ?*pl*(priority \"Low\")) (retract ?a72 ?b72 ?c72 ?d72))")

Table 5.4.4.23 Rule 69 - 72 with JESS Implementation
Rule 73
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS 'Modification'
THEN PRIORITY IS Medium

JESS
(defrule pr73
  (?a73<- (crispval2 ?ut &:(eq ?ut "DBA"))
  ?b73<- (crispval3 ?an&:(eq ?an "UPDATE"))
  ?c73<- (shostf ?t&:(fuzzy-match ?t "VeryLow"))
  ?d73<- (suserf ?t1&(fuzzy-match ?t1 "VeryHigh"))
  => (modify "pl"(priority "Medium") (retract ?a73 ?b73 ?c73 ?d73))
)

Rule 74
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS 'Deletion'
THEN PRIORITY IS Medium

JESS
(defrule pr74
  (?a74<- (crispval2 ?ut &:(eq ?ut "DBA"))
  ?b74<- (crispval3 ?an&:(eq ?an "DELETE"))
  ?c74<- (shostf ?t&:(fuzzy-match ?t "VeryLow"))
  ?d74<- (suserf ?t1&(fuzzy-match ?t1 "VeryHigh"))
  => (modify "pl"(priority "Medium") (retract ?a74 ?b74 ?c74 ?d74))
)

Rule 75
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS 'Modification'
THEN PRIORITY IS Medium

JESS
(defrule pr75
  (?a75<- (crispval2 ?ut &:(eq ?ut "DBA"))
  ?b75<- (crispval3 ?an&:(eq ?an "UPDATE"))
  ?c75<- (shostf ?t&:(fuzzy-match ?t "VeryLow"))
  ?d75<- (suserf ?t1&(fuzzy-match ?t1 "High"))
  => (modify "pl"(priority "Medium") (retract ?a75 ?b75 ?c75 ?d75))
)

Rule 76
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS 'Deletion'
THEN PRIORITY IS Medium

JESS
(defrule pr76
  (?a76<- (crispval2 ?ut &:(eq ?ut "DBA"))
  ?b76<- (crispval3 ?an&:(eq ?an "DELETE"))
  ?c76<- (shostf ?t&:(fuzzy-match ?t "VeryLow"))
  ?d76<- (suserf ?t1&(fuzzy-match ?t1 "High"))
  => (modify "pl"(priority "Medium") (retract ?a76 ?b76 ?c76 ?d76))
)

Table 5.4.4.24 Rule 73 - 76 with JESS Implementation
Rule 77

IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS Medium

JESS
(defrule pr77
?a77<-(crispval2 ?ut &: (eq ?ut "NormalUser"))
?b77<-(crispval3 ?an &: (eq ?an "UPDATE"))
?c77<-(shostf ?t &:(fuzzy-match ?t "Low"))
?d77<-(suserf ?t1 &:(fuzzy-match ?t1 "VeryHigh"))
=> (modify ?*pl*(priority "Medium")) (retract ?a77 ?b77 ?c77 ?d77))

Rule 78

IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS Medium

JESS
(defrule pr78
?a78<-(crispval2 ?ut &: (eq ?ut "NormalUser"))
?b78<-(crispval3 ?an &: (eq ?an "DELETE"))
?c78<-(shostf ?t &:(fuzzy-match ?t "Low"))
?d78<-(suserf ?t1 &:(fuzzy-match ?t1 "VeryHigh"))
=> (modify ?*pl*(priority "Medium")) (retract ?a78 ?b78 ?c78 ?d78))

Rule 79

IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS Medium

JESS
(defrule pr79
?a79<-(crispval2 ?ut &: (eq ?ut "NormalUser"))
?b79<-(crispval3 ?an &: (eq ?an "UPDATE"))
?c79<-(shostf ?t &:(fuzzy-match ?t "Low"))
?d79<-(suserf ?t1 &:(fuzzy-match ?t1 "High"))
=> (modify ?*pl*(priority "Medium")) (retract ?a79 ?b79 ?c79 ?d79))

Rule 80

IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS Medium

JESS
(defrule pr80
?a80<-(crispval2 ?ut &: (eq ?ut "NormalUser"))
?b80<-(crispval3 ?an &: (eq ?an "DELETE"))
?c80<-(shostf ?t &:(fuzzy-match ?t "Low"))
?d80<-(suserf ?t1 &:(fuzzy-match ?t1 "High"))
=> (modify ?*pl*(priority "Medium")) (retract ?a80 ?b80 ?c80 ?d80))

Table 5.4.25 Rule 77 - 80 with JESS Implementation
Rule 81
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSACTION IS 'Insertion'
THEN PRIORITY IS Medium

JESS
(defrule pr81
?a81<-((crispval2 ?ut &:(eq ?ut "DBA"))
?b81<-((crispval3 ?an&:(eq ?an "INSERT"))
?c81<-((shostf ?t&:(fuzzy-match ?t "Low"))
?d81<-((suserf ?t1&:(fuzzy-match ?t1 "VeryHigh"))
=> (modify ?"pl"(priority "Medium")) (retract ?a81 ?b81 ?c81 ?d81))

Rule 82
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSACTION IS 'Copy'
THEN PRIORITY IS Medium

JESS
(de правило pr82
?a82<-((crispval2 ?ut &:(eq ?ut "DBA"))
?b82<-((crispval3 ?an&:(eq ?an "SELECT"))
?c82<-((shostf ?t&:(fuzzy-match ?t "Low"))
?d82<-((suserf ?t1&:(fuzzy-match ?t1 "VeryHigh"))
=> (modify ?"pl"(priority "Medium")) (retract ?a82 ?b82 ?c82 ?d82))

Rule 83
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS High AND
TRANSACTION IS 'Insertion'
THEN PRIORITY IS VeryLow

JESS
(defrule pr83
?a83<-((crispval2 ?ut &:(eq ?ut "DBA"))
?b83<-((crispval3 ?an&:(eq ?an "INSERT"))
?c83<-((shostf ?t&:(fuzzy-match ?t "Low"))
?d83<-((suserf ?t1&:(fuzzy-match ?t1 "High"))
=> (modify ?"pl"(priority "VeryLow")) (retract ?a83 ?b83 ?c83 ?d83))

Rule 84
IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS High AND
TRANSACTION IS 'Copy'
THEN PRIORITY IS VeryLow

JESS
(defrule pr84
?a84<-((crispval2 ?ut &:(eq ?ut "DBA"))
?b84<-((crispval3 ?an&:(eq ?an "SELECT"))
?c84<-((shostf ?t&:(fuzzy-match ?t "Low"))
?d84<-((suserf ?t1&:(fuzzy-match ?t1 "High"))
=> (modify ?"pl"(priority "VeryLow")) (retract ?a84 ?b84 ?c84 ?d84))

Table 5.4.26 Rule 81 - 84 with JESS Implementation
Rule 85

IF USERTYPE IS 'DBA' AND 
SUSPICIOUS HOST FREQ IS VeryLow AND 
SUSPICIOUS USER FREQ IS VeryHigh AND 
TRANSCATION IS 'Insertion'

THEN PRIORITY IS Low

JESS

defrule pr85

?a85<-crispval2 ?ut &:(eq ?ut "DBA")
?b85<-crispval3 ?an&:(eq ?an "INSERT")
?c85<-shostf ?t&:(fuzzy-match ?t "VeryLow")
?d85<-suserf ?t1&:(fuzzy-match ?t1 "VeryHigh")
=> (modify ?*pl*(priority "Low")) (retract ?a85 ?b85 ?c85 ?d85)

Rule 86

IF USERTYPE IS 'DBA' AND 
SUSPICIOUS HOST FREQ IS VeryLow AND 
SUSPICIOUS USER FREQ IS VeryHigh AND 
TRANSCATION IS 'Copy'

THEN PRIORITY IS Low

JESS

defrule pr86

?a86<-crispval2 ?ut &:(eq ?ut "DBA")
?b86<-crispval3 ?an&:(eq ?an "SELECT")
?c86<-shostf ?t&:(fuzzy-match ?t "VeryLow")
?d86<-suserf ?t1&:(fuzzy-match ?t1 "VeryHigh")
=> (modify ?*pl*(priority "Low")) (retract ?a86 ?b86 ?c86 ?d86)

Rule 87

IF USERTYPE IS 'DBA' AND 
SUSPICIOUS HOST FREQ IS VeryLow AND 
SUSPICIOUS USER FREQ IS High AND 
TRANSCATION IS 'Insertion'

THEN PRIORITY IS Low

JESS

defrule pr87

?a87<-crispval2 ?ut &:(eq ?ut "DBA")
?b87<-crispval3 ?an&:(eq ?an "INSERT")
?c87<-shostf ?t&:(fuzzy-match ?t "VeryLow")
?d87<-suserf ?t1&:(fuzzy-match ?t1 "High")
=> (modify ?*pl*(priority "Low")) (retract ?a87 ?b87 ?c87 ?d87)

Rule 88

IF USERTYPE IS 'DBA' AND 
SUSPICIOUS HOST FREQ IS VeryLow AND 
SUSPICIOUS USER FREQ IS High AND 
TRANSCATION IS 'Copy'

THEN PRIORITY IS Low

JESS

defrule pr88

?a88<-crispval2 ?ut &:(eq ?ut "DBA")
?b88<-crispval3 ?an&:(eq ?an "SELECT")
?c88<-shostf ?t&:(fuzzy-match ?t "VeryLow")
?d88<-suserf ?t1&:(fuzzy-match ?t1 "High")
=> (modify ?*pl*(priority "Low")) (retract ?a88 ?b88 ?c88 ?d88)

Table 5.4.27 Rule 85 - 88 with JESS Implementation
<table>
<thead>
<tr>
<th>Rule</th>
<th>Condition</th>
<th>Action</th>
</tr>
</thead>
</table>
| 89   | USERTYPE IS 'DBA' AND SUSPICIOUS HOST FREQ IS Low AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS 'Insertion' THEN PRIORITY IS VeryLow | JESS (deffrule pr89)
|      |           |        |
|      |           |        |
| 90   | USERTYPE IS 'DBA' AND SUSPICIOUS HOST FREQ IS Low AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS 'Copy' THEN PRIORITY IS VeryLow | JESS (deffrule pr90)
|      |           |        |
|      |           |        |
| 91   | USERTYPE IS 'DBA' AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS 'Insertion' THEN PRIORITY IS VeryLow | JESS (deffrule pr91)
|      |           |        |
|      |           |        |
| 92   | USERTYPE IS 'DBA' AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS 'Copy' THEN PRIORITY IS VeryLow | JESS (deffrule pr92)
|      |           |        |
|      |           |        |

Table 5.4.28 Rule 89 - 92 with JESS Implementation
Rule 93  IF USERTYPE IS ‘DBA’ AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS VeryLow

JESS
(deffrule pr93
?a93<-crispval2 ?ut &:(eq ?ut "DBA")
?b93<-crispval3 ?an&:(eq ?an "UPDATE")
?c93<-shostf ?t&:(fuzzy-match ?t "VeryLow")
?d93<-suserf ?t1&:(fuzzy-match ?t1 "Low")
=> (modify ?"pl"(priority "VeryLow")) (retract ?a93 ?b93 ?c93 ?d93))

Rule 94  IF USERTYPE IS ‘DBA’ AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS VeryLow

JESS
(deffrule pr94
?a94<-crispval2 ?ut &:(eq ?ut "DBA")
?b94<-crispval3 ?an&:(eq ?an "DELETE")
?c94<-shostf ?t&:(fuzzy-match ?t "VeryLow")
?d94<-suserf ?t1&:(fuzzy-match ?t1 "Low")
=> (modify ?"pl"(priority "VeryLow")) (retract ?a94 ?b94 ?c94 ?d94))

Rule 95  IF USERTYPE IS ‘DBA’ AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS VeryLow AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS VeryLow

JESS
(deffrule pr95
?a95<-crispval2 ?ut &:(eq ?ut "DBA")
?b95<-crispval3 ?an&:(eq ?an "UPDATE")
?c95<-shostf ?t&:(fuzzy-match ?t "VeryLow")
?d95<-suserf ?t1&:(fuzzy-match ?t1 "VeryLow")
=> (modify ?"pl"(priority "VeryLow")) (retract ?a95 ?b95 ?c95 ?d95))

Rule 96  IF USERTYPE IS ‘DBA’ AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS VeryLow AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS VeryLow

JESS
(deffrule pr96
?a96<-crispval2 ?ut &:(eq ?ut "DBA")
?b96<-crispval3 ?an&:(eq ?an "DELETE")
?c96<-shostf ?t&:(fuzzy-match ?t "VeryLow")
?d96<-suserf ?t1&:(fuzzy-match ?t1 "VeryLow")
=> (modify ?"pl"(priority "VeryLow")) (retract ?a96 ?b96 ?c96 ?d96))

Table 5.4.29 Rule 93 - 96 with JESS Implementation
Rule 97

IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS Medium

JESS
(defrule pr97
 ?a97<- (crispval2 ?ut &: (eq ?ut "NormalUser"))
 ?b97<- (crispval3 ?an &: (eq ?an "UPDATE"))
 ?c97<- (shostf ?t &: (fuzzy-match ?t "VeryLow"))
 ?d97<- (suserf ?t1 &: (fuzzy-match ?t1 "VeryHigh"))
 => (modify ?"pl"(priority "Medium")) (retract ?a97 ?b97 ?c97 ?d97))

Rule 98

IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS VeryHigh AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS Medium

JESS
(defrule pr98
 ?a98<- (crispval2 ?ut &: (eq ?ut "NormalUser"))
 ?b98<- (crispval3 ?an &: (eq ?an "DELETE"))
 ?c98<- (shostf ?t &: (fuzzy-match ?t "VeryLow"))
 ?d98<- (suserf ?t1 &: (fuzzy-match ?t1 "VeryHigh"))
 => (modify ?"pl"(priority "Medium")) (retract ?a98 ?b98 ?c98 ?d98))

Rule 99

IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS Medium

JESS
(defrule pr99
 ?a99<- (crispval2 ?ut &: (eq ?ut "NormalUser"))
 ?b99<- (crispval3 ?an &: (eq ?an "UPDATE"))
 ?c99<- (shostf ?t &: (fuzzy-match ?t "VeryLow"))
 ?d99<- (suserf ?t1 &: (fuzzy-match ?t1 "High"))

Rule 100

IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS High AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS Medium

JESS
(defrule pr100
 ?a100<- (crispval2 ?ut &: (eq ?ut "NormalUser"))
 ?b100<- (crispval3 ?an &: (eq ?an "DELETE"))
 ?c100<- (shostf ?t &: (fuzzy-match ?t "VeryLow"))
 ?d100<- (suserf ?t1 &: (fuzzy-match ?t1 "High"))
 => (modify ?"pl"(priority "Medium")) (retract ?a100 ?b100 ?c100 ?d100))

Table 5.4.30 Rule 97 - 100 with JESS Implementation
Rule 101 | IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS Low AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Insertion' THEN PRIORITY IS Low

JESS (defrule pr101
  ?a101<-:(crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b101<-:(crispval3 ?an &:(eq ?an "INSERT"))
  ?c101<-:(shostf ?t &:(fuzzy-match ?t "Low"))
  ?d101<-:(suserf ?t1 &:(fuzzy-match ?t1 "VeryHigh"))
=> (modify ?*pl*(priority "Low")) (retract ?a101 ?b101 ?c101 ?d101))

Rule 102 | IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS Low AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Copy' THEN PRIORITY IS Low

JESS (defrule pr102
  ?a102<-:(crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b102<-:(crispval3 ?an &:(eq ?an "SELECT"))
  ?c102<-:(shostf ?t &:(fuzzy-match ?t "Low"))
  ?d102<-:(suserf ?t1 &:(fuzzy-match ?t1 "VeryHigh"))
=> (modify ?*pl*(priority "Low")) (retract ?a102 ?b102 ?c102 ?d102))

Rule 103 | IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS Low AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Insertion' THEN PRIORITY IS Low

JESS (defrule pr103
  ?a103<-:(crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b103<-:(crispval3 ?an &:(eq ?an "INSERT"))
  ?c103<-:(shostf ?t &:(fuzzy-match ?t "Low"))
  ?d103<-:(suserf ?t1 &:(fuzzy-match ?t1 "High"))
=> (modify ?*pl*(priority "Low")) (retract ?a103 ?b103 ?c103 ?d103))

Rule 104 | IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS Low AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Copy' THEN PRIORITY IS Low

JESS (defrule pr104
  ?a104<-:(crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b104<-:(crispval3 ?an &:(eq ?an "SELECT"))
  ?c104<-:(shostf ?t &:(fuzzy-match ?t "Low"))
  ?d104<-:(suserf ?t1 &:(fuzzy-match ?t1 "High"))
=> (modify ?*pl*(priority "Low")) (retract ?a104 ?b104 ?c104 ?d104))

Table 5.4.31 Rule 101 - 104 with JESS Implementation
Rule 105

IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS 'Insertion'
THEN PRIORITY IS VeryLow

JESS
(defrule pr105
?a105<- (crispval2 ?ut &: (eq ?ut "DBA"))
?b105<- (crispval3 ?an &: (eq ?an "INSERT"))
?c105<- (shostf ?t &: (fuzzy-match ?t "VeryLow"))
?d105<- (suserf ?t1 &: (fuzzy-match ?t1 "Low"))
=> (modify ?*pl*(priority "VeryLow")) (retract ?a105 ?b105 ?c105 ?d105))

Rule 106

IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS 'Copy'
THEN PRIORITY IS VeryLow

JESS
(defrule pr106
?a106<- (crispval2 ?ut &: (eq ?ut "DBA"))
?b106<- (crispval3 ?an &: (eq ?an "SELECT"))
?c106<- (shostf ?t &: (fuzzy-match ?t "VeryLow"))
?d106<- (suserf ?t1 &: (fuzzy-match ?t1 "Low"))

Rule 107

IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS VeryLow AND
TRANSCATION IS 'Insertion'
THEN PRIORITY IS VeryLow

JESS
(defrule pr107
?a107<- (crispval2 ?ut &: (eq ?ut "DBA"))
?b107<- (crispval3 ?an &: (eq ?an "INSERT"))
?c107<- (shostf ?t &: (fuzzy-match ?t "VeryLow"))
?d107<- (suserf ?t1 &: (fuzzy-match ?t1 "VeryLow"))
=> (modify ?*pl*(priority "VeryLow")) (retract ?a107 ?b107 ?c107 ?d107))

Rule 108

IF USERTYPE IS 'DBA' AND
SUSPICIOUS HOST FREQ IS VeryLow AND
SUSPICIOUS USER FREQ IS VeryLow AND
TRANSCATION IS 'Copy'
THEN PRIORITY IS VeryLow

JESS
(defrule pr108
?a108<- (crispval2 ?ut &: (eq ?ut "DBA"))
?b108<- (crispval3 ?an &: (eq ?an "SELECT"))
?c108<- (shostf ?t &: (fuzzy-match ?t "VeryLow"))
?d108<- (suserf ?t1 &: (fuzzy-match ?t1 "VeryLow"))

Table 5.4.3.2 Rule 105 - 108 with JESS Implementation
Rule 109

IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Insertion'
THEN PRIORITY IS Low

JESS

(defrule pr109
  ?a109<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b109<- (crispval3 ?an &:(eq ?an "INSERT"))
  ?c109<- (shostf ?t &:(fuzzy-match ?t "VeryLow"))
  ?d109<- (suserf ?t1 &:(fuzzy-match ?t1 "VeryHigh"))

Rule 110

IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS VeryHigh AND TRANSCATION IS 'Copy'
THEN PRIORITY IS Low

JESS

(defrule pr110
  ?a110<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b110<- (crispval3 ?an &:(eq ?an "SELECT"))
  ?c110<- (shostf ?t &:(fuzzy-match ?t "VeryLow"))
  ?d110<- (suserf ?t1 &:(fuzzy-match ?t1 "VeryHigh"))
  => (modify *pl*(priority "Low")) (retract ?a110 ?b110 ?c110 ?d110))

Rule 111

IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Insertion'
THEN PRIORITY IS Low

JESS

(defrule pr111
  ?a111<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b111<- (crispval3 ?an &:(eq ?an "INSERT"))
  ?c111<- (shostf ?t &:(fuzzy-match ?t "VeryLow"))
  ?d111<- (suserf ?t1 &:(fuzzy-match ?t1 "High"))
  => (modify *pl*(priority "Low")) (retract ?a111 ?b111 ?c111 ?d111))

Rule 112

IF USERTYPE IS 'NormalUser' AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS High AND TRANSCATION IS 'Copy'
THEN PRIORITY IS Low

JESS

(defrule pr112
  ?a112<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
  ?b112<- (crispval3 ?an &:(eq ?an "SELECT"))
  ?c112<- (shostf ?t &:(fuzzy-match ?t "VeryLow"))
  ?d112<- (suserf ?t1 &:(fuzzy-match ?t1 "High"))
  => (modify *pl*(priority "Low")) (retract ?a112 ?b112 ?c112 ?d112))

Table 5.4.4.33 Rule 109 - 112 with JESS Implementation
Rule 113  
IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS Low

JESS  
(defrule pr113
  (?a113<- (crispval2 ?ut &:(eq ?ut "NormalUser")))
  (?b113<- (crispval3 ?an&:(eq ?an \"UPDATE\")))
  (?c113<- (shostf ?t&:(fuzzy-match ?t \"Low\")))
  (?d113<- (suserf ?t1&:(fuzzy-match ?t1 \"Low\"))
=> (modify ?*pl*(priority \"Low\")) (retract ?a113 ?b113 ?c113 ?d113)
)

Rule 114  
IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS Low

JESS  
(defrule pr114
  (?a114<- (crispval2 ?ut &:(eq ?ut "NormalUser")))
  (?b114<- (crispval3 ?an&:(eq ?an \"DELETE\")))
  (?c114<- (shostf ?t&:(fuzzy-match ?t \"Low\"))
  (?d114<- (suserf ?t1&:(fuzzy-match ?t1 \"Low\"))
=> (modify ?*pl*(priority \"Low\")) (retract ?a114 ?b114 ?c114 ?d114)
)

Rule 115  
IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS VeryLow AND
TRANSCATION IS ‘Modification’
THEN PRIORITY IS Low

JESS  
(defrule pr115
  (?a115<- (crispval2 ?ut &:(eq ?ut "NormalUser")))
  (?b115<- (crispval3 ?an&:(eq ?an \"UPDATE\")))
  (?c115<- (shostf ?t&:(fuzzy-match ?t \"Low\")))
  (?d115<- (suserf ?t1&:(fuzzy-match ?t1 \"VeryLow\"))
=> (modify ?*pl*(priority \"Low\")) (retract ?a115 ?b115 ?c115 ?d115))

Rule 116  
IF USERTYPE IS ‘NormalUser’ AND
SUSPICIOUS HOST FREQ IS Low AND
SUSPICIOUS USER FREQ IS Low AND
TRANSCATION IS ‘Deletion’
THEN PRIORITY IS Low

JESS  
(defrule pr116
  (?a116<- (crispval2 ?ut &:(eq ?ut "NormalUser")))
  (?b116<- (crispval3 ?an&:(eq ?an \"DELETE\")))
  (?c116<- (shostf ?t&:(fuzzy-match ?t \"Low\")))
  (?d116<- (suserf ?t1&:(fuzzy-match ?t1 \"VeryLow\"))

Table 5.4.34 Rule 113 - 116 with JESS Implementation
### Rule 117

**IF** USERTYPE IS ‘NormalUser’ AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS ‘Modification’ **THEN** PRIORITY IS VeryLow

**JESS**

```jess
(defrule pr117
?a117<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
?b117<- (crispval3 ?an &:(eq ?an "UPDATE"))
?c117<- (shostf ?t &:(fuzzy-match ?t "VeryLow"))
?d117<- (suserf ?t1 &:(fuzzy-match ?t1 "Low"))
```

### Rule 118

**IF** USERTYPE IS ‘NormalUser’ AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS Low AND TRANSCATION IS ‘Deletion’ **THEN** PRIORITY IS VeryLow

**JESS**

```jess
(defrule pr118
?a118<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
?b118<- (crispval3 ?an &:(eq ?an "DELETE"))
?c118<- (shostf ?t &:(fuzzy-match ?t "VeryLow"))
?d118<- (suserf ?t1 &:(fuzzy-match ?t1 "Low"))
=> (modify "pl"(priority "VeryLow")) (retract ?a118 ?b118 ?c118 ?d118))
```

### Rule 119

**IF** USERTYPE IS ‘NormalUser’ AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS ‘Modification’ **THEN** PRIORITY IS VeryLow

**JESS**

```jess
(defrule pr119
?a119<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
?b119<- (crispval3 ?an &:(eq ?an "UPDATE"))
?c119<- (shostf ?t &:(fuzzy-match ?t "VeryLow"))
?d119<- (suserf ?t1 &:(fuzzy-match ?t1 "VeryLow"))
=> (modify "pl"(priority "VeryLow")) (retract ?a119 ?b119 ?c119 ?d119))
```

### Rule 120

**IF** USERTYPE IS ‘NormalUser’ AND SUSPICIOUS HOST FREQ IS VeryLow AND SUSPICIOUS USER FREQ IS VeryLow AND TRANSCATION IS ‘Deletion’ **THEN** PRIORITY IS VeryLow

**JESS**

```jess
(defrule pr120
?a120<- (crispval2 ?ut &:(eq ?ut "NormalUser"))
?b120<- (crispval3 ?an &:(eq ?an "DELETE"))
?c120<- (shostf ?t &:(fuzzy-match ?t "VeryLow"))
?d120<- (suserf ?t1 &:(fuzzy-match ?t1 "VeryLow"))
=> (modify "pl"(priority "VeryLow")) (retract ?a120 ?b120 ?c120 ?d120))
```

Table 5.4.4.35 Rule 117 - 120 with JESS Implementation
Table 5.4.4.36 Rule 121 - 124 with JESS Implementation
Rule 125  IF USERTYPE IS ’NormalUser’ AND
          SUSPICIOUS HOST FREQ IS VeryLow AND
          SUSPICIOUS USER FREQ IS Low AND
          TRANSCATION IS ‘Insertion’
          THEN PRIORITY IS VeryLow

JESS (defrule pr125
  {?a125<-\(\text{crispval2} \ ?ut \ &: (eq \ ?ut \ "NormalUser")}\)
  {?b125<-\(\text{crispval3} \ ?an & (eq \ ?an \ "INSERT")\)}
  {?c125<-\(\text{shostf} \ ?t & (fuzzy-match \ ?t \ "VeryLow")\)}
  {?d125<-\(\text{suserf} \ ?t1 & (fuzzy-match \ ?t1 \ "Low")\)}
  " => (modify \ ?*pl*(priority \ "VeryLow")\) (retract \ ?a125 \ ?b125 \ ?c125 \ ?d125))

Rule 126  IF USERTYPE IS ’NormalUser’ AND
          SUSPICIOUS HOST FREQ IS VeryLow AND
          SUSPICIOUS USER FREQ IS Low AND
          TRANSCATION IS ‘Copy’
          THEN PRIORITY IS VeryLow

JESS (defrule pr126
  {?a126<-\(\text{crispval2} \ ?ut \ &: (eq \ ?ut \ "NormalUser")\)
  {?b126<-\(\text{crispval3} \ ?an & (eq \ ?an \ "SELECT")\)}
  {?c126<-\(\text{shostf} \ ?t & (fuzzy-match \ ?t \ "VeryLow")\)}
  {?d126<-\(\text{suserf} \ ?t1 & (fuzzy-match \ ?t1 \ "Low")\)}
  " => (modify \ ?*pl*(priority \ "VeryLow")\) (retract \ ?a126 \ ?b126 \ ?c126 \ ?d126))

Rule 127  IF USERTYPE IS ’NormalUser’ AND
          SUSPICIOUS HOST FREQ IS VeryLow AND
          SUSPICIOUS USER FREQ IS VeryLow AND
          TRANSCATION IS ‘Insertion’
          THEN PRIORITY IS VeryLow

JESS defrule pr127
  {?a127<-\(\text{crispval2} \ ?ut \ &: (eq \ ?ut \ "NormalUser")\)
  {?b127<-\(\text{crispval3} \ ?an & (eq \ ?an \ "INSERT")\)}
  {?c127<-\(\text{shostf} \ ?t & (fuzzy-match \ ?t \ "VeryLow")\)}
  {?d127<-\(\text{suserf} \ ?t1 & (fuzzy-match \ ?t1 \ "VeryLow")\)}
  " => (modify \ ?*pl*(priority \ "VeryLow")\) (retract \ ?a127 \ ?b127 \ ?c127 \ ?d127))

Rule 128  IF USERTYPE IS ’NormalUser’ AND
          SUSPICIOUS HOST FREQ IS VeryLow AND
          SUSPICIOUS USER FREQ IS VeryLow AND
          TRANSCATION IS ‘Copy’
          THEN PRIORITY IS VeryLow

JESS defrule pr128
  {?a128<-\(\text{crispval2} \ ?ut \ &: (eq \ ?ut \ "NormalUser")\)
  {?b128<-\(\text{crispval3} \ ?an & (eq \ ?an \ "SELECT")\)}
  {?c128<-\(\text{shostf} \ ?t & (fuzzy-match \ ?t \ "VeryLow")\)}
  {?d128<-\(\text{suserf} \ ?t1 & (fuzzy-match \ ?t1 \ "VeryLow")\)}
  " => (modify \ ?*pl*(priority \ "VeryLow")\) (retract \ ?a128 \ ?b128 \ ?c128 \ ?d128))

Table 5.4.4.37 Rule 125 - 128 with JESS Implementation
CONCLUDING REMARKS

In this chapter, a simple implementation of a neuro-fuzzy approach used to prioritization for detected intrusion assistant is portrayed. This assistant is located on the Biometric Template storage database. The intrusion detection is executed in back-ground. When it detects suspicious or illegal activities, it notifies the security administrator. Using about 128 fuzzy rules the priority of detected intrusion is decided. The system consists of a user interface module, an inference engine, a knowledgebase of detected intrusion at biometric template storage. Inference engine is implemented using FuzzyJESS which is fuzzy approach of Java Expert System Shell. The module is developed using Java.