

Activity Profiles for Intrusion Detection

L.J. Kohout, A. Yasinsac, E. McDuffie

Abstract— This paper one of the series of papers in which we lay down the foundations for design of trustworthy computing systems. We propose here a methodology to perform behavior-based intrusion detection on security protocols using fuzzy relational methods. Fuzzified Allen temporal algebra plays an important role in capturing the time dimension of concurrent activities of principals as well as of intruders.

Keywords— Intrusion detection, Temporal logic, Allen Algebra, Fuzzy relations, Internet.

I. INTRODUCTION

Security of Internet traffic depends on cryptography implemented through security protocols. These protocols are vulnerable to subtle flaws that manifest as attacks on privacy, integrity, and denial of service. Fortunately, Intrusion Detection methods offer an opportunity to improve our confidence in Internet traffic. To date, efforts to develop online security protocol attack detection have focused mostly on utilizing misuse detection. In this paper, we propose a methodology to perform behavior-based intrusion detection on security protocols using fuzzy relational methods.

Our methods of behavior-based dynamic computer protection provide the means that could contribute towards solving intrusion detection problem. This paper one of the series of papers in which we lay down the foundations for fuzzy logic based foundations of design of trustworthy computing systems. The present paper is based on the idea that intrusions may be recognized in the security protocols that these hosts routinely employ [20].

After highlighting in Sec II some aspects of the currently used intrusion detection approaches that are relevant to behavior-based intrusion detection we move in Sec. III to describing an Activity Structures based approach to dynamic protection of distributed information processing systems. In Sec. VI we describe a generic architecture for behavior-based intrusion detection. An architecture for gathering protocol activity in a secure enclave, that was described by Yasinsac [20] and is presently under construction in the Security and Assurance in Information Technology (SAIT) Laboratory at Florida State University forms the core which is explored by the extension using fuzzy relational computing methods.

II. CURRENT APPROACHES TO INTRUSION DETECTION

A. Denning Model

For recognizing anomalies one needs an anomaly detection model. In her seminal paper [3], Dorothy Denning outlined the process of profiling used to detect intrusions. She also detailed several statistical measures that can be

used to represent normal behavior in a computer system. Her paper serves as the foundation of most of the work that has occurred since, and we employ the conceptual structure of her model as the epistemological framework here. Because she proposes only a model, her work does not address the functional challenges to implementing such an IDS. Our work goes beyond the model to a more detailed architecture pertinent to a specific detection environment.

There are six components to the Denning model: Subjects, Objects, Audit Records, Profiles, Anomaly Records, and Activity Rules. Security protocol environments are well understood in the security community. They consist of subjects similar to those in the Denning model. More specifically, security protocols occur in sessions, each consisting of at least two subjects that are parties to the protocol. Objects in this environment are the security protocols themselves. Events are generated when subjects participate in security protocol sessions by sending and receiving protocol messages.

Audit records are generated based on the events, metrics, and the random variables that represent behavior in this environment. Sending and receiving messages in a security protocol session are events. Interpretation of these events results in generation of an audit record. For example, receipt of an "Unexpected Termination" event will result in recording of a "Incomplete Session" audit record.

Profile elements must contain enough information to recognize normal behavior, but must also be sufficiently rigorous to distinguish normal behavior from malicious activity within the system.

There are many such telling items within security protocols. Normal behavior of a security protocol includes activity of principals, requesting:

- secure sessions with other principals,
- public keys from trusted third parties,
- signature verification from trusted third parties, and
- suspending and re-establishing a communication session.

Profile entries consist of numerical entries that represent the recorded events for each principal. For example, there are entries to keep track of the number of sessions that each principal has started in each of several time intervals as well as the recorded average, minimum, maximum, and standard deviation of corresponding counts for each principal.

Activity Records contain context-specific information to allow appropriate response to the abnormality that was detected.

B. Statistical Methods for Anomaly Detection

One can distinguish normal behavior from abnormal behavior that has a high likelihood of being malicious. This is indicated by the types of measures that are applied to

The authors are with Dept. of Computer Science, Florida State University, Tallahassee, Florida 32306-4530, USA. E-mail: kohout@cs.fsu.edu

these profile elements. As with all behavior-based intrusion detection systems, we assume that, after a reasonable break-in period, user behavior with respect to our analysis environment (security protocols) will achieve an equilibrium, with few natural, significant deviations.

Another behavior category is the number of sessions that each principal is involved in, either as a protocol originator, a desired party of a protocol session initiated another principal, or as a trusted third party. We can increase the amount of data we track regarding these sessions, counting, for example, the number of sessions that each principal originates with every other principal. Over a reasonable duration of learning, these counters will establish patterns that allow our detection system to predict behavior, or more importantly, recognize or notice when abnormal changes in the behavior occur.

We utilize two metrics proposed by Dr. Denning to model behavior: Event Counters and Interval Timers. Simply put, we count the number of occurrences of a particular event within some predetermined timeframe and compare that count against the user profile. If the count differs from the normal use parameter by some threshold, an anomaly is reported.

We examine a set of random variables that reflect the characteristics of security protocols. For example, an important finding of cryptographic protocol verification research is that many attacks on protocols are possible only when intruders can leverage concurrent sessions by one or more principals. This is particularly true when at least one key is used in two concurrent sessions. In fact, Kelsey et al. [6] show how concurrent, or parallel, session attacks can be constructed on arbitrary security protocols. Thus, the number of concurrent sessions that principals engage is pertinent information in partitioning appropriate from malicious behavior. Here one does not deal with singletons to which probability theory assigns the degrees of probability but also with subsets of events. At least Dempster-Shafer or more general non-additive fuzzy measures may be required. The usefulness of fuzzy probabilities in this context should also be empirically tested.

III. USING FUZZY RELATIONAL METHODS TO DETECT ATTACKS ON SECURITYPROTOCOLS

The previous discussion focuses on a method for applying probability-based statistical methods to events to detect abnormal behavior. Probability theory based statistics, however, does not capture adequately those anomalies that are best described by non-probabilistic measures. We now turn to fuzzy methods for classification of abnormal event strings.

A. Representing Normal and Abnormal Behavior

In the most general terms we are concerned with application of fuzzy logics and relational computational algorithms to forming activity profiles in distributed information processing systems in order to distinguish desirable from undesirable activities. We apply BK-products of relations and fuzzy measures (which subsume proba-

bility as a special case). Fuzzy logics allow us to create possibility profiles and usability profiles and combine these with probability profiles when necessary to detect unusual or abnormal activities. Fast Fuzzy Relational Algorithms that can be executed on conventional computing architectures as well as soft computing architectures, in particular Neuro-Fuzzy Networks can be used for computations. The BK-nonassociative relational products in the feedback loop used instead on the traditional back-propagation speed up training of the neuro-fuzzy networks considerably.

B. Intrusion Detection Issues in Computing Systems

As noted earlier, our target environment for this intrusion detection environment is logically secure enclaves that are protected by encryption and that experience a high incidence of security protocol activity. A security protocol can be viewed as a set of transactions between computer network users and system resources. Specific sets of transactions define specific security protocols.

This environment is well suited for fuzzy relational analysis because security protocols are symbolic patterns. Not only are the protocols patterns but security attacks can also be viewed as patterns to be recognized. In fact, all user activity over a network can be viewed as patterns. Being able to determine whether a given behavioral pattern is normal or abnormal, a security protocol or an attack, is key to the problem of security protocol attack detection. This cannot be dealt with without having an adequate methodological framework for dealing in a trustworthy way with activities of distributed families of processes and agents.

C. Designing Trustworthy Distributed Computations

With the advent of distributed information technology and emergence of agent based computational paradigm new problems arise. In such frameworks, one has to deal with embedding of streams of activities of an agent (or a group of agents) within a wider distributed community of agents that cooperate or compete. In order to design and implement trustworthy systems in this context, one has to put adequate constraints on the streams of activities of communities of agents. This cannot be done in an haphazard way but requires a methodology that can correctly interleave desirable activities and effectively suppress undesirable activities. One has to take taking into consideration not only statical, invariant dependencies of actions and activities of interacting agents but also the dynamics of actions.

IV. DYNAMIC PROTECTION OF COMMUNITIES OF AGENTS

A. Cooperating and competing agents

An agent is not usually isolated, but is in interaction with other agents of a distributed system and with the environment.

Interaction of processes can be co-operative or competitive. Both co-operative and competitive interaction require:

- 1) communication,

- 2) co-ordination,
- 3) adherence to some rules of behaviour that guarantee correct interaction,
- 4) adequate characterisation of undesirable activities.

B. Protection and Intention Structures

An adequate dynamic protection mechanism is needed to guard an agent against the undesirable activities of other agents who may not adhere to the specified rules. This can be provided by **protection** and **intention structures** [8],[12],[13]. This knowledge is used in order to link and compare the protection rules with intentions of agents. To be able to provide such a link we need to refine the notion of *task*.

Definition: *task descriptors*.

Let γ be an activity leading to a certain aim or having a particular purpose. A task descriptor T_γ , associated with the activity γ , performing a set of tasks T , is a partition of T into two disjoint subsets

$$T_\gamma = (T_p, T_f)$$

where T_p is the subset of *permitted* tasks and T_f the subset of *forbidden* tasks, of T .

Care should be taken to insure that forbidden tasks are not realized. This leads to another important distinction, separating the notion of *protection* and *permission* structures ([8], ch. 5,6 and 7).

Permission structure is a mechanism for avoidance of disastrous consequences. For example in Intelligent Systems [8] the role of this structure is twofold:

1. If the predicted behavioral state belongs to the forbidden region, a rapid action should be taken, which brings the ‘intelligent’ system into the permitted region of the world model state space.
2. Only such intentions should be converted into actions, which can not take system into any state that belongs to the forbidden region.

C. A Brief History of Activity Structures Based Fuzzy Dynamic Protection Structures

The definition of task descriptors was first presented in [7]. A task in this context was defined as a concurrent family of activities that were generated by appropriate families of concurrent processes [7],[8]). This led to the definition of dynamic protection structures.

The concept of *Dynamic Protection Structure* was first outlined in [12],[13]. Therein Kohout and Gaines incorporated the original *Protection Matrix* model of G.S. Graham and Peter Denning [4] retaining *capabilities*, but substantially extending it by introducing *passes* and *permits*. Passes and permits qualitatively change the nature of the conceptual model by providing the descriptive semantics that is sufficient for capturing the dynamic nature of distributed computing. This conceptual model of Kohout and Gaines was expressed more formally by an automaton with three-levels of rewriting rules — at the levels of *capabilities*, *passes* and *permits* thus providing the CPP dynamic protection formal model.

Full relational formalization of the protection structures provided by Bandler and Kohout has further been fuzzified in [2],[10],[11] The structural/algorithmic/logical aspects of protected dynamic interactions of agents have been developed in considerable detail [12],[13],[7],[2],[10], [11]. The concept of *connected protection structures* important for distributed computing was introduced and further elaborated in [17],[14].

D. Activity Structures

The framework of dynamic protection is linked to one of the behavioral categories of Activity Structures [9], chapter 5. Activity Structures framework (AS) has 6 functional behavioral structures [8, Chapter 10].

- *Information Handling Structures* (IH)

IHC – control structure

IHD – domain-of-expertise knowledge structure

IHI – inference structure

- *Constraints Structures* (C)

CP – protection structure

CM – man-computer interaction structure

CT – technological designs and implementation constraints structure

Recognizing what are the main behavioral categories is indispensable for recognizing and specifying what are undesirable activities in any distributed information processing system. Connected Protection Structures (CPSs) [17], cite00.8 decouple activities within a connecting system from activities performed across multiple connecting systems. As in other functional structures, CPSs employ concepts of *participants* and *actions*. The decoupling is achieved by differentiating actions into the following two types.

- *internal* actions – actions performed by participants onto participants of the same system, and
- *external* actions – action performed by participants of one system onto participants of different systems.

Activity Structures have been applied to analysis of fuzzy cognitive maps describing desirable and undesirable activities in problem solving behaviour [5]. This is clearly relevant to simulation of intentions of intruders, but that is outside the scope of the present paper. The temporal pattern and its concurrent interaction of the Activity Structures functional behavior categories and indeed of any activities is, however, highly relevant here. Interval methods for representing such temporal patterns of activities are discussed in the next section.

V. REPRESENTING TEMPORAL PATTERNS OF BEHAVIOR

There are many temporal representations that have been developed and studied. Prominent in AI and real time systems are temporal modal logics. Early proponents of the use of temporal modalities in dynamic protection of computing agent behavior were Kohout and Gaines. That approach is useful in design of well-protected distributed systems. In intrusion detection, we have uncertainties of a higher order, hence interval systems that can be linked to

computing with words and computing with perception are preferable.

A. Allen's Temporal Interval relations

This section focuses on Allen's temporal interval relations [1]. This type of relation has appeared in a number of different automatic scheduling systems in manufacturing and also Artificial Intelligence based automatic scheduling systems. Other temporal representation often compare themselves or build on Allen's work.

There are 13 different Allen temporal relations. These relations cover all the possible binary interval relations that can exist between two time intervals. The following is a description of these relations:

1 BEFORE \Leftrightarrow an interval a must start and end before an interval b starts

$$| - - a - - | | - - b - - |$$

2 AFTER \Leftrightarrow the inverse of the BEFORE relation

3 DURING \Leftrightarrow an interval a must start after the beginning of an interval b and end before the end of interval b

$$| - - a - - | \\ | - - - - b - - - - |$$

4 DURING-BY \Leftrightarrow the inverse of the DURING relation

5 FINISH \Leftrightarrow intervals a and b must both end at the same time

$$| - - - - a - - - - | \\ | - - - - - b - - - - - |$$

6 FINISH-BY \Leftrightarrow the inverse of the FINISH relation

MEET \Leftrightarrow an interval a must end at the same time that an interval b begins

$$| - - a - - | - - - b - - - |$$

7 MEET-BY \Leftrightarrow the inverse of the MEET relation

8 START \Leftrightarrow intervals a and b must both start at the same time

$$| - - a - - | \\ | - - - b - - - |$$

9 START-BY \Leftrightarrow the inverse of the START relation

10 OVERLAP \Leftrightarrow an interval a must end after an interval b has begun and before it ends

$$| - - - a - - - | \\ | - - - - b - - - - |$$

11 OVERLAP-BY \Leftrightarrow the inverse of the OVERLAP relation

12 EQUAL \Leftrightarrow intervals a and b must both start and end at the same time

$$| - - - a - - - | \\ | - - - b - - - |$$

This type of representation is called IA for *interval algebra*. If events are considered instead of tasks with duration then the resulting representation is called PA for point algebra. Since events can occur at a specific point in time and a task is normally thought of as having some duration in time, it becomes clear that a complete system should be able to represent both types of temporal possibilities.

B. Recursive Allen Temporal Algebra RATA

Two of the main concepts used in application of Allen Temporal Relations to the description of concurrent activ-

ities of processes are refinements introduced by McDuffie [16] namely Recursive Allen Temporal Algebra (RATA) and Temporal Dependent Interval Calculus (TDIC). RATA builds on Allen's relations by allowing the parameters of the operations to also be operations. For example:

MEETS(STSRSTS(FINISH(i,DURING(j,k)), MEETS(l,m)),n)

is a valid expression in RATA. By allowing for the recursive calling of operations very complex temporal relations can be formed.

C. Fuzzification

In this work we look at the possibility of the fuzzification of the RATA operators themselves. The claim is made that the thirteen Allen relations represent all the possible binary relations that two time intervals can assume. By way of recursion, RATA extends Allen to cover all possible temporal relation between any number of time intervals. The fuzzification of RATA further extends Allen by adding mathematical rigor and providing a new class of possible temporal manipulations [15].

C.1 A Fuzzy Nature of Time Representation

The inverse relations represent a simple re-ordering of the time intervals as they are applied to the given temporal operation. For example: Before(x,y) \equiv Before⁻¹(y,x). By applying the ideas of both fuzzy subsets and power-sets to these relations (that is to consider all the possible temporal relations between two time intervals as a fuzzy powerset with both crisp and fuzzy subsets with in) then the proper a-cuts will results in the determination of the proper fuzzy subsets. Clearly, the Overlap relation is fuzzy by nature, since it does not indicate to what extent the two time intervals are overlapped. Similar considerations apply to other Allen temporal relations.

C.2 Approximation of Fuzzy Allen Temporal Relations by Fuzzy Trapezoidal Membership Function

General fuzzy Allen temporal intervals can be made more explicit by specific representations of the membership function of fuzzified temporal intervals. The simplest fuzzy computational representation is the approximation of a fuzzy temporal Allen interval by a trapezoidal membership function:

$$\mu_A(x) = \begin{cases} 0 & \text{when } -\infty < x < a \\ (x-a)/(b-a) & \text{when } a < x \leq b \\ (d-x)/(d-c) & \text{when } c \leq x < d \\ 0 & \text{when } d < x < \infty \end{cases}$$

This trapezoidal membership function has the ascending segment $\nearrow(a,b)$ and the descending segment $\searrow(c,d)$. Slopes of the *left ascending* segment and the *right descending* segment of the trapezoidal membership function are determined by the formulas:

$$k_l = h/\Delta_l; \quad k_r = h/\Delta_r$$

where h is the height of the fuzzy membership function defining the fuzzy Allen time interval; Δ_l and Δ_r are the

x coordinates of the ascending segment $\nearrow(a, b)$ and the descending segment $\searrow(c, d)$, respectively.

By using the fuzzy techniques already described, it is possible to construct a complete model for the temporal transition between crisp and fuzzy, binary and recursive temporal relations. This computational model will also prove to be very useful when applied to complex scheduling problems in a number of different domains.

C.3 Qualitative Changes of Time Intervals

Fuzzy extension of Allen temporal relations bring in an entirely new dimension, namely dynamic transformation of elastic constraints determining fuzzy Allen relations. This transformation of elastic constraints allows us to represent qualitative changes, metamorphosis of one Allen temporal operator into another one.

Take for example two time intervals of different durations. Such that $|x|$ is the duration of time interval x and $|x| < |y|$ indicates that the duration of time interval x is shorter than that of y . If the start time of interval y is held constant while the start time of interval x is allowed to change from much earlier than that of y to much later than that of time interval y , then the two time intervals will transition through all the six basic relations plus a number of their inverses as follows:

BEFORE \rightarrow MEETS \rightarrow OVERLAP \rightarrow STARTS \rightarrow DURING \rightarrow FINISHES \rightarrow OVERLAP $^{-1}$ \rightarrow MEETS $^{-1}$ \rightarrow BEFORE $^{-1}$.

In order to represent these qualitative changes adequately by trapezoidal approximations introduced above, we need to introduce some additional concepts. We capture the increment of change by a state transition. We shall call this *transformational increment* of a variable. The new state after some infinitesimal δx or finite Δx transition of a variable x will be denoted by the same name but distinguished by decoration x' (prime) as it is done for example in Z-notation in software engineering where we deal with promotion, preconditions, refinements, etc.

The sequence of transformations BEFORE \rightarrow MEETS \rightarrow OVERLAP... etc. given above can be approximated by fuzzy trapezoidal membership function with variable slope. In order to do this, we need to introduce the slope change. A new slope obtained by a transformational increment is

$$k'_l = h/\Delta'_l; \quad k'_r = h/\Delta'_r$$

From this one can work out the boundary conditions for qualitative changes of interval temporal categories as well as dynamic transmutation of Fuzzy RATA operators.

D. Fuzzy Allen Temporal Relation Within the Framework of the Alternative Set Theory

It is not obvious when the 0 value of a Fuzzy Allen Interval changes into a non-zero value and when a non-zero value becomes 0 again. This is also remains so even in the trapezoidal approximations. Hence we do not deal with sets (be these crisp or fuzzy) but with *semisets*. This necessitates introducing the apparatus of Alternative Set Theory [18],[19] that was developed by Petr Vopěnka, the

founder of internationally acclaimed Prague School of Logic and Model Theory.

D.1 σ - and π -Classes of Fuzzy Allen Intervals

We have seen that the approximation of any Fuzzy Allen Temporal Interval FATI by a trapezoidal fuzzy membership function yields the formula

$$\mu_{FATI}(x) = \begin{cases} (x-a)/(b-a) & \text{when } a < x \leq b \\ (d-x)/(d-c) & \text{when } c \leq x < d \end{cases}$$

Its complement $\text{COMPL}\{\mu_{FATI}(x)\}$ is given by

$$\text{COMPL}\{\mu_{FATI}(x)\} = \begin{cases} 0 & \text{when } -\infty < x < a \\ 0 & \text{when } d < x < \infty \end{cases}$$

In general, we have an indiscernibility surrounding the points a and d of both FATI and of its approximation $\mu_{Fati}(x)$.

D.2 σ - and π -Classes of Fuzzy Allen Intervals

The following properties of special semisets called σ - and π -classes (Vopěnka ([18]) clearly show that the RATA approximations described in the previous section involve these special semisets.

X is a σ -class iff there exists a countable class \mathcal{M} of set-definable classes such that $X = \cup \mathcal{M}$. *Class X is a σ -class if and only if $\mathbf{V} - X$ is a π -class. Let X be a set-definable class. Then X is simultaneously a σ -class and a π -class.*

Lemma: Vopěnka ([18]):

(a) Let X be a σ -class. Then there exists a sequence $\{X_n \mid n \in \mathbf{FN}\}$ of set-definable classes such that for every n it holds that $X_n \subseteq X_{n+1}$ and $X = \cup\{X_n \mid n \in \mathbf{FN}\}$ is a σ -class.

(b) Let X be π -class. Then there exists a sequence $\{X_n \mid n \in \mathbf{FN}\}$ of set-definable classes such that for every n it holds that $X_{n+1} \subseteq X_n$ and $X = \cup\{X_n \mid n \in \mathbf{FN}\}$ is a π -class.

\mathbf{FN} are finite natural numbers. It holds that \mathbf{FN} is a semiset contained in the set of all natural numbers [18].

VI. A SYSTEM FOR BEHAVIOR-BASED INTRUSION DETECTION

The methods and concepts described in the previous sections are utilized in building system for behavior-based intrusion detection.

Behavior-based intrusion detection systems have three core functions:

1. gather ongoing activity in the target environment,
2. construct, store, and maintain profiles that accurately represent user behavior,
3. compare ongoing activity to profiles to detect attacks.

In this section we describe the generic relational architecture that exploits fuzzy relational computations in a substantial way.

A. The Architecture

1. The Event Generator (EG) collects and filters security protocol events for delivery to the Data Preprocessor. Events reflect the activity of sending and receiving messages in protocols. The Event Generator is also referred to as the Protocol Activity Monitor (PAM). These two terms are interchangeable and reflect two possible modes of operation, one being a test mode where the EG is actually producing simulated network traffic for testing purposes while in the PAM mode a live network is the input.
2. The Data Preprocessor (DP) has a primary function of routing global information about the network to the GFRBES while information about individual protocols or profiles are sent to the appropriate NFN for further analysis. Secondary functions include the proper formatting of the data for both Fuzzy Classifier (FC) and GFRBES.
3. The Global Fuzzy Rule-Based Expert System (GFRBES) is where all the global data and information about the network activity is analyzed. The use of a fuzzy rule-based expert system allows for the most flexible reasoning about the widest variety of information possible. Comprehensive global network status reports are the output of this sub-system.
4. The Fuzzy Classifier is employed a number of different situations that may appear in the preprocessed information. The goal is to determine if a session is valid or not in terms of the constraints that the classifier was trained to recognize. Again, comprehensive status reports are generated as the output of this sub-system.
5. The Meta System Reasoner (MSR) may be considered the report generator mechanism of the system. It is here where all reasoned output from the other system components are gathered, synthesized, and their semantic meaning is interpreted and reported.

The behavioral sequences are categorized by the event generator (EG/PAM) as belonging to the following categories: (i) Known protocols. (ii) Known attacks. (iii) Sequences that have not been identified as belonging to either of the two above listed categories. Hence, (iii) may contain the mixture of sequences belonging to the following categories: (a) acceptable protocols; (b) benign behavioral sequences (irrelevant to the problem) and; (c) unknown attacks.

VII. CONCLUSION

In this paper, we accept the event generation architecture of [20] and focus our efforts on the latter two functions: building profiles and recognizing anomalies. This approach is applicable in Highly Distributed Agent Systems. Distributed Artificial Intelligence (DAI) and Multi-Agent Systems (MAS) have become important as new tools for management of multi-disciplinary information. The major goal for DAI and MAS is to understand the organization and co-operation of groups or "societies" of agents. Since the behavior of distributed systems of agents is very complex, dealing with malicious actions in such systems requires the use of new concepts, theories and practical methods in order to understand, design, implement and

verify and protect such systems.

REFERENCES

- [1] J.F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
- [2] W. Bandler and L.J. Kohout. Applications of fuzzy logics to computer protection structures. In *Proc. Ninth International Symposium on Multiple-Valued Logic, Bath, England, 29-31 May, 1979*, pages 200–207. IEEE 79CH1408-4C, New York, 1979.
- [3] D.E. Denning. An intrusion-detection model. In *Proc. of 1986 IEEE Computer Society Symp. on Research in Security and Privacy*, pages 118–131, 1986.
- [4] G.S. Graham and Denning P.J. Protection-principles and practice. *Proceedings of Spring Joint Computer Conference*, 40:417–429, 1972.
- [5] Benjoe Juliano and Wyllis Bandler. *Tracing Chains-of-Thought: Fuzzy Methods in Cognitive Diagnosis*. Physica-Verlag Heidelberg, 1996.
- [6] B. Schneier Kelsey and D. Wagner. Protocol interactions and chosen protocol attack. In *Security Protocols, Proc. of 5th Intl. Workshop*, pages 91–104, Berlin, April 1997 1998. Springer-Verlag.
- [7] L.J. Kohout. Analysis of computer protection structures by means of multiple-valued logics. In *Proc. of the 8th Internat. Symposium on Multiple-Valued Logic*, pages 260–268, New York, 1978. IEEE.
- [8] L.J. Kohout. *A Perspective on Intelligent Systems: A Framework for Analysis and Design*. Chapman and Hall & Van Nostrand, London & New York, 1990. A Scientific Monograph, 255 pages. In 1991, received an international prize from The International Institute for Advanced Studies in Systems Research: "The best book of the year in the area of AI Systems".
- [9] L.J. Kohout, J. Anderson, and W. et al. Bandler. *Knowledge-Based Systems for Multiple Environments*. Ashgate Publ. (Gower), Aldershot, U.K., 1992. A Scientific Monograph, 382 pages. Written by the principal author in collaboration with others. Awarded "Outstanding Scholarly Contribution Award" by the Systems Research Foundation in 1993.
- [10] L.J. Kohout and W. Bandler. Analysis of capability-based computer protection models by means of fuzzy logics. In *Proc. of Eleventh Internat. Symposium on Multiple-Valued Logic*, pages 95–99, New York, May 1981. IEEE.
- [11] L.J. Kohout and W. Bandler. Computer Security Systems: Fuzzy Logics. In M.G. Singh, editor, *Systems and Control Encyclopedia*. Pergamon Press, Oxford, 1987.
- [12] L.J. Kohout and B.R. Gaines. The logic of protection. In *Lecture Notes in Computer Science vol. 34*, pages 736–751. Springer Verlag, Berlin – New York, 1975.
- [13] L.J. Kohout and B.R. Gaines. Protection as a general systems problem. *Internat. Journal of General Systems*, 3:1–21, 1976.
- [14] L.J. Kohout and P. Santiprabhob. Fuzzy dynamic protection structures for trustworthy distributed computations. In *Proc. of the 1st Internat. Conference on Intelligent Technologies, In-Tech2000*, Bangkok, December 2000. Assumption University. 10 pages.
- [15] E. McDuffie and L.J. Kohout. Embedding Alen temporal algebra into fuzzy recursive structures. In *2001 IEEE Internat. Conference on Systems, Man & Cybernetics: SMC 2001 Conf. Proc.*, pages 3420–2424. Systems, Man and Cybernetics Society of the IEEE, IEEE, October 2001. CD-ROM Proceedings, IEEE catalog number 01CH37236C.
- [16] E.L. McDuffie, M. Schneider, F.B. Buoni, E. Schnaider, and L. A. Martin-Vega. Scheduling and temporal dependent interval calculus. In *Intelligent Scheduling of Robots and Flexible Manufacturing Systems*, pages 183–190, 1996.
- [17] P. Santiprabhob and L.J. Kohout. Connected protection structures: A key to secure distributed knowledge-based systems. In M. Fishman, editor, *Proc. of FLAIRS-92 Symposium on Knowledge-Based Systems*, pages 134–138, 1992.
- [18] P. Vopěnka. *Introductino into Mathematics in the Alternative Set Theory, (in Slovak)*. Alfa, Bratislave, 1990.
- [19] P. Vopěnka. The philosophical foundations of alternative set theory. *Int. Journal of General Systems, special issue of Fuzzy Sets & Systems in Czechoslovakia*, 20(1):115–126, 1991.
- [20] A. Yasinac. An environment for security protocol intrusion detection. *J. of Computer Security*, 2002, in press.