SECURATE: A Security Evaluation and Analysis System Using Fuzzy Metrics

by

Eric H. Michelman and Lance J. Hoffman

Memorandum No. UCB/ERL M77/36

21 July 1977

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

SECURATE: A Security Evaluation and Analysis System Using Fuzzy Metrics

Eric H. Michelman and Lance J. Hoffman

Computer Science Division
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

ABSTRACT

An interactive security evaluation and analysis system which uses fuzzy metrics is described. The system models the installation to be analyzed as a set of object-threat-feature triples. The associated measures--object values, threat likelihoods, and feature resistances--are then used as input to security evaluation functions. The user specifies these measures in terms of "fuzzy" linguistic variables. The system, implemented in APL, is currently operational on an IBM 370/145.

After initial design goals are presented, the actual design implemented is discussed, including the alternatives considered and why certain ones were chosen or discarded.

CR Categories: 2.4, 4.6, 8.1

Keywords: security evaluation, fuzzy-set applications

Research sponsored by National Science Foundation Grant MCS76-09214.

1. INTRODUCTION

This paper describes SECURATE, an interactive computer installation security evaluation and analysis system, based upon Clements' work in modelling a computer installation as a set of triples composed of objects, threats, and security features and upon his "fuzzy" security rating functions (CLEMENTS 1977).

The purpose of SECURATE is to provide data processing managers and security system analysts with a means of analyzing their installation's security. Specifically, this may include security ratings for the installation as a whole as well as subsections, determining weak and strong points, and comparing the effectiveness of alternative security designs. The main purpose, however, is more general than providing the capability for specific analyses. The system is meant to be an aid to help the user increase his or her understanding of, and control over, security design and evaluation issues at a given installation. As such, the tone of the system is to provide a meaningful basis for thoughtful consideration of security problems and to enable the user to try out different ideas easily and effectively. However, the system is not meant to be a substitute for a human decision maker.

Section 2 reviews relevant aspects of Clements' underlying framework. Section 3 discusses the design goals and the design chosen for SECURATE. Section 4 discusses implementation issues, including system structure and the use of APL. Section 5 discusses issues involved in designing the user interface. After the system was implemented, it was used on seven installations by students who were doing risk analyses of the installations. Feedback from this initial group of users is discussed in Section 6.

1. INTRODUCTION

This paper describes SECURATE, an interactive computer installation security evaluation and analysis system, based upon Clements' work in modelling a computer installation as a set of triples composed of objects, threats, and security features and upon his "fuzzy" security rating functions (CLEMENTS 1977).

The purpose of SECURATE is to provide data processing managers and security system analysts with a means of analyzing their installation's security. Specifically, this may include security ratings for the installation as a whole as well as subsections, determining weak and strong points, and comparing the effectiveness of alternative security designs. The main purpose, however, is more general than providing the capability for specific analyses. The system is meant to be an aid to help the user increase his or her understanding of, and control over, security design and evaluation issues at a given installation. As such, the tone of the system is to provide a meaningful basis for thoughtful consideration of security problems and to enable the user to try out different ideas easily and effectively.. However, the system is not meant to be a substitute for a human decision maker.

Section 2 reviews relevant aspects of Clements' underlying framework. Section 3 discusses the design goals and the design chosen for SECURATE. Section 4 discusses implementation issues, including system structure and the use of APL. Section 5 discusses issues involved in designing the user interface. After the system was implemented, it was used on seven installations by students who were doing risk analyses of the installations. Feedback from this initial group of users is discussed in Section 6.

2. TECHNICAL BASIS

As noted, the technical basis for the security evaluation system is the work done by Clements. He has defined an abstraction of a computer security system based upon a view of a security system as a set of security objects, each with a loss value, a set of security threats, each with a likelihood, and a set of security features, each with a resistance.

To address the problem of imprecision in the approximation of values, likelihoods, and resistances, Clements proposes the use of linguistic variables in the specification of these measures and, correspondingly, the use of fuzzy set theory for the combination of the measures into security ratings.

2.1 The Basic System Model

Clements' model focused on those resources within computing systems which are vulnerable to some security threat. These resources are grouped as the set of security objects--*O*. Each object in the set possesses a loss value to its owner.

Associated with each security object is a number of activities which a potential intruder may employ to compromise the security of that object. These potential intrusion activities form the set of security threats--*T*. Each threat has associated with it a likelihood of occurrence.

The object-threat relations form a bipartite directed graph (fig. 2.1) in which edge T_iO_j exists only if threat T_i is a viable means of compromising object O_j. The relations of threats to objects is not one to one; a threat may compromise any number of objects and an object may be vulnerable to more than one threat.

The model is completed with the introduction of a third set, that of security features--*F*. A security feature performs a firewall function by presenting some degree of resistance to a penetration attempt. This resistance measure is refered to as the feature resistance.

The set of security features transforms the bipartite graph of fig. 2.1 into the tripartite graph of fig. 2.2. In a "protected" system all edges are of the form T_iF_k and F_kO_j. Any edge of the form T_iO_j identifies an unprotected object.

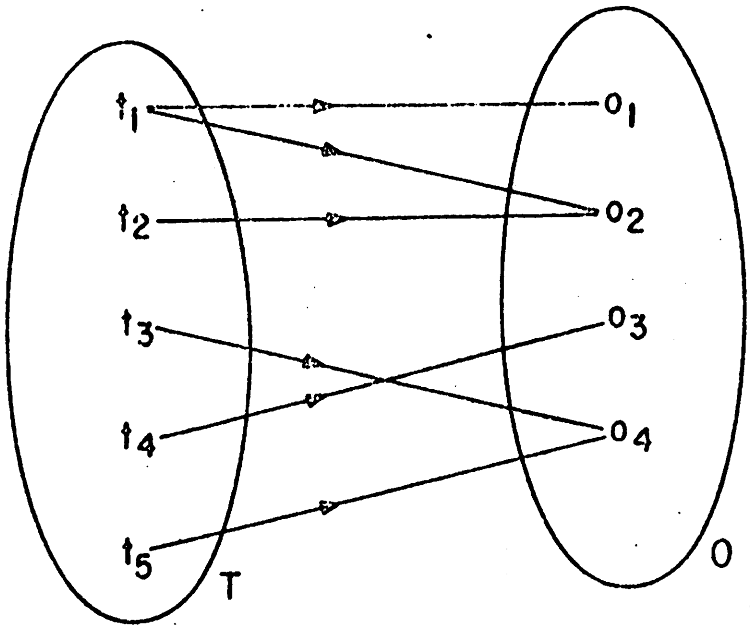


Figure 2.1 The threat-object relation

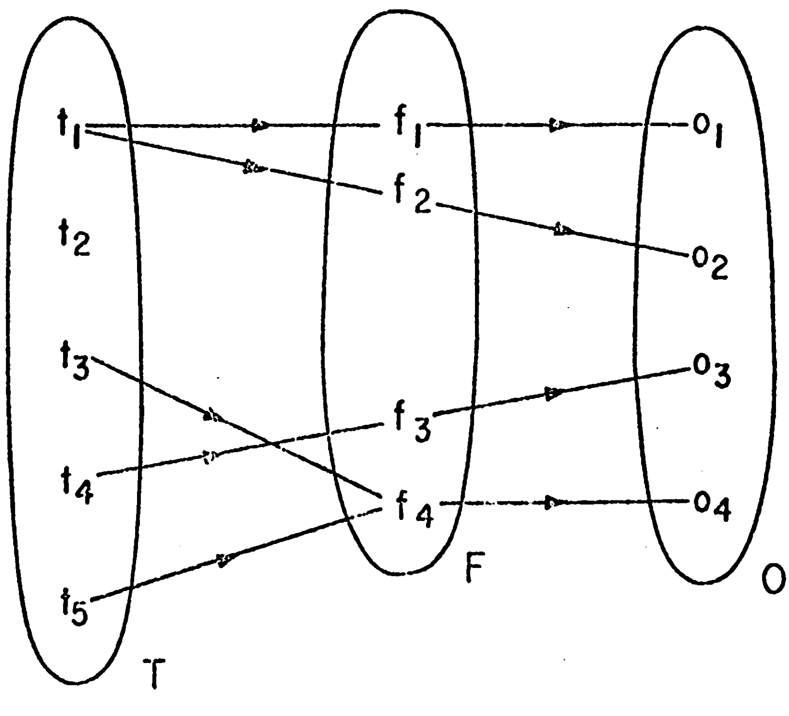


Figure 2.2 The basic security system

2.2 The Use of Linguistic Variables

In attempting to specify the object values, threat likelihoods, and feature resistances one is confronted with the problem of imprecision. In evaluating a computer system's security we must rely on human judgement to provide approximations of these measures. Further, the problem is aggravated when we attempt to produce security ratings from these measures. The assignment of a numerical security rating would be inconsistent with the complexity of the data processing installation when viewed as a system. For example, stating that an installation is ".65 secure" would have limited appeal for imparting a sense of how secure the installation is. In addition, the precision implied by such a rating is likely to cause skepticism.

Clements suggests that it is possible to make meaningful measurements of the security of a computer system through the use of linguistic variables--variables which assumes values which are words rather than numbers (ZADEH 1973).

Using this approach the specification of the object values, threat likelihoods, and feature resistances, as well as the resultant security rating would be in terms of measures such as **high**, **low**, and **medium**. Appropriate modifiers provide finer resolution by allowing terms such as **very high**, **somewhat low**, etc.

Each linguistic variable is a fuzzy set whose members are real numbers in the interval [0,1]. These values comprise the compatibility function, μ_f, for the specific linguistic variable. For example, if $\mu_{high}(0.8) = 0.9$, the 0.9 represents the degree to which a non-fuzzy rating of 0.8 agrees with a fuzzy rating of **high**. Fig. 2.3 illustrates what the complete compatibility functions for **high** and **very high** might be. More detail on base scales and compatibility functions can be found in (ZADEH 1973).

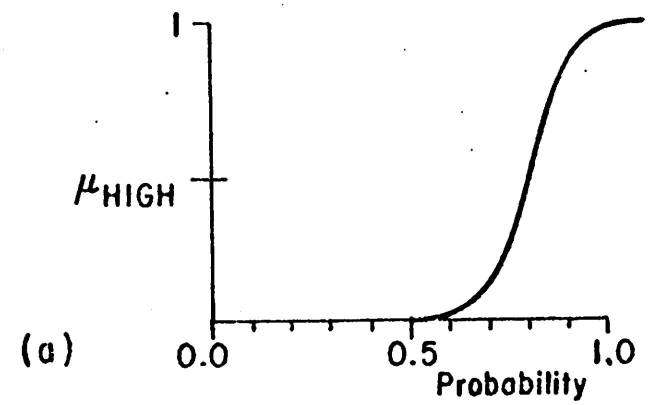


Figure 2.3a Compatibility function of **high** *probability*

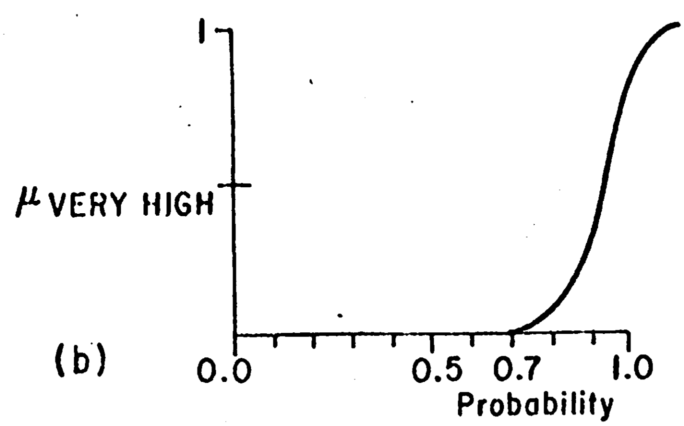


Figure 2.3b Compatibility function of **very high** *probability*

2.3 The Security System Model

The basic model may be specified in terms of a barrier set B in which each element is a composite linguistic variable B_l with three components, corresponding to a object-threat-feature triple. Each component consists of a name and a linguistic value. The structure of B_l is illustrated in fig. 2.4.

Note that objects, threats, and measures appearing in more than one triple may have different values, likelihoods, or resistances, respectively.

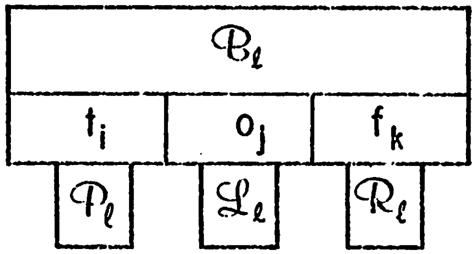


Figure 2.4 The security barrier as a composite linguistic variable

2.4 The Evaluation Process

The user assigns linguistic values (**high**, **medium**, **very high**, etc.) to the component variables P_l, L_l, R_l at each barrier in the system. These measures determine the contribution of the barrier to total system security. How this is done is shown in detail in Section 3.3.1.

3. TECHNICAL DESIGN

3.1 Design Goals

As noted in the Introduction, the objective of the system is to help a security system analyst deal with a rather unstructured and poorly defined problem, that of analyzing an installation's security. Implied in this is that instead of indicating a certain decision to be made or a particular course of action to be taken, the system is to supply appropriate functions to assist the user in an effective analysis.

3.2 The Object Hierarchy and Threats Listing

The evaluation system incorporates a hierarchical structure of objects commonly found in computer installations (MICHELMAN 1977). Associated with the object hierarchy is a listing of corresponding threats and features.

The object hierarchy is used extensively throughout the evaluation system to structure both the analysis and the input. We feel that structuring an installation provides more interesting and informative results as well as making it simpler to analyze intelligently. The alternatives were to forego any structuring of the model or allowing the user to specify his own grouping with no default. Having no facility for structuring the installation--analyzing a straight list of triples--would make it virtually impossible to perform a systematic analysis. The user could only rate the entire installation with no facility for analyzing the components. However, since allowing the user to specify his own grouping may be useful, the system does provide a facility to do that. Using the default is considerably more convenient and less time-consuming, though.

The system allows the user to specify threat and feature numbers as part of the input. This is only a user convenience for identification purposes, though, as the numbers are not used in the analyses.

Another category, flaws, is also presented. Flaws are defined as characteristics of a computing system which enhance the likelihood of a threat succeeding in compromising an object. The purpose of the flaws category is to map what a user may perceive as threats into the threats as viewed by Clements' security model. Flaws are not considered by the evaluation system; they are provided only for user reference.

The object hierarchy and threats, features, and flaws listings are presented in Appendix A.

3.3 System Structure

The basic design of the system is taken directly from Clements' proposals. This includes modelling the installation as a set of triples and using fuzzy set theory to produce security ratings.

There are two phases involved in using the system: (1) inputing a description of the installation and (2) using the security analysis functions.

The installation to be analyzed is described by a set of triples. Each triple consists of an object value, a threat likelihood, and a feature resistance. Each triple is considered to be a "security point of interest". There is one triple for each object-threat pair the user wishes to consider. The number of triples for a given installation is up to the user, more triples implying a more specific representation.

The object value, threat likelihood, and feature resistance are specified by the user in terms of linguistic variables. The terms which may be used are listed, along with their syntax, in an internal system table. While it would not be difficult to incorporate a facility to enable a user to add his own terms, this has not been done due to the difficulties involved in accurately translating a user's English terms into fuzzy set operators and base variables. The vocabulary and syntax of the language, along with examples, is shown in figure 3.1.

The basic system structure is illustrated in figure 3.2.

Once the installation to be analyzed is described in terms of these triples, the functions described in section 3.3.1 can be invoked by the user to evaluate and analyze its security. As Clements had already implemented the scoring functions which produce a security rating for a given set of triples, our implementation effort involved mainly establishing (1) a facility to create the set of triples, (2) analysis functions which make use of the scoring functions, and (3) a user interface.

3.3.1 The Evaluation Functions

There are presently four security evaluation functions implemented:

A) Overall System Rating--This function returns a security rating for the entire installation. That is, it rates the entire set of triples.

B) Individual Subsection Rating--a security rating is returned for a specified subsection of the installation. Only triples for that subsection (including offspring) are considered. For example, for an individual subsection rating of the central machine, the evaluation system would consider triples specified for the central machine and each of its offspring--the CPU, main memory, I/O devices, and the operator's console. Refer to Appendix A for the actual hierarchy listing.

<sentence> ::= <compound phrase> | <simple phrase>

<compound phrase> ::= <conjunctive phrase> | <range phrase>

<simple phrase> ::= <relational phrase> | <hedged primary>

<conjunctive phrase> ::= <relational phrase> AND <relational phrase>

<range phrase> ::= <hedged primary> TO <hedged primary>

<relational phrase> ::= <composite relation> THAN <hedged primary>

<composite relation> ::= <relation hedge> <relation> | <relation>

<relation hedge> ::= NOT | MUCH | SLIGHTLY

<relation> ::= LOWER | HIGHER

<hedged primary> ::= <hedge> <primary> | <primary> | <fuzzy number>

<hedge> ::= NOT | VERY | MOREORLESS | QUITE | PRETTY |
SORTOF | REALLY | EXTREMELY | INDEED

<primary> ::= LOW | HIGH | MEDIUM

<fuzzy number> ::= <fuzzifier> <number>

<fuzzifier> ::= ABOUT

<number> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10

Some of the rating phrases which may be generated with this grammar are:

high
low
medium
not high
moreorless medium
indeed low
low to medium
(about 4) to about 6
slightly lower than pretty high
not higher than medium
(much higher than low) and slightly lower than sortof high

Figure 3.1 Language BNF with examples

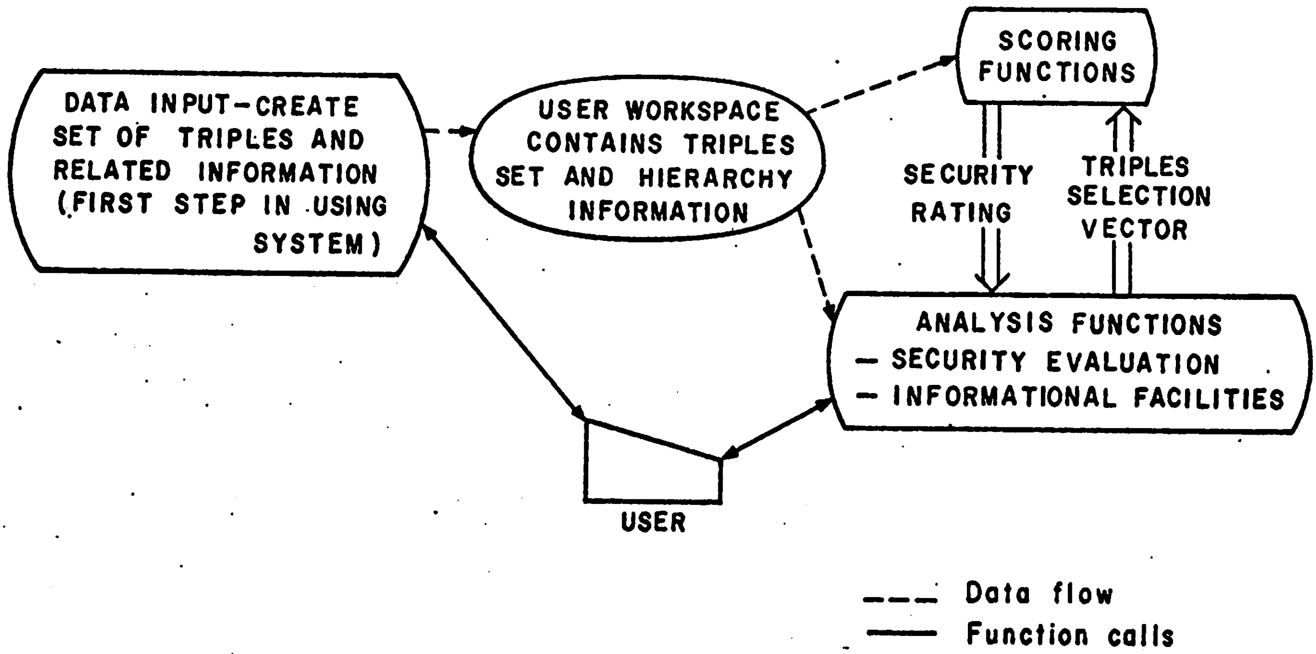


Figure 3.2 The basic system structure

C) Sectional Ratings--with either the top level of the installation hierarchy or one of the subsections having been specified, this function returns an individual rating for each subsection at the next lower level. For example, if the top level of the hierarchy was specified for a sectional analysis, security ratings would be printed out for each of the following subsections: hardware, software, the computer center, personnel, documentation, and the backup system.

D) Worst Subsection Ratings--this performs the same functions as the sectional ratings function with the additional feature that it highlights which subsection received the lowest rating.

In addition to choosing which of the above evaluation functions to use, the user must also choose among four methods of producing a security rating for a given set of triples. The four scoring functions, as implemented by Clements, are:

A) Weakest Link--this will look for the weakest feature resistance and return that as the security rating. The theory here is that the system is only as secure as its weakest link.

B) Selected Weakest Link--this produces a weakest link rating based on those triples which satisfy the condition that either the object value or the threat likelihood is greater than a user specified minimum. The theory here is that one would only want to consider triples where the object is of at least a certain value or the threat is of at least a certain likelihood.

C) Fuzzy Mean--this performs a fuzzy mean on the feature resistances and returns the result as the rating. The theory here is that a system's security is the mean of the security of its components.

D) Weighted Fuzzy Mean--this performs a fuzzy mean on the feature resistance weighted by the greater of the object value and threat likelihood for each triple. The theory is that of (C), with the additional assumption that the more valuable objects and those with more likely threats should receive greater weight in the security rating.

E) Fuzzy Mean With Each Major Subsection Weighted By Maximum Object Value-- for each major subsection of the object specified, this finds the fuzzy mean of the resistances. It then weights these fuzzy means by the maximum object value found in the triples for each major subsection and averages these weighted means. In other words, it finds the fuzzy means for each major subsection and weights them by their respective maximum object value. The theory is similar to (D), but with the assumption that the major subsections should be weighted by their relative values, irrespective of the number of triples they each have.

In choosing a scoring function, the user in effect describes how he views security. Once a scoring function is chosen, it stays in effect for all of the analysis functions until it is respecified.

3.3.2 Establishing the Representation of the Installation

Before the analysis functions can be used on an installation, the user must input the information necessary to create the set of triples and the related hierarchical information.

The system starts with the assumption that the installation will be basically similar to that modelled by the hierarchy in Appendix A. As such, the evaluation system has the hierarchy programmed in, although the user can modify it appropriately as he supplies the triples information.

Given the initial hierarchical structure and the user's modifications to it, the system leads the user through the hierarchy, giving him the opportunity at each node to add offspring or specify triples. If a triple is specified for an object with offspring, it is assumed to refer to that object and each of its offspring. Refer to Appendix B for an example of the system in use.

The user has the option of associating threat and feature numbers with each triple. These numbers are solely for identification purposes; no analysis functions consider them. They may refer to the lists of threats and features associated with the object hierarchy, or may be numbers chosen by the user according to his own numbering scheme. If a number used is one of those in the threat or feature listings supplied in Appendix A (nos. 1-129 for threats and nos. 1-274 for features), the corresponding will be printed out by the display function.

Once the triples are entered, they may be printed out using the display function. For each triple this prints out: the triple number, the object name, number, and value, the threat name, number, and value, and the feature resistance. See Appendix B, an example of the system in use, for an example of the display output.

Once the information describing the installation is entered it is automatically saved and may be used later with repeated applications of the system.

4. IMPLEMENTATION

The implementation effort was started in January, 1977. The functions which return a security rating when given a set of triples had already been implemented by Clements in APL on the UCLA 360/91. The system was initially working by the middle of March, although considerable debugging and refinement took place later. In April we moved the system to the UCSF VM/370 system because of space limitations on the UCLA system. The system described here is that running as of June, 1977.

4.1 Design Goals

As we couldn't be sure which functions would be most useful (something which is different for different users), a primary implementation goal was that the system be easy to modify. This implies that it be modular and have easily understandable code, something not to be taken for granted with APL. It also accounts for our lack of concern for optimization, which would have been counter-productive during implementation.

4.2 System Structure

The modular structure required for the necessary flexibility in development was fairly easy to achieve. At the center of the system is the scoring facility implemented by Clements. Given a set of triples, it returns a rating using one of four scoring functions. Additional scoring functions may be added by users familiar with APL. Each of the security evaluation functions is interfaced to this common kernel, passing it an appropriate set of triples to be rated and then processing the result (fig. 3.2).

The triples are kept in a user's file along with a variable containing the object numbers corresponding to each triple, a variable containing the threat number for each triple, and four variables containing the hierarchical information.

When a user wants to start doing an analysis, the variables containing the information for his installation are loaded into the APL workspace along with the analysis functions. He can then call any of the analysis functions simply by entering its name. An example of the system in use is shown in Appendix B.

The program flow is simple and straightforward when a user calls a security rating function. The function called determines which triples are to be rated (depending on which section(s) of the installation is to be rated) and passes an appropriate index vector to the scoring routine. Following are descriptions of the system tables involved. Figure 4.1 illustrates the algorithm involved in selecting triples to be rated.

ΔMAP--this contains a linear list of the object numbers found in the hierarchy. The indices of the object numbers in ΔMAP are the OBJECTID's used by the system internally.

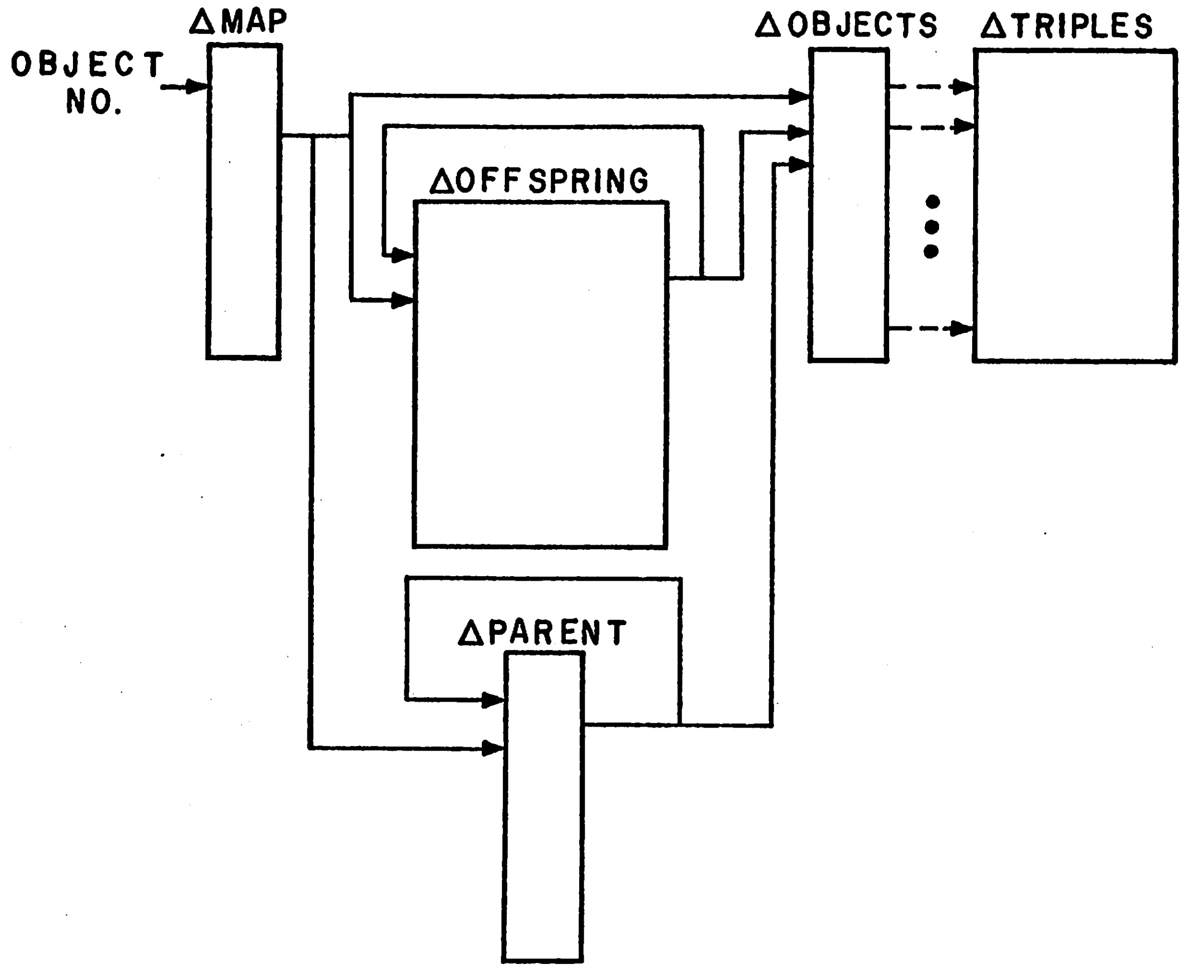
ΔOFFSPRING--each row contains the OBJECTID's of the offspring of the object whose OBJECTID is equal to the row number.

ΔPARENT--contains the parent OBJECTID of each object, again, indexed by OBJECTID.

ΔTRIPLES--this contains the triples as input by the user. There are three lines per entry corresponding to an object value, a threat likelihood, and a feature resistance.

ΔOBJECTS--this contains one entry for each triple, indicating the object number of the object associated with each triple.

To set up the triples and the hierarchy information, the user calls a program which leads him through the standard object hierarchy, giving him the opportunity to add offspring and specify triples at each node in the hierarchy. Much of the programming in this section is devoted to making sure that the hierarchical structure stays consistent, both internally and with regard to the set of triples. This is important as the analysis functions use the hierarchy information to select the triples to be rated.



Algorithm for selecting triples to be rated:

1) Search ΔMAP for OBJECT NO., the index becomes the new OBJECT ID.
2) Look up the "OBJECT ID"th row in ΔOFFSPRING for the OBJECT ID's of the offspring objects. This process is recursive.
3) Look up the "OBJECT ID"th element in ΔPARENT for the OBJECT ID of the parent object. This process is recursive.
4) Search OBJECTS for entries matching the original OBJECT ID, or the OBJECT ID's of parents and offspring. These indices are the triple numbers of the triples to be rated.

Note that each of these steps, with the exception of recursion, is easily performed by one APL statement.

Figure 4.1 Triple selection for evaluation

4.3 The Use of APL

APL is extremely well suited to applications involving linguistic variables and fuzzy set operations. Using appropriately named functions and variables, the linguistic variables can be easily converted into the corresponding base variables (ZADEH 1973) using the APL "execute" function. For example, HIGH might be a vector consisting of (0 0 0 0 0 .1 .5 .9 1), representing the linguistic variable **high**. VERY might be a function which sharpens the curve given to it as its argument, perhaps squaring the argument. Then, as shown in figure 4.2, if VALUE were a variable containing the character string "VERY HIGH", executing it would return the vector <0 0 0 0 0 .01 .25 .81 1>, representing the base variable for the linguistic variable **very high** (Figure 2.3 gives the curves representing **high** and **very high**). The important point here is that APL eliminates the need to do any parsing of the input values; the linguistic variables input just get executed and thusly transformed into the base variables. Additionally, the built-in APL matrix operations are well suited to the fuzzy set operators, which use vectors and matrices extensively. These operators are described in detail in (CLEMENTS 1977).

```
      ∇VERY[⎕]∇
      ∇ OUT←VERY IN
[1]   OUT←IN×IN
      ∇

      HIGH
0 0 0 0 0 0.1 0.5 0.9 1

      VALUE
VERY HIGH

      ⍎VALUE
0 0 0 0 0 0.01 0.25 0.81 1
```

Figure 4.2 APL execution of linguistic variables

Software development is comparatively easy in APL due to its interpretive nature. Contributing to this are the system facilities for debugging, such as the trace capability.

On the negative side, APL is interpretive; this makes it significantly slower than compiled programs for repeated runs. In addition, it is poorly suited to applications not involving vectors or arrays. The latter point is important for the security evaluation system since most of the code deals with the user interface and the analysis functions. Not only were these awkward to program, but they run rather slowly (these two points not being unrelated). The rating functions, however, which make heavy use of the matrix capabilities while performing fuzzy set operations, are well suited to APL.

5. THE USER INTERFACE

From the start of the project, an important objective was to design and implement the system so that it would be as hospitable to the users as possible.

Our goals concerning user oriented features were primarily to keep the system simple, easy to use, and non-tedious. More specifically, we were concerned with the following points:

A) User Understanding--for obvious reasons, achieving adequate user understanding is very important. Not only won't the system be useful if the user doesn't understand it, but it won't be used.

B) Simple, Non-tedious Interface--a similar, much simpler system was developed by a student at Berkeley as a term project. A unanimous criticism of that system was that it took too long to use and the data entry was too tedious. As our system was to require considerably more information, it seemed important to keep the interaction as short, concise, and painless as possible.

C) Useful Analysis Functions--while it may seem that this is the most important point, it may actually be the least. A system which a user understands and is comfortable using is more likely to be used and be helpful than a system that doesn't possess these qualities, even if the functions provided by the first aren't quite as useful as those provided by the second.

The design question in this area which we spent the most time considering was the form of the user interface for inputing the installation data. The process was simplified somewhat by the use of the hierarchical model of objects and threats. Since the users used this as a guide for collecting their data, it provided a convenient basis for structuring the input. We initially prompted the user for all the information. This turned out to be overly tiresome, however, as the same questions would be asked over and over, covering all the possibilities for each object. Two modifications made the process for more manageable. The first was to have the user specify keywords (or abbreviations thereof) followed by the relevant information, instead of prompting him for the information. This greatly reduced the number of lines appearing on the screen. The second modification was to draw up forms which correspond in format exactly with what would appear on the screen. The combined effect of these two modifications was to allow the user to write down on the forms only the necessary information and then transfer it easily to the system. Figures 5.1 A and B, excerpts from Appendix B, show an example of the input form and the corresponding data entry.

Refer to the users' manual (HOFFMAN 1977) for further information concerning the user interface.

OBJECT NO: 1

ADD, A name or number: A METERING EQUIPMENT

VALUE, V object value:

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE

OBJECT NO: 11

ADD, A name or number:

VALUE, V object value: V VERY HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
8	MEDIUM	2	PRETTY HIGH
10	PRETTY LOW	29 30	MEDIUM

Figure 5.1a Data input form

```
ENTER THE OBJECT NUMBER FOR THE NEXT OBJECT:
1
HARDWARE
:
ADD METERING EQUIPMENT
METERING EQUIPMENT RECEIVED OBJECT NUMBER 71
:
O
OBJECT NO 11, CENTRAL MACHINE IS NEXT.
:
V VERY HIGH
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→
8 MEDIUM 2 PRETTY HIGH
→
10 PRETTY LOW 29 30 MEDIUM
→

:
```

Figure 5.1b Data entry

6. USER REACTIONS

Shortly after development started on the evaluation system, we arranged to have it tested by students who were doing risk analyses of computer installations as term projects. Some of these people were full time students while others were part-time students who worked full time at their installation. In all, the evaluation system was used to analyze seven installations, including one at the Bank of America and one at the Pacific Gas & Electric Co.

In addition to receiving reactions to the system when it was tested, we received useful feedback from these people during the design phase. This was especially true for the user interface. Through a series of group meetings we were able to present different design questions and options to our group of users. Their reactions were very useful in determining what features would be well accepted and how they should be presented.

6.1 Use of the System

Prior to our users actually sitting down at a terminal to use the system, we had to familiarize them with the workings of the system and they had to collect the necessary triples information for their respective installations.

As the familiarization process had been going on from the start via the series of meetings, when the time came to use the system we had only to instruct the users in the details of its operation. The input format forms which we distributed were very useful for both collecting the data and, by integrating the system commands with the input data in a coherent way, familiarizing the users with the system's operation prior to using it. Usually, a user would input the installation data and do some initial analysis during the first terminal session; he would then come back once or twice to do additional analysis.

6.2 User Reactions

Each of the users wrote up their impressions of the system as part of their coursework. This included the evaluation of its usefulness as well as suggestions for improvement. From their papers, as well as conversations with them, it seems clear that the system achieved its goal of increasing understanding of installation security. In fact, a couple of users remarked that just filling out the forms made the strengths and weaknesses of the installation's security a lot clearer. Apparently just focusing their thoughts into a logical, well defined framework enabled them to view the situation more clearly and--before even using the system-- to gain some of the insights we had hoped the system would provide.

The most interesting observations were those concerning the use of fuzzy variables. There appears to be a definite tradeoff between user acceptance and ease of use. The concept of fuzzy variables was new to all of the users and it was greeted with a certain amount of skepticism. While their acceptance of the idea grew as they continued to be exposed to it and had experience in using it, some of them remained skeptical. On the other hand, some of them commented, and we strongly feel to be true, that the use of these words instead of numbers was a definite help in minimizing the tedium involved in collecting the input data. The largest installation turned out to be represented by 136 triples, which came to over 300 different measurements the user had to make. Pinpointing each one on a scale of 1 to 10 appears to us to be a lot more taxing than rating each one as a linguistic variable. Although we didn't do any comparative studies (which in retrospect would have been a good idea), many users seemed to agree with this in informal discussions.

The most common criticism was the lack of comprehensive input checking. When the system was first used it didn't check for bad data and would consequently blow up when it tried to process such data. While this only took about a minute to fix, it was very annoying and irritating to the users to have to ask for assistance every time they made a mistake or typo. Since then we have implemented facilities for complete checking of input form and vocabulary.

7. SUMMARY

We have described an interactive security evaluation and analysis system which uses fuzzy metrics. The system models the installation to be analyzed as a set of object-threat-feature triples. The associated measures--object values, threat likelihoods, and feature resistances--are then used as input to security evaluation functions. The user specifies these features in terms of "fuzzy" linguistic variables. The system, implemented in APL, is currently operational on an IBM 370/145.

REFERENCES

(CLEMENTS 1977) Don Clements, "Fuzzy Ratings for Computer Security Evaluation", Memorandum No. UCB/ERL M77/41, June 1977, Electronics Research Laboratory, College of Engineering, University of California, Berkeley.

(HOFFMAN 1977) Lance J. Hoffman, Eric H. Michelman, and Don Clements, "SECURATE User's Manual", Memorandum No. UCB/ERL M77/49, Electronics Research Laboratory, College of Engineering, University of California, Berkeley.

(MICHELMAN 1977) Eric H. Michelman, "A Practical Framework for Computer Installation Security", Memorandum No. M77/4, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, June 1977.

(ZADEH 1973) L. A. Zadeh, "The Concept of the Linguistic Variable and its Application to Approximate Reasoning", Memorandum No. ERL-M411, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, 15 October 1973.

Appendix A

The Object Hierarchy and

Threats, Features, and Flaws Listings

The Object Hierarchy

1. Hardware
2. Software
3. The Computer Center
4. Personnel
5. Documentation
6. Backup system

1. Hardware
    1.1 Central machine
        1.1.1 CPU
        1.1.2 Main memory
        1.1.3 I/O channels
        1.1.4 Operator's console
    1.2 Storage medium
        1.2.1 Magnetic media
            1.2.1.1 Disk packs
            1.2.1.2 Magnetic tapes
            1.2.1.3 Diskettes (floppies)
            1.2.1.4 Cassettes
            1.2.1.5 Other
        1.2.2 Non-magnetic media
            1.2.2.1 Punched cards
            1.2.2.2 Paper tape
            1.2.2.3 Paper printout
            1.2.2.4 Other
    1.3 Communications equipment
        1.3.1 Communications lines
        1.3.2 Communications processor
        1.3.3 Multiplexor
    1.4 I/O devices
        1.4.1 User directed I/O devices
            1.4.1.1 Printer
            1.4.1.2 Card reader
            1.4.1.3 Card punch
            1.4.1.4 Paper tape reader
            1.4.1.5 Paper tape punch
            1.4.1.6 Terminals
                1.4.1.6.1 Local terminals
                1.4.1.6.2 Remote terminals
            1.4.1.7 Modems
        1.4.2 Storage I/O devices
            1.4.2.1 Disk drives
            1.4.2.2 Tape drives

2. Software

- 2.1 Operating system
- 2.2 Programs
  - 2.2.1 Applications
    - 2.2.1.1 Source
    - 2.2.1.2 Non-source
  - 2.2.2 Contract programs and packages
  - 2.2.3 System utilities
  - 2.2.4 Test programs
- 2.3 Data
  - 2.3.1 Personal data
    - 2.3.1.1 Payroll
    - 2.3.1.2 Personnel
    - 2.3.1.3 Other personal data (Privacy Act of 1974, §3(a)(4))
  - 2.3.2 Institution data
    - 2.3.2.1 Marketing
    - 2.3.2.2 Financial
    - 2.3.2.3 Operations
    - 2.3.2.4 Planning
    - 2.3.2.5 Other

3. The Computer Center

3.1 Resource supply systems

3.1.1 Air conditioning

3.1.2 Power

3.1.3 Water

3.1.4 Lighting

3.2 Building

3.2.1 Structure

3.2.2 Computer operations

3.2.2.1 Computer room

3.2.2.2 Data reception

3.2.2.3 Tape and disc library

3.2.2.4 CE room

3.2.2.5 Data preparation area

3.2.2.6 Physical plant room

3.2.2.7 Stationery storage

3.3 Waste materials

3.3.1 Paper

3.3.2 Ribbons

3.3.3 Magnetic materials

4. Personnel

4.1 Computer personnel

4.1.1 Supervisory personnel

4.1.2 Systems analysts

4.1.3 Programmers

4.1.3.1 Applications programmers

4.1.3.2 Systems programmers

4.1.4 Operators

4.1.4.1 First shift

4.1.4.2 Second and third shifts

4.1.5 Librarians

4.1.6 Temporary employees and consultants

4.1.7 Maintenance personnel

4.1.8 System evaluators and auditors

4.1.9 Clerical personnel

4.2 Building personnel

4.2.1 Janitors

4.2.2 Watchmen

4.3 Institution executives

4.4 Other personnel

5. Documentation

5.1 Software documentation

5.1.1 File

5.1.2 Program

5.1.3 JCL

5.1.4 System

5.2 Hardware documentation

5.3 Operations

5.3.1 Schedules

5.3.2 Operations guidelines and manuals

5.3.3 Audit documents

6. Backup system

6.1 Hardware

6.1.1 Replacement for equipment detailed in section 1

6.1.2 Replacement time

6.2 Backup for software detailed in section 2

6.3 The Computer Center

6.3.1 Electric power generation

6.3.2 Generator fuel supply

6.3.3 Water supply

6.4 Auxiliary personnel

6.5 Documentation, operational procedures

6.5.1 Vital records

6.5.2 Priority run schedules

6.5.3 Backup for documentation in section 5

Threats and Flaws

The structure of the threats is based on the object hierarchy, which is used as an outline. Threats are listed after the objects they refer to, the objects being specified by name and number from the object hierarchy. A threat listed after a non-terminal node of the object hierarchy refers to all objects decending from that node.

The numbers of relevant flaws are listed after each threat. The flaw numbers are preceded by an "F" and are ordered sequentially within each of the six main object/threat categories. The flaws themselves are listed along with their corresponding numbers after threat listings for each of the six main categories.

1. Hardware

1.1 Central machine

1) Malicious destruction - F1.1

2) Hardware error - F1.4

3) Hardware tampering - F1.1, F1.4, F1.5

4) modified operation

5) loss of data

6) modification of data

7) Tampering with panel controls

8) Unauthorized use - F1.2

9) Unauthorized change in operating characteristics during operation - F1.2

10) Human error - F1.6, F1.7

1.2 Storage media

11) Theft - F1.3

12) Unauthorized modification - F1.3

13) Unauthorized read - F1.3

1.3 Communications equipment

14) <same threats as 1.1 Central machine>

1.4 I/O devices

15) <same threats as 1.1 Central machine>

Hardware Flaws

F1.1 Inadequate plant security

F1.2 Lack of status indicators

F1.3 Inadequate storage library security

authorization

guard

labeling

diligence in keeping materials stored properly

F1.4 Lack of machine checks, hardware and software

F1.5 Unsupervised or unauthenticated CE activity

F1.6 Operator ignorance

F1.7 Misleading documentation, incomplete or inadequate

2. Software

16) A. Unauthorized access: R/W/E - F2.1, F2.2

17) Modification of operating system and system routines

18) Inadequate controls on I/O facilities - F2.3, F2.4

19) Password compromise - F2.5, F2.6, F2.7, F2.8

20) Unsecured storage medium - F2.9, F2.10, F2.11, F2.12

21) Access outside of allocated memory - F2.13, F2.14, F2.15

22) Modification of stored state vector - F2.16

23) Unauthorized CE activity

24) Line tapping and spoofing

25) Erroneous or inadequate usage of protection facilities - F2.17, F2.18, F2.19

26) B. Unauthorized access: read

27) Extra copies of output printed

28) duplicates printed

29) printing restarted before end

30) Use of erroneous distribution labels

31) Use of erroneous distribution lists

32) Theft of mail

33) Exposed output - F2.20, F2.21

34) in user possession

35) within distribution system

36) at operator's console

37) work in progress

38) Unauthorized reading of terminal buffers

39) Indirect exposure of output - F2.22, F2.23

40) C. Unauthorized access: write

41) Modification or spoof of mail transactions

42) Unauthorized modification of data during preparation - F2.24

43) Data preparation errors - F2.24

44) Modification of original written data input - F2.25

2.1 Operating system

45) Defective implementation - F2.26, F2.27, F2.28, F2.29, F2.30, F2.31, F2.32

2.2 Programs

46) Inadequate debugging

47) Incomplete operation specifications

48) Inadequate or erroneous error handling

49) Exposure following abnormal end

50) Improper operation

2.2.2 Contract programs and packages

51) Dishonest programs

2.2.4 Test programs

52) Unexpected alteration of real data

Software Flaws

F2.1 Faulty access control mechanism

F2.2 Non-functional protected state mechanism

F2.3 Ability to use self-modifying I/O code

F2.4 Ability to write file into other user's catalog

F2.5 Printout of password at terminal

F2.6 Exposed input on spooling facility

F2.7 Use of user selected password

F2.8 Storage of password in unencrypted form

F2.9 Inadequate physical access controls

F2.10 Inadequate operator procedure

F2.11 Ability to spoof operator

F2.12 Improper labeling

F2.13 Inadequate base/bounds checking

F2.14 Unprotected storage after system crash

F2.15 Unprotected storage during system initialization

F2.16 State vector stored in user storage

F2.17 User interface of protection system too complex

F2.18 Inaccurate documentation

F2.19 Incomplete documentation

F2.20 Materials left exposed during emergency

F2.21 Output not checked for proper content

F2.22 Sensitive jobs printed with new ribbon

F2.23 Exposed waste materials

F2.24 Inadequate total and edit checks

F2.25 Inadequate control of hard copy input data

F2.26 Excessive complexity

F2.27 Non-detected bugs (inadequate testing)

F2.28 Improper design specifications

F2.29 Access control based on checking for lack of permission

F2.30 Effectiveness of protection system based on ignorance

F2.31 Overprivileged system modules

F2.32 Lack of violation recording and review

3. The Computer Center

3.1 Resource supply systems

53) Natural calamities
54) Fire
55) Flood
56) Earthquake
57) Manmade disasters
58) Smoke
59) Rioting
60) Bombing
61) Vandalism
62) Fate (chance events)
63) Equipment breakdown
64) Shutdown of building facilities

3.1.2 Power

65) Blackout
66) Fluctuations
67) Grounding problems

3.1.3 Water

68) Disruption
69) Contamination
70) Temperature variations

3.1.4 Lighting

71) Blackout

3.2 The Building

72) Natural calamities
73) Fire
74) Flood
75) Earthquake
76) Manmade disasters
77) Smoke
78) Rioting
79) Bombing
80) Vandalism

3.2.2 Computer operations area

81) Shocks and vibrations

82) Communications breakdown

83) Illegal entry and burglary

3.2.2.1 Computer room

84) Magnets

85) Electromagnetic radiation, to and from

3.2.2.2 Data reception

86) Unauthorized intruders

3.2.2.3 Tape and disk library

87) Magnets

3.2.2.6 Physical plant room

88) Sabotage

3.3 Waste materials

89) Unauthorized reading

90) Theft

4. Personnel

91) Bribery - F4.1

92) Dissatisfaction or malice - F4.1, F4.2

93) Towards the institution

94) Towards management

95) Towards other workers

96) Towards others (possibly unknown)

97) Greed - F4.1, F4.2

98) Competitor encouraged

99) Entrepreneurial tendencies

100) Incompetence - F4.1

101) Coercion - F4.1, F4.2

102) Competitor plants (industrial espionage)

103) Carelessness - F4.1

Personnel Flaws

F4.1 Personal instability

F4.2 Job insecurity

5. Documentation

104) Loss - F5.1, F5.2

105) Thievery - F5.1, F5.2

106) Unauthorized viewing - F5.1, F5.2

107) Unauthorized modification - F5.1, F5.2

Documentation Flaws

F5.1 Inadequate signout procedures

F5.2 Documentation left unsecured

6. Backup system

108) Limited or no accessibility - F6.1, F6.2, F6.3, F6.4, F6.5

6.1 Hardware

109) Incompatibility with other equipment in use
110) Ignorance of operation
111) <additionally, same considerations as section 1, Hardware threats>

6.2 Software

112) Not up to date
113) Incompatible system components
114) Ignorance of use
115) Lack of necessary data
116) <additionally, same considerations as section 2, Software threats>

6.3 The Computer Center

117) Malfunctioning power generation system
118) Shortage of generator fuel
119) Shortage of operation materials
120) <additionally, same considerations as section 3, Computer Center threats>

6.4 Personnel

121) Lack of transportation to backup site
122) Lack of communication

6.5 Documentation, operational procedures

123) Inadequate communications facilities
124) Incompatible run procedures
125) Inadequate office, other operational facilities
126) Unplanned emergency run schedules
127) Inadequate personnel direction
128) Confusion during disaster - F6.6
129) <additionally, same considerations as section 5, Documentation threats>

Backup System Flaws

F6.1 Excessive time involved in traveling to backup installation
F6.2 Excessive distance involved in traveling to backup installation
F6.3 Excessive cost involved in transportation to backup installation
F6.4 Ignorance about how to get at backup (real-time)
F6.5 Non-existence of all or part of backup
F6.6 Lack of simulated disaster tests

PRINTFEATURES

FEATURE NO	THREAT NOS	FEATURE NAME
1	1	*PHYSICAL SECURITY*
2		*GUARD*
3		*ID CARD DOOR*
4		*PROPER LOCATION OF CENTER*
5		*SECURE DOOR AND WINDOW LOCKS*
6		*PERSONAL SEARCHES*
7		*TWO OPERATOR SYSTEM*
8		*ENTRANCE LOG*
9		*OUTSIDE LIGHTING*
10		*FENCE*
11		*ALARM SYSTEM*
12		*CLOSED CIRCUIT TV*
13		*ID BADGES*
14		*SECURE DOORS AND WINDOWS*
15	2	*ADEQUATE MAINTENANCE*
16		*ERROR CORRECTING CODES*
17		*INTERNAL MACHINE CHECKS*
18		*REDUNDANT PROCESSORS*
19	3 4 5 6	*<THE SAME FEATURES AS THREAT NO. 1>*
20		*SUPERVISION AND AUTHENTICATION OF CE'S*
21		*LOCKS AND ALARMS ON MACHINE COVERS*
22	7	*<THE SAME FEATURES AS THREAT NO. 1>*
23	8	*AUTOMATIC LOG*
24		*LOCKS ON CONTROLS*
25		*<ADDITIONALLY, THE SAME FEATURES AS THREAT NO. 1>*
26	9	*STATUS INDICATORS*
27		*AUTOMATIC LOG*
28	10	*PROPER LABELLING*
29		*OPERATOR TRAINING*
30		*DETAILLED, ACCURATE, ACCESSIBLE DOCUMENTATION*
31	11	*PHYSICAL ACCESS CONTROLS*
32		*PACKAGE AND BRIEFCASE INSPECTION*
33		*GATE-PASS SYSTEM*
34		*SECURE LIBRARY FACILITY*
35		*PROPER LABELLING*
36	12	*CONTROL CHECKS*
37		*CHECKSUM ON DATA*
38		*EFFECTIVE STORAGE ACCESS CONTROLS*
39		*HEADER CHECKING*
40		*PREVENTIVE MEASURES*
41		*WRITE-INHIBIT SWITCHES*
42		*RING OUT FOR TAPES*
43	13	*DATA ENCRYPTION*
44		*EFFECTIVE STORAGE ACCESS CONTROLS*
45	14 15	*<THE SAME FEATURES AS THREATS 1-13>*
46	16	*EFFECTIVE AUTHORIZATION AND ACCESS CONTROL MECHANISM*

46	16	EFFECTIVE AUTHORIZATION AND ACCESS CONTROL MECHANISM
47		MINIMUM AUTHORIZATION POLICY
48	17	EFFECTIVE AUTHORIZATION AND ACCESS CONTROL MECHANISM
49		MINIMUM AUTHORIZATION POLICY
50		DUAL AUTHORIZATION REQUIRED FOR CHANGES
51		SUPER USER AUTHORIZATION REQUIRED FOR CHANGES
52		LOG OF ATTEMPTED VIOLATIONS
53	18	SELF-MODIFYING I/O ROUTINES NOT ALLOWED
54	19	DIRECTION IN PASSWORD CHOICE
55		STORE IN ENCRYPTED FORM
56		AUTOMATIC DELAY AFTER INVALID LOGIN ATTEMPT
57		ENCRYPTED TRANSMISSIONS TO TERMINALS
58		USE OF INTERACTIVE AUTHENTICATION PROCEDURE
59	20	ADEQUATE ACCESS CONTROLS
60		ADEQUATE AND ENFORCED LIBRARY FACILITY
61		USAGE LOG
62		PROPER LABELLING
63	21	PROPER SYSTEM DESIGN
64		EFFECTIVE AUTHORIZATION AND ACCESS CONTROL MECHANISM
65		ADEQUATE I/O CONTROLS
66		PROTECTION OF STATE VECTOR
67	22	STORAGE IN PROTECTED STORAGE
68	23	ADMINISTRATIVE CONTROLS
69		HUMAN VERIFICATION
70		SUPERVISION
71		LIMITED CE ACCESS
72	24	ENCRYPTION
73	25	EFFECTIVE HUMAN ENGINEERING
74		CLEAR, EASY TO USE PROTECTION FACILITIES
75		ADEQUATE DOCUMENTATION
76		USER EDUCATION
77	26	<SEE FEATURES FOR THREATS 27-39>
78	27	PRINT LOG
79		SECURITY CONSCIOUS I/O ROUTINES
80	28	PRINT LOG
81	29	PRINT LOG
82		SECURITY CONSCIOUS I/O ROUTINES
83	30 31	CAREFUL ADMINISTRATIVE PROCEDURES
84	32	CAREFUL ADMINISTRATIVE PROCEDURES
85		IMPORTANT MAIL SENT REGISTERED OR BY COURIER
86		DELIVERY CONFIRMATION
87	33	TRACE LOG OF SENSITIVE OUTPUT
88		LIBRARY FACILITY FOR SENSITIVE OUTPUT
89		<SEE ALSO FEATURES FOR THREATS 34-37>
90	34	CLEAN DESK POLICY
91		USER EDUCATION
92	35	GUARDING WORK IN TRANSIT
93	36	<REFER TO FEATURES FOR THREATS 1-13>

[illegible]	[illegible]	[illegible]
94	37	GUARDING WORK IN PROGRESS
95	38	BUFFER ERASE MECHANISM
96	39	PAPER SHREDDER
97		USE OF OLD RIBBONS FOR SENSITIVE JOBS
98		DESTRUCTION OF CARBON PAPER AND RIBBONS
99	40	<REFER TO FEATURES FOR THREATS 41-44>
100	41	CAREFUL ADMINISTRATIVE PROCEDURES
101		IMPORTANT MAIL SENT REGISTERED OR BY COURIER
102		DELIVERY CONFIRMATION
103	42 43	SECOND PERSON VERIFICATION
104		CHECKSUMS
105		SOFTWARE CHECKS
106	44	VERIFICATION CHECKS
107		CHECKSUMS
108		SOFTWARE CHECKS
109		ORIGINATOR VERIFICATION
110	45	TESTING
111		AUDIT PROGRAMS
112		TESTING AND VERIFICATION
113		PENETRATION ATTEMPTS
114	46	PROGRAM TESTING AND VALIDATION
115	47	ADEQUATE DOCUMENTATION AND DESIGN SPECS
116	48 49 50	ADEQUATE DOCUMENTATION AND DESIGN SPECS
117		PROGRAM TESTING AND VALIDATION
118		PROGRAMMER EDUCATION
119	51	PROGRAM TESTING AND VALIDATION
120		CODE INSPECTION, RECOMPILATION
121		CHOOSING WRITER WHO COULD NOT BENEFIT
122	52	TESTING ON SETUP DATA
123		CONTAINMENT OF TEST PROGRAMS
124	53	<REFER TO FEATURES FOR THREATS 54-56>
125	54	FIRE EXTINGUISHING SYSTEM
126		HEAT/SMOKE/FIRE DETECTORS WITH ALARMS
127		FIRE EXTINGUISHERS
128		AUTOMATIC EXTINGUISHING SYSTEM
129		FIRE PROTECTION MEASURES
130		FIREWALLS
131		FIREPROOF VAULT
132		FIRE DRILLS
133		FIRE PREVENTION MEASURES
134		NO SMOKING POLICY
135		USE OF FIRE RESISTANT MATERIALS
136	55	FLOOD PREVENTION MEASURES
137		ADEQUATE DRAINAGE
138		WATER SHUTOFF VALVE
139		WATER PROOF MACHINE COVERS
140		LOCATION NOT FLOOD-PRONE
141		COMPUTER ROOM NOT LOCATED IN THE BASEMENT
142		WATER PIPES NOT LOCATED DIRECTLY ABOVE THE EQUIPMENT

143	56	LOCATION NOT ON ACTIVE FAULT
144		ADEQUATE STRUCTURAL RE-ENFORCEMENT
145	57	COORDINATED PLAN WITH POLICE
146		<ALSO REFER TO FEATURES FOR THREAT NO. 1>
147	58	SMOKE DETECTORS
148		<ALSO REFER TO FEATURES FOR THREAT NO. 57>
149	59	FAVORABLE LOCATION CHOICE
150		<ALSO REFER TO FEATURES FOR THREAT NO. 57>
151	60 61	<REFER TO FEATURES FOR THREAT NO. 57>
152	62	MONITORING EQUIPMENT AND ALARM SYSTEM
153	63	PREVENTIVE MAINTENANCE
154		HARDWARE CHECKS
155	64	ADEQUATE ADMINISTRATIVE PROCEDURES
156		BACKUP FACILITIES
157	65	AUXILIARY POWER SUPPLY FOR MACHINE AND SECURITY DEVICES
158		MACHINE FEATURE FOR GRACEFUL SHUTDOWN ON POWER FAILURE
159	66	POWER SUPPLY LINE FILTER
160		VOLTAGE STABILIZER FOR POWER SUPPLY
161		MONITORING SYSTEM WITH ALARM
162	67	ELECTRICAL INSPECTION
163	68	AUXILIARY WATER SUPPLY
164		FLOW MONITOR WITH ALARM
165	69	WATER FILTERS
166	70	TEMPERATURE CONTROLLERS
167		TEMPERATURE MONITOR WITH ALARM
168	71	EMERGENCY LIGHTS
169		AUXILIARY POWER SUPPLY
170	72	ALARM SYSTEM
171		CONTINGENCY PLANS
172	73	<REFER TO FEATURES FOR THREAT NO. 54>
173	74	WATER TIGHT WINDOWS AND DOORS IN OPERATIONS AREA
174		<ALSO REFER TO FEATURES FOR THREAT NO. 55>
175	75	<REFER TO FEATURES FOR THREAT NO. 56>
176	76	<REFER TO FEATURES FOR THREAT NO. 57>
177	77	<REFER TO FEATURES FOR THREAT NO. 58>
178	78	<REFER TO FEATURES FOR THREAT NO. 59>
179	79	<REFER TO FEATURES FOR THREAT NO. 60>
180	80	<REFER TO FEATURES FOR THREAT NO. 61>
181	81	PROPER PHYSICAL AREA DESIGN AND CONSTRUCTION
182	82	BACKUP COMMUNICATIONS EQUIPMENT
[illegible]		[illegible]

183		PRACTICED CONTINGENCY PLANS
184	83 84	<REFER TO FEATURES FOR THREAT NO. 1>
185	85	ELECTRICAL SHIELDING
186		ELECTRICAL SHIELDING OF OPERATIONS AREA
187		STORAGE OF MAGNETIC MEDIA IN SHIELDING SAFES
188	86	<REFER TO FEATURES FOR THREAT NO. 1>
189	87	<REFER TO FEATURES FOR THREAT NO. 1>
190		SECURE LIBRARY FACILITIES
191		SECURE TAPE AND DISK LIBRARY
192		ONLY AUTHORIZED PERSONNEL ALLOWED TO ENTER LIBRARY
193	88	<REFER TO FEATURES FOR THREAT NO. 1>
194	89	PAPER SHREDDER
195		USE OF OLD RIBBONS WITH SENSITIVE JOBS
196		INCINERATORS
197		EMPLOYEE AWARENESS AND EDUCATION
198		SECURE DISPOSAL BINS
199	90	PAPER SHREDDER
200		INCINERATORS
201		EMPLOYEE AWARENESS AND EDUCATION
202		SECURE DISPOSAL BINS
203	91	REASONABLE AND INDUSTRY COMPARABLE SALARIES
204		REFERENCE CHECKING
205		CAREFUL SUPERVISION
206	92	REASONABLE AND INDUSTRY COMPARABLE SALARIES
207		REFERENCE CHECKING
208		CAREFUL SUPERVISION
209		EMPLOYEE MORALE PROGRAMS
210	93	PROMPT EMPLOYEE COMPLAINT HANDLING
211		<ALSO REFER TO FEATURES FOR THREAT NO. 92>
212	94	IMMEDIATE NOTICE ON LAYOFF (WITH APPROPRIATE PAY)
213		PROMPT EMPLOYEE COMPLAINT HANDLING
214		<REFER ALSO TO FEATURES FOR THREAT NO. 92>
215	95 96 97 98 99	<REFER TO FEATURES FOR THREAT NO. 92>
216	100	ADEQUATE EMPLOYEE TRAINING
217		<ALSO REFER TO FEATURES FOR THREAT NO. 92>
218	101	REFERENCE CHECKING
219		LIMIT EMPLOYEE AUTHORITY
220		NEED TO KNOW POLICY
221	102	REFERENCE CHECKING
222		CORPORATE INTELLIGENCE
223	103	ADEQUATE EMPLOYEE TRAINING
224		<ALSO REFER TO FEATURES FOR THREAT NO. 92>
225	104	USE LOG
226		LIBRARY STORAGE
227	105	USE LOG
228		LIBRARY STORAGE
229		CLEAN DESK POLICY
230	106	USE LOG

230	106	USE LOG
231		LIBRARY STORAGE
232		CLEAR CLASSIFICATION LABELLING
233		PROPER DISPOSAL
234		CLEAN DESK POLICY
235	107	CLEARLY DEFINED AUTHORIZATION FOR MODIFICATION
236		CLEAR CLASSIFICATION LABELLING
237		CLEAN DESK POLICY
238		USE LOG
239		PROTECTED LIBRARY STORAGE
240	108	GOOD COMMUNICATION SYSTEM BETWEEN THE SITES
241		SIMULATED DISASTER TESTS
242		RECIPROCAL AGREEMENTS BETWEEN COMPANIES (INCLUDES PERSONNEL)
243	109	USE OF SIMILAR EQUIPMENT FOR BACKUP (WITH PERIODIC RECHECKING)
244	110	ADEQUATE EMPLOYEE TRAINING
245		SIMULATED DISASTER TESTS
246	111	(ALSO REFER TO THE SECTION ON HARDWARE)
247	112 113	SIMULATED DISASTER TESTS
248		PROGRAM FOR BACKUP MAINTENANCE
249	114	ADEQUATE EMPLOYEE TRAINING
250		SIMULATED DISASTER TESTS
251	115	DUPLICATE DATA STORED SAFELY
252		SIMULATED DISASTER TESTS
253	116	(SEE ALSO SECTION ON SOFTWARE)
254	117	BACKUP GENERATOR AND FUEL
255	118	BACKUP STORE OF FUEL
256	119	BACKUP STORE OF OPERATIONS MATERIALS
257	120	(SEE ALSO SECTION ON THE COMPUTER CENTER)
258	121	PROPER PLANNING
259		SIMULATED DISASTER TESTS
260	122	CONTINGENCY PLANS FOR REACHING PERSONNEL AWAY FROM WORK
261		SIMULATED DISASTER TESTS
262	123	PROPER PLANNING
263		SIMULATED DISASTER TESTS
264	124	PROGRAM FOR BACKUP MAINTENANCE
265		SIMULATED DISASTER TESTS
266	125	PROPER PLANNING
267		SIMULATED DISASTER TESTS
268	126	PROGRAM FOR BACKUP MAINTENANCE
269		SIMULATED DISASTER TESTS
270		PROPER PLANNING
271	127 128	PROPER PLANNING
272		ADEQUATE EMPLOYEE TRAINING
273		SIMULATED DISASTER TESTS
274	129	(ALSO REFER TO THE SECTION ON DOCUMENTATION)

Appendix B

A Sample Run

We present here an example of the system in use. Included is:

(1) a list of the triples representing the sample installation

(2) input forms--one blank form and a set of completed forms

(3) a terminal session which illustrates the data entry process and use of the analysis functions

Following is a list of the triples representing the sample installation. The threat and feature numbers refer to the names as listed in Appendix A. The format of the triples below is:

object info : object value
threat info : threat likelihood (threat name) threat number
feature info: feature resistance (feature name) feature numbers(s)

1. Hardware

1.1 Central Machine

object info : **very high**
threat info : **medium** (unauthorized use) #8
feature info: **pretty high** (guard) #2

object info : **very high**
threat info : **pretty low** (human error) #10
feature info: **medium** (operator training, documentation) #29 30

1.2 Storage Media

object info : **high**
threat info : **high** (unauthorized read) #13
feature info: **pretty low** (encryption, system protection) #43 44

object info : **high**
threat info : **low** (theft) #11
feature info: **fairly high** (physical access controls) #31

Metering Equipment (add to hierarchy under Hardware)

object info : **low**
threat info : **low** (hardware tampering--modified operation) #4
feature info: **high** (alarmed cabinets) #21

2. Software

object info : **very high**
threat info : **medium** (unauthorized access: read/write) #16
feature info: **medium to pretty high** (authorization and access control mechanism) #46

2.1 Operating System

object info : **high**
threat info : **medium** (defective implementation) #45
feature info: **medium** (testing and verification) #112

2.2 Programs

object info : **medium**
threat info : **fairly high** (inadequate debugging) #46
feature info: **(fairly low) to medium** (testing and validation) #114

2.3 Data

object info : **high**
threat info : **high** (reading of unsecured storage media) #20
feature info: **pretty low** (library facility and use log) #60 61

object info : **high**

threat info : **medium to high** (unauthorized reading of exposed output) #33

feature info: **low** (user and employee diligence) #90 91

object info : **high**

threat info : **pretty high** (data preparation errors) #43

feature info: **high** (verification and edit checks) #103 104 105

2.3.2 Institution Data

object info : **(fairly high) to high**

threat info : **sortof low** (competitor subterfuge) #0

feature info: **low to medium** (legal recourse, employee loyalty, guards) #0

2.3.2.2 Financial Data

object info : **(fairly high) to high**

threat info : **high** (employee theft) #0

feature info: **low** (audit checks) #0

3. The Computer Center

3.1 Resource Supply Systems

object info : **very high**

threat info : **sortof low** (earthquake) #56

feature info: **low** (adequate structural reenforcement) #144

object info : **very high**

threat info : **fairly low** (fire) #54

feature info: **medium** (alarms, extinguishers) #126 127

3.2 The Building

object info : **medium**
threat info : **fairly low** (fire) #73
feature info: **medium** (alarms, extinguishers) #126 127

3.2.2.1 Computer Room

object info : **high**
threat info : **low** (magnets) #84
feature info: **(pretty low) to medium** (guards) #2

object info : **high**
threat info : **medium** (unauthorized intruders) #86
feature info: **pretty high** (guards, alarmed doors) #2 11

OBJECT NO: ____________________

ADD, A name or number ____________________

VALUE, V object value ____________________

THREAT NO | THREAT LIKELIHOOD | FEATURE NOS | FEATURE RESISTANCE

OBJECT NO: ____________________

ADD, A name or number ____________________

VALUE, V object value ____________________

THREAT NO | THREAT LIKELIHOOD | FEATURE NOS | FEATURE RESISTANCE

OBJECT NO: ____________________

ADD, A name or number ____________________

VALUE, V object value ____________________

THREAT NO | THREAT LIKELIHOOD | FEATURE NOS | FEATURE RESISTANCE

OBJECT NO: ____________________

ADD, A name or number ____________________

VALUE, V object value ____________________

THREAT NO | THREAT LIKELIHOOD | FEATURE NOS | FEATURE RESISTANCE

OBJECT NO: 1

ADD, A name or number: A METERING EQUIPMENT

VALUE, V object value

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE

OBJECT NO: 11

ADD, A name or number

VALUE, V object value: V VERY HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
8	MEDIUM	2	PRETTY HIGH
10	PRETTY LOW	29 30	MEDIUM

OBJECT NO: 12

ADD, A name or number

VALUE, V object value: V HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
13	HIGH	43 44	PRETTY LOW
11	LOW	31	FAIRLY HIGH

OBJECT NO: METERING EQUIPMENT

ADD, A name or number

VALUE, V object value: L LOW

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
4	LOW	21	HIGH

OBJECT NO: 2

ADD, A name or number

VALUE, V object value V VERY HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
16	MEDIUM	46	MEDIUM TO PRETTY HIGH

OBJECT NO: 21

ADD, A name or number

VALUE, V object value V HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
45	MEDIUM	112	MEDIUM

OBJECT NO: 22

ADD, A name or number

VALUE, V object value V MEDIUM

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
46	FAIRLY HIGH	114	(FAIRLY LOW) TO MEDIUM

OBJECT NO: 23

ADD, A name or number

VALUE, V object value V HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
20	HIGH	60 61	PRETTY LOW
33	MEDIUM TO HIGH	90 91	LOW
43	PRETTY HIGH	103 104 105	HIGH

OBJECT NO: 232

ADD, A name or number

VALUE, V object value V (FAIRLY HIGH) TO HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
0	SORTOF LOW	0	LOW TO MEDIUM

OBJECT NO: 2322

ADD, A name or number

VALUE, V object value V (FAIRLY HIGH) TO HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
0	HIGH	0	LOW

OBJECT NO: 31

ADD, A name or number

VALUE, V object value V VERY HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
56	SORTOF LOW	114	LOW
54	FAIRLY LOW	126 127	MEDIUM

OBJECT NO: 32

ADD, A name or number

VALUE, V object value V MEDIUM

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
73	FAIRLY LOW	126 127	MEDIUM

OBJECT NO: 3221

ADD, A name or number

VALUE, V object value V HIGH

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE
84	LOW	2	PRETTY LOW TO MEDIUM
86	MEDIUM	2 11	PRETTY HIGH

OBJECT NO:

ADD, A name or number

VALUE, V object value

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE

OBJECT NO:

ADD, A name or number

VALUE, V object value

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE

OBJECT NO:

ADD, A name or number

VALUE, V object value

THREAT NO	THREAT LIKELIHOOD	FEATURE NOS	FEATURE RESISTANCE

```
SECURATE
HI THERE.
PLEASE WAIT A FEW MOMENTS WHILE WE SET THINGS UP.


HI AGAIN.
ENTER THE NAME OF YOUR WORKSPACE ('NONE' FOR THE FIRST TIME):
NONE
DO YOU WANT TO USE A SYSTEM MODEL OTHER THAN THE STANDARD COMPUTER INSTALLATION MODEL? N


YOU ARE NOW ENTERING THE DATA ENTRY PHASE.

DO YOU WANT TO USE THREAT NUMBERS? Y
DO YOU WANT TO USE FEATURE NUMBERS? Y
ENTER A NAME FOR YOUR FILE: EXAMPLE
DO YOU WANT YOUR DATA TO BE ENCRYPTED WHEN IT IS FILED? Y
ENTER A PASSWORD TO BE ASSOCIATED WITH YOUR FILE:
[redacted]
YOU MUST REMEMBER THIS PASSWORD AS YOU WILL NEED TO SPECIFY IT TO ACCESS YOUR DATA AT A LATER DATE.
ENTER THE OBJECT NUMBER FOR THE NEXT OBJECT: 1
HARDWARE
: ADD METERING EQUIPMENT
METERING EQUIPMENT RECEIVED OBJECT NUMBER 71
: O
OBJECT NO 11, CENTRAL MACHINE IS NEXT.
: V VERY HIGH
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 8 MEDIUM 2 PRETTY HIGH
→ 10 PRETTY LOW 29 30 MEDIUM .
→
: N
OBJECT NO 12, STORAGE MEDIA IS NEXT.
: V HIGH
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 13 HIGH 43 44 PRETTY LOW
→ 11 LOW 31 FAIRLY HIGH
→
: N
OBJECT NO 13, COMMUNICATIONS EQUIPMENT IS NEXT.
: N
OBJECT NO 14, I/O DEVICES IS NEXT.
: N
OBJECT NO 71, METERING EQUIPMENT IS NEXT.
: V LOW
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 4 LOW 21 HIGH
→
: N
ENTER THE OBJECT NUMBER FOR THE NEXT OBJECT: 2
SOFTWARE
: V VERY HIGH
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 16 MEDIUM 46 MEDIUM TO PRETTY HIGH
→
: O
OBJECT NO 21, OPERATING SYSTEM IS NEXT.
: V HIGH
```

```
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 45 MEDIUM 112 MEDIUM
→
: N
OBJECT NO 22, PROGRAMS IS NEXT.
: V MEDIUM
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 46 FAIRLY HIGH 114 (FAIRLY LOW) TO MEDIUM
→
: N
OBJECT NO 23, DATA IS NEXT.
: V HIGH
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 20 HIGH 60 61 PRETTY LOW
→ 33 MEDIUM TO HIGH 90 91 LOW
→ 43 PRETTY HIGH 103 104 105 HIGH
→
: O
OBJECT NO 231, PERSONAL DATA IS NEXT.
: N
OBJECT NO 232, INSTITUTION DATA IS NEXT.
: V (FAIRLY HIGH) TO HIGH
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 0 SORTOF LOW 0 LOW TO MEDIUM
→
: O
OBJECT NO 2321, MARKETING DATA IS NEXT.
: N
OBJECT NO 2322, FINANCIAL DATA IS NEXT.
: V (FAIRLY HIGH) TO HIGH
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 0 HGIH 0 LOW
HGIH IS NOT A RECOGNIZABLE WORD.
NO ACTION WAS TAKEN FOR THIS ENTRY.  TRY AGAIN.
→ 0 HIGH 0 LOW
→
: N
OBJECT NO 2323, OPERATIONS DATA IS NEXT.
: N
OBJECT NO 2324, PLANNING DATA IS NEXT.
: N
OBJECT NO 2325, OTHER DATA IS NEXT.
: N
ENTER THE OBJECT NUMBER FOR THE NEXT OBJECT: 3
THE COMPUTER CENTER
: O
OBJECT NO 31, RESOURCE SUPPLY SYSTEMS IS NEXT.
: V VERY HIGH
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 56 SORTOF LOW 114 LOW
→ 54 FAIRLY LOW 126 127 MEDIUM
→
YOUR WORK IS NOW BEING SAVED.
CHECKPOINT: WORK TO THIS POINT HAS BEEN SAVED.
: N
OBJECT NO 32, THE BUILDING IS NEXT.
: V MEDIUM
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 73 FAIRLY LOW 126 127 MEDIUM
→
: O
OBJECT NO 321, THE BUILDING STRUCTURE IS NEXT.
: N
```

```
OBJECT NO 322. COMPUTER OPERATIONS AREA IS NEXT.
: O
OBJECT NO 3221. COMPUTER ROOM IS NEXT.
: V HIGH
THREAT NO  THREAT LIKELIHOOD  FEATURE NOS  FEATURE RESISTANCE
→ 84 LOW 2 (PRETTY LOW) TO MEDIUM
→ 86 MEDIUM 2 11 PRETTY HIGH
→
: N
OBJECT NO 3222. DATA RECEPTION AREA IS NEXT.
: N
OBJECT NO 3223. TAPE AND DISK LIBRARY IS NEXT.
: N
OBJECT NO 3224. CE ROOM IS NEXT.
: N
OBJECT NO 3225. DATA PREPARATION AREA IS NEXT.
: N
OBJECT NO 3226. PHYSICAL PLANT ROOM IS NEXT.
: N
OBJECT NO 3227. STATIONERY STORAGE IS NEXT.
: N
OBJECT NO 33. WASTE MATERIALS IS NEXT.
: N
ENTER THE OBJECT NUMBER FOR THE NEXT OBJECT: 0
DO YOU WANT TO ADD ANY MORE OBJECTS WHICH ARE NOT IN THE HIERARCHY? N
YOUR WORK IS NOW BEING SAVED.
CHECKPOINT: WORK TO THIS POINT HAS BEEN SAVED.
TO RECEIVE INSTRUCTIONS IN USING THE ANALYSIS FUNCTIONS, ENTER 'INSTRUCTIONS'.
```

```
 INSTRUCTIONS.
THE FOLLOWING ANALYSIS FUNCTIONS ARE AVAILABLE. TO INVOKE SIMPLY TYPE IN THE NAME

OVERALLRATING -- THIS FUNCTION WILL RATE THE ENTIRE INSTALLATION. THE RATING WILL THEN
  (ALSO ORATE)     BE PRINTED OUT

SECTIONRATINGS -- THIS FUNCTION WILL RATE THE SUBSECTIONS OF A SPECIFIED OBJECT SECTION.
  (ALSO SRATE)      FOR EXAMPLE IF HARDWARE, OBJECT 1, IS SPECIFIED, THIS FUNCTION WILL RETURN
                    RATINGS FOR EACH OF THE MAIN SUBSECTIONS OF HARDWARE: THE CENTRAL MACHINE,
                    STORAGE MEDIA, COMMUNICATIONS EQUIPMENT, AND I/O DEVICES.

INDIVIDUALRATING -- THIS FUNCTION WILL RETURN THE RATING FOR A SPECIFIED SUBSECTION OF THE HIERARCHY.
  (ALSO IRATE)
WORSTSUBSECTION -- THIS FUNCTION WILL EVALUATE THE SUBSECTIONS OF EITHER THE ENTIRE INSTALLATION OR
  (ALSO WRATE)       A SPECIFIED SUBSECTION OF THE INSTALLATION AND PRINT OUT THAT SUBSECTION WITH
                     THE LOWEST RATING.
DO YOU WANT TO SEE A DESCRIPTION OF THE RATING FUNCTIONS? Y

  THE FOLLOWING RATING FUNCTIONS ARE AVAILABLE:
    1) WEAKEST LINK
    2) SELECTED WEAKEST LINK
    3) FUZZY MEAN
    4) FUZZY MEAN WEIGHTED BY VALUE
    5) FUZZY MEAN WITH EACH MAJOR SUBSECTION WEIGHTED BY MAXIMUM OBJECT VALUE

ENTER THE NUMBER OF THE RATING FUNCTION YOU WISH TO USE: 3
```

```
DISPLAY.

FOLLOWING IS A LIST OF OBJECTS ADDED, THEIR ASSIGNED OBJECT
NUMBERS, AND THEIR PARENT IN THE HIERARCHY:
OBJECT             OBJECT NO  PARENT
METERING EQUIPMENT  71         1

                   OBJECTS                              THREATS                                  FEATURES

TRIPLE * NUMBER          NAME              * NUMBER          NAME                  * NUMBER           NAME
 NO    *                VALUE              *               LIKELIHOOD              *               RESISTANCE
***************************************************************************************************************************
       ***                                 ***                                     ***
1      * 11     CENTRAL MACHINE            * 8      UNAUTHORIZED USE               * 2    GUARD
       *  VERY HIGH                        *   MEDIUM                              *  PRETTY HIGH
       ***                                 ***                                     ***
2      * 11     CENTRAL MACHINE            * 10     HUMAN ERROR                    * 29   OPERATOR TRAINING
       *                                   *                                       * 30   DETAILLED, ACCURATE, ACCESSIBL
       *  VERY HIGH                        *   PRETTY LOW                          *  MEDIUM
       ***                                 ***                                     ***
3      * 12     STORAGE MEDIA              * 13     UNAUTHORIZED READ              * 43   DATA ENCRYPTION
       *                                   *                                       * 44   EFFECTIVE STORAGE ACCESS CONTR
       *  HIGH                             *   HIGH                                *  PRETTY LOW
       ***                                 ***                                     ***
4      * 12     STORAGE MEDIA              * 11     THEFT                          * 31   PHYSICAL ACCESS CONTROLS
       *  HIGH                             *   LOW                                 *  FAIRLY HIGH
       ***                                 ***                                     ***
5      * 71     METERING EQUIPMENT         * 4      HARDWARE TAMPERING--MODIFIED O* 21   LOCKS AND ALARMS ON MACHINE CO
       *  LOW                              *   LOW                                 *  HIGH
       ***                                 ***                                     ***
6      * 2      SOFTWARE                   * 16     UNAUTHORIZED ACCESS--R/W/E     * 46   EFFECTIVE AUTHORIZATION AND AC
       *  VERY HIGH                        *   MEDIUM                              *  MEDIUM TO PRETTY HIGH
       ***                                 ***                                     ***
7      * 21     OPERATING SYSTEM           * 45     DEFECTIVE IMPLEMENTATION       * 112  TESTING AND VERIFICATION
       *  HIGH                             *   MEDIUM                              *  MEDIUM
       ***                                 ***                                     ***
8      * 22     PROGRAMS                   * 46     INADEQUATE DEBUGGING           * 114  PROGRAM TESTING AND VALIDATION
       *  MEDIUM                           *   FAIRLY HIGH                         *  (FAIRLY LOW) TO MEDIUM
       ***                                 ***                                     ***
9      * 23     DATA                       * 20     UNSECURED STORAGE MEDIA        * 60   ADEQUATE AND ENFORCED LIBRARY
       *                                   *                                       * 61   USAGE LOG
       *  HIGH                             *   HIGH                                *  PRETTY LOW
       ***                                 ***                                     ***
10     * 23     DATA                       * 33     EXPOSED OUTPUT                 * 90   CLEAN DESK POLICY
       *                                   *                                       * 91   USER EDUCATION
       *  HIGH                             *   MEDIUM TO HIGH                      *  LOW
       ***                                 ***                                     ***
11     * 23     DATA                       * 43     DATA PREPARATION ERRORS        * 103  SECOND PERSON VERIFICATION
       *                                   *                                       * 104  CHECKSUMS
       *                                   *                                       * 105  SOFTWARE CHECKS
       *  HIGH                             *   PRETTY HIGH                         *  HIGH
       ***                                 ***                                     ***
12     * 232    INSTITUTION DATA           * 0                                     * 0
       *  FAIRLY HIGH TO HIGH              *   SORTOF LOW                          *  LOW TO MEDIUM
       ***                                 ***                                     ***
13     * 2322   FINANCIAL DATA             * 0                                     * 0
       *  FAIRLY HIGH TO HIGH              *   HIGH                                *  LOW
       ***                                 ***                                     ***
14     * 31     RESOURCE SUPPLY SYSTEMS    * 56     EARTHQUAKE                     * 144  ADEQUATE STRUCTURAL RE-ENFORCE
       *  VERY HIGH                        *   SORTOF LOW                          *  LOW
       ***                                 ***                                     ***
```

```
15    * 31     RESOURCE SUPPLY SYSTEMS     * 54     FIRE            * 126  HEAT/SMOKE/FIRE DETECTORS WITH
      *                                    *                        * 127  FIRE EXTINGUISHERS
      *  VERY HIGH                         *   FAIRLY LOW           *  MEDIUM
     ***                                  ***                      ***
16    * 32     THE BUILDING                * 73     FIRE            * 126  HEAT/SMOKE/FIRE DETECTORS WITH
      *                                    *                        * 127  FIRE EXTINGUISHERS
      *  MEDIUM                            *   FAIRLY LOW           *  MEDIUM
     ***                                  ***                      ***
17    * 3221   COMPUTER ROOM               * 84     MAGNETS         * 2    GUARD
      *  HIGH                              *   LOW                  *  (PRETTY LOW) TO MEDIUM
     ***                                  ***                      ***
18    * 3221   COMPUTER ROOM               * 86     UNAUTHORIZED INTRUDERS   * 2    GUARD
      *                                    *                        * 11   ALARM SYSTEM
      *  HIGH                              *   MEDIUM               *  PRETTY HIGH
```

```
     RATESET
DO YOU WANT TO SEE A DESCRIPTION OF THE RATING FUNCTIONS? Y

   THE FOLLOWING RATING FUNCTIONS ARE AVAILABLE:
     1) WEAKEST LINK
     2) SELECTED WEAKEST LINK
     3) FUZZY MEAN
     4) FUZZY MEAN WEIGHTED BY VALUE
     5) FUZZY MEAN WITH EACH MAJOR SUBSECTION WEIGHTED BY MAXIMUM OBJECT VALUE

ENTER THE NUMBER OF THE RATING FUNCTION YOU WISH TO USE: 1
     OVERALLRATING


************************************************************************
*
*  NAME                   RATING (USING WEAKEST LINK)
*
*  THE INSTALLATION          LOW
*
************************************************************************


     RATESET
DO YOU WANT TO SEE A DESCRIPTION OF THE RATING FUNCTIONS? N
ENTER THE NUMBER OF THE RATING FUNCTION YOU WISH TO USE: 2
     SECTIONALRATING
ENTER THE PARENT OBJECT NUMBER (0 FOR THE TOP LEVEL IN THE HIERARCHY): 0
SPECIFY MINIMUM FOR HARDWARE : MEDIUM
4 ELEMENT(S) USED
SPECIFY MINIMUM FOR SOFTWARE : HIGH
1 ELEMENT(S) USED
SPECIFY MINIMUM FOR THE COMPUTER CENTER : PRETTY HIGH
4 ELEMENT(S) USED


************************************************************************
*
*  NAME                   RATING (USING SELECTED WEAKEST LINK)
*
*  HARDWARE                  PRETTY LOW
*  SOFTWARE                  PRETTY HIGH
*  THE COMPUTER CENTER       PRETTY HIGH
*
************************************************************************


     SETRATE 1
     SRATE
ENTER THE PARENT OBJECT NUMBER (0 FOR THE TOP LEVEL IN THE HIERARCHY): 0


************************************************************************
*
*  NAME                   RATING (USING WEAKEST LINK)
*
*  HARDWARE                  PRETTY LOW
*  SOFTWARE                  LOW
*  THE COMPUTER CENTER       LOW
*
```

```
***********************************************************************

     SETRATE 3
     ORATE


***********************************************************************
*
*  NAME                      RATING (USING FUZZY MEAN)
*
*  THE INSTALLATION             EXTREMELY MEDIUM
*
***********************************************************************


     WORSTSUBSECTION
ENTER THE PARENT OBJECT NUMBER (0 FOR THE TOP LEVEL IN THE HIERARCHY): 0


***********************************************************************
*
*  NAME                      RATING (USING FUZZY MEAN)
*
*  HARDWARE                     ((SLIGHTLY LOWER ) THAN FAIRLY HIGH )AND (SLIGHTLY HIGHER ) THAN SORTOF HIGH
*  SOFTWARE                     SORTOF MEDIUM
*  THE COMPUTER CENTER          VERY MEDIUM
*
*  THE LOWEST RATING WAS GIVEN TO:
*       SOFTWARE
*
***********************************************************************


     WRATE
ENTER THE PARENT OBJECT NUMBER (0 FOR THE TOP LEVEL IN THE HIERARCHY): 2


***********************************************************************
*
*  NAME                      RATING (USING FUZZY MEAN)
*
*  OPERATING SYSTEM             MOREORLESS MEDIUM
*  PROGRAMS                     MOREORLESS MEDIUM
*  DATA                         SORTOF MEDIUM
*
*  THE LOWEST RATING WAS GIVEN TO:
*       DATA
*
***********************************************************************


     SETRATE 4
     WRATE
ENTER THE PARENT OBJECT NUMBER (0 FOR THE TOP LEVEL IN THE HIERARCHY): 2
```

```
**************************************************************************
*
*  NAME                      RATING (USING FUZZY MEAN WEIGHTED BY VALUE)
*
*  OPERATING SYSTEM            (MOREORLESS MEDIUM ) TO (SORTOF HIGH )
*  PROGRAMS                    MOREORLESS MEDIUM
*  DATA                        SORTOF MEDIUM
*
*  THE LOWEST RATING WAS GIVEN TO:
*       DATA
*
**************************************************************************
```

```
     MODTRIP
ENTER THE TRIPLE NUMBER: 10
ENTER THE NUMBER OF THE CATEGORY TO BE MODIFIED-
     1) OBJECT NUMBER
     2) THREAT NUMBER
     3) FEATURE NUMBER(S)
     4) OBJECT VALUE
     5) THREAT LIKLIHOOD
     6) FEATURE RESISTANCE
: 6
ENTER THE NEW FEATURE RESISTANCE: MEDIUM

     DISPLAY


FOLLOWING IS A LIST OF OBJECTS ADDED, THEIR ASSIGNED OBJECT
NUMBERS, AND THEIR PARENT IN THE HIERARCHY:
OBJECT             OBJECT NO  PARENT
METERING EQUIPMENT  71         1

                 OBJECTS                              THREATS                                   FEATURES

TRIPLE * NUMBER           NAME         * NUMBER        NAME                      * NUMBER           NAME
 NO    *                VALUE          *            . LIKELIHOOD                 *               RESISTANCE
*******************************************************************************************************
     ***                                ***                                       ***
1    * 11     CENTRAL MACHINE           * 8      UNAUTHORIZED USE                 * 2    GUARD
     *  VERY HIGH                       *   MEDIUM                                *  PRETTY HIGH
     ***                                ***                                       ***
2    * 11     CENTRAL MACHINE           * 10     HUMAN ERROR                      * 29   OPERATOR TRAINING
     *                                  *                                         * 30   DETAILLED, ACCURATE, ACCESSIBL
     *  VERY HIGH                       *   PRETTY LOW                            *  MEDIUM
     ***                                ***                                       ***
3    * 12     STORAGE MEDIA             * 13     UNAUTHORIZED READ                * 43   DATA ENCRYPTION
     *                                  *                                         * 44   EFFECTIVE STORAGE ACCESS CONTR
     *  HIGH                            *   HIGH                                  *  PRETTY LOW
     ***                                ***                                       ***
4    * 12     STORAGE MEDIA             * 11     THEFT                            * 31   PHYSICAL ACCESS CONTROLS
     *  HIGH                            *   LOW                                   *  FAIRLY HIGH
     ***                                ***                                       ***
5    * 71     METERING EQUIPMENT        * 4      HARDWARE TAMPERING--MODIFIED O* 21   LOCKS AND ALARMS ON MACHINE CO
     *  LOW                             *   LOW                                   *  HIGH
     ***                                ***                                       ***
6    * 2      SOFTWARE                  * 16     UNAUTHORIZED ACCESS--R/W/E       * 46   EFFECTIVE AUTHORIZATION AND AC
     *  VERY HIGH                       *   MEDIUM                                *  MEDIUM TO PRETTY HIGH
     ***                                ***                                       ***
7    * 21     OPERATING SYSTEM          * 45     DEFECTIVE IMPLEMENTATION         * 112  TESTING AND VERIFICATION
     *  HIGH                            *   MEDIUM                                *  MEDIUM
     ***                                ***                                       ***
8    * 22     PROGRAMS                  * 46     INADEQUATE DEBUGGING             * 114  PROGRAM TESTING AND VALIDATION
     *  MEDIUM                          *   FAIRLY HIGH                           *  (FAIRLY LOW) TO MEDIUM
     ***                                ***                                       ***
9    * 23     DATA                      * 20     UNSECURED STORAGE MEDIA          * 60   ADEQUATE AND ENFORCED LIBRARY
     *                                  *                                         * 61   USAGE LOG
     *  HIGH                            *   HIGH                                  *  PRETTY LOW
     ***                                ***                                       ***
10   * 23     DATA                      * 33     EXPOSED OUTPUT                   * 90   CLEAN DESK POLICY
     *                                  *                                         * 91   USER EDUCATION
     *  HIGH                            *   MEDIUM TO HIGH                        *  MEDIUM
     ***                                ***                                       ***
```

```
11      * 23     DATA                    * 43     DATA PREPARATION ERRORS   * 103  SECOND PERSON VERIFICATION
        *                                *                                  * 104  CHECKSUMS
        *                                *                                  * 105  SOFTWARE CHECKS
        *  HIGH                          *   PRETTY HIGH                    *  HIGH
      ***                              ***                                ***
12      * 232    INSTITUTION DATA        * 0                                * 0
        *  FAIRLY HIGH TO HIGH           *   SORTOF LOW                     *  LOW TO MEDIUM
      ***                              ***                                ***
13      * 2322   FINANCIAL DATA          * 0                                * 0
        *  FAIRLY HIGH TO HIGH           *   HIGH                           *  LOW
      ***                              ***                                ***
14      * 31     RESOURCE SUPPLY SYSTEMS * 56     EARTHQUAKE                * 144  ADEQUATE STRUCTURAL RE-ENFORCE
        *  VERY HIGH                     *   SORTOF LOW                     *  LOW
      ***                              ***                                ***
15      * 31     RESOURCE SUPPLY SYSTEMS * 54     FIRE                      * 126  HEAT/SMOKE/FIRE DETECTORS WITH
        *                                *                                  * 127  FIRE EXTINGUISHERS
        *  VERY HIGH                     *   FAIRLY LOW                     *  MEDIUM
      ***                              ***                                ***
16      * 32     THE BUILDING            * 73     FIRE                      * 126  HEAT/SMOKE/FIRE DETECTORS WITH
        *                                *                                  * 127  FIRE EXTINGUISHERS
        *  MEDIUM                        *   FAIRLY LOW                     *  MEDIUM
      ***                              ***                                ***
17      * 3221   COMPUTER ROOM           * 84     MAGNETS                   * 2    GUARD
        *  HIGH                          *   LOW                            *  (PRETTY LOW) TO MEDIUM
      ***                              ***                                ***
18      * 3221   COMPUTER ROOM           * 86     UNAUTHORIZED INTRUDERS    * 2    GUARD
        *                                *                                  * 11   ALARM SYSTEM
        *  HIGH                          *   MEDIUM                         *  PRETTY HIGH


      SETRATE 3

      WRATE
ENTER THE PARENT OBJECT NUMBER (0 FOR THE TOP LEVEL IN THE HIERARCHY): 2


*************************************************************************
*
*  NAME                     RATING (USING FUZZY MEAN)
*
*  OPERATING SYSTEM            MOREORLESS MEDIUM
*  PROGRAMS                    MOREORLESS MEDIUM
*  DATA                        SORTOF MEDIUM
*
*  THE LOWEST RATING WAS GIVEN TO:
*       DATA
*
*************************************************************************


      MODTRIP
ENTER THE TRIPLE NUMBER: 9
ENTER THE NUMBER OF THE CATEGORY TO BE MODIFIED-
     1) OBJECT NUMBER
     2) THREAT NUMBER
     3) FEATURE NUMBER(S)
     4) OBJECT VALUE
     5) THREAT LIKLIHOOD
     6) FEATURE RESISTANCE
: 6
ENTER THE NEW FEATURE RESISTANCE: MEDIUM
```

```
WRATE
ENTER THE PARENT OBJECT NUMBER (0 FOR THE TOP LEVEL IN THE HIERARCHY): 2


************************************************************************
*
*  NAME                       RATING (USING FUZZY MEAN)
*
*  OPERATING SYSTEM              MOREORLESS MEDIUM
*  PROGRAMS                      MOREORLESS MEDIUM
*  DATA                          MEDIUM
*
*  THE LOWEST RATING WAS GIVEN TO:
*       OPERATING SYSTEM
*       PROGRAMS
*
************************************************************************
```