Which Models Should Be Applied To Measure Computer Security and Information Assurance?

Leon Reznik
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623-5608

Abstract-**The paper attempts to investigate a feasibility of developing new models and methodologies integrating probabilistic and soft computing techniques and applying them to measurement of computer system security. It reviews the methods, which have been developed and applied by now for this purpose, and analyses their applicability. It concludes that neither of methodologies being applied for measurement of computer security and/or reliability may be considered as comprehensive and good. Summarizing the required features of the measurement models, the paper concludes that new synergetic models and approaches need to be developed.**

I. Introduction

Information assurance and computer security has become one of the most important aspects of information technology and the hottest research field. The goal of this research and development is to improve system security. However, we still do not know how to measure and /or evaluate security and its attributes. Designers often apply information assurance or security technology without the ability to evaluate the impact of those mechanisms to the overall system. And as usual our inability to measure definitely acts as a main constraint to our ability to improve.

Computer and network system security is recognized as a complex issue with no clear definition, with the research domain having no sharp boundaries. Over recent years and especially months, one could see substantial changes in the security problems pattern and their significance on a national and international scenes. The complexity of the problem could be partly explained by the fact that many modern computer systems, whose security should be measured, could be considered as unbounded and emergent, with dynamic changes of their properties and behavior. Examples would be the Internet, any system with distributed administrative control without central authority, any system with remote access, any system with unknown users, and any system containing commercial-off-the-shelf software. The definition of unbounded system is given both formally and informally in [1]. For examples illustrating the role of emergent behavior in the composition process, see [2].

Fisher and Lipson [1] have confidence and some limited evidence that effective solutions to security problems in unbounded networks can arise from revised assumptions coupled with advances in diversity, robustness, adaptability, and algorithmic solutions (which they call emergent algorithms) that generate predictable nonfunctional global properties from simple local interactions.

Any comprehensive security metric clearly should compose measurements of many properties of various natures, with the possibility for the measurement results to be expressed in different scales (numerical, linguistic, comparative). The number of those properties related to security are listed and commented in section II. The methods of verifying the system security either by theoretical methods based on the model produced or by practical tests and experiments are described in section III. Next section IV attempts to evaluate these methods for their potential employment in a global security measurement and a short section V lists some results of soft computing application in related issues, mainly security monitoring and intrusion detection. Based on this analysis section VI formulates the features of the models, which would be feasible to have in the security measurement, and the last section VII concludes on the theoretical models to be developed.

II. What to measure to evaluate security?

The rise of cybersecurity research, a huge increase of the security products available on the market and the potential treat of cyberterrorist's attacks have significantly enhanced the necessity of development of the proper metrics to measure the security. Among those metrics should be ones able to measure [3]:

- system resistance or a capability to repel against a range of attacks and types of aggression,
- recognition ability to detect attacks as they occur and to evaluate the extent of damage and compromise,
- recovery, which can be understood as either restoration of abilities or graceful degradation or in other words the capability to maintain essential services and assets during attacks, limit the extent of damage, and restore full services following attack.

Among the attributes to be measured to assess computer security are availability of the resources and information, reliability, safety, confidentiality, integrity, and maintainability. Koopman [4] more recently stated that a wide variety of metrics and measurement methodologies have been proposed for the software reliability area, which can be applied for computer security assessment. These metrics tend to use varying combinations of software complexity, development phase defect discovery, testing phase defect discovery, and in-service defect discovery to make predictions about residual defect rates (a broad discussion of such approaches can be found in [5]).

III. WHICH METHODS ARE CURRENTLY EMPLOYED?

Formal verification methods. The computer security community has developed various models of secure systems. The emphasis of such models is usually on confidentiality, that is, on preventing the unauthorized disclosure of information. Early models were based on access control, privacy is enforced by restricting the operations that the active entity in a system (the subjects) are allowed to perform on the data repositories of the system (the objects) In some research those models are called Lampson-style after paper [6]. Such models have well-known limitations [7,8]: they do not separate security policy and enforcement mechanisms, they require knowledge of the internals of a system to identify objects and subjects, and, more importantly, they do not consider covert channels.

One of the most influential documents of the time, produced by NCSC (National Computer Security Center) is the US Computer System evaluation criteria, better known as "The Orange Book"[9]. Amoroso [10] summarizes its goals as follows:

1. To provide a standard metric for the NCSC to improve the security of different computer systems,
2. To guide computer system vendors in the design and development of secure systems
3. To provide means for specifying requirements in Government contracts.

In this context, the major results of the early 80s by the formal methods community were in development of theorem proving tools. The general-purpose Boyer-Moore theorem prover [11], also developed during the same time period, was representative of the state of the art in automated theorem proving tools and was applied to many examples.

More abstract models that address these issues have been proposed. They are all related to the concept of non-interference put forward by Goguen and Meseguer [12]. In such models, security is defined as the absence of unauthorized information flows between users of a system. The security requirements are constraints on the set of sequences of events that can be produced on the input and output interfaces of the system. A survey of these various information flow models and of other security models is given by McLean [8]. The ingredients are always the same: a model of computer systems, a definition of user's actions and observations, and a security property that attempts to characterize the absence of information flow between users. The key modeling choices include whether deterministic or non-deterministic systems are considered and, in the latter case, whether or not probabilistic models are used. Security models also differ in other respects, such as in the representation of time and of input and output events. Security could be examined in connection to three properties: noninterference [12, 13], nondeducibility [14], and causality [15,16].

The system model, which is applied to estimate those properties was proposed by Wittbold and Johnson and by Gray [17,18] and is described in [19]. Noninterference was originally proposed by Goguen and Meseguer [12] in the context of deterministic systems. McCullough [13] defined generalized noninterference as an extension of noninterference to nondeterministic systems. Stavridou and Duterte [19] definition is slightly weaker than generalized noninterference and is essentially the same as proposed by McLean [20]. In noninterference models, information is considered to flow from *A* to *B* if changing the sequence of input values on *A*, while leaving all other input values unchanged can modify the sequence of output values observed on *B*. Nondeducibility is an alternative security property proposed by Sutherland [14]. It is based on the observation that there is a flow of information from *A* to *B* if, by observing *B,* one can deduce something about what happened on *A*. Observing that an output sequence occurred on *B* tells us that the corresponding global trace is an element that satisfies certain requirements [19]. Nondeducibility forbids deductions of the above form. It requires that every sequence observable on *B* is compatible with every sequence Q observable on *A*.

Causality: Roscoe [16] defines two security properties, which are closely related to the notion of causality proposed earlier by Bieber and Cuppens [15]. In both approaches, a system is secure if it appears deterministic to its low level users. Simpson et al. [21] have recently adapted Roscoe's definition of noninterference to safety and fault tolerance. The system appears deterministic to a user who can observe only the ports of *A* and *B*. Input values received on ports not in *A* have then no influence on what such a user can observe, and there is no information flow from such ports to *B*.

The key question about any formal definition of security is how much security it provides in practice. For deterministic systems, noninterference is a very good criterion. Millen [22] showed that if there is no interference from *A* to B in a deterministic system, then the capacity of the communication channel of input *A* and output *E* is zero. The only problem is that noninterference is too strong in some cases. In particular, systems that rely on encryption mechanisms to ensure confidentiality can violate noninterference [20]. On the other hand, nondeducibility is a fairly weak property, as shown by McCullough [13] and by Wittbold and Johnson [17]. Systems that are apparently nonsecure can still satisfy nondeducibility. It may still be useful in certain cases where noninterference and causality do not apply. Since nondeducibilily applies to arbitrary sets of ports, it can be used to specify that output sequences on separate output ports are independent. This may be useful if one wants to prevent deductions about secret outputs from the observation of nonsecret outputs. Such a constraint cannot be expressed by using noninterference or causality.

For nondeterministic systems, both noninterference and nondeducibility suffer from the same limitation: they ignore probabilistic inference. As discussed by Wittbold and Johnson [17], there are visibly insecure nondeterministic systems that still satisfy noninterference and nondeducibility requirements. Better models have been developed that take

probabilities into account [23,24,18]. The probabilistic noninterference model defined by Gray [18] achieves the same result as noninterference in the deterministic case. Probabilistic noninterference ensures that the communication channel from a high-level user to a low-level user has capacity zero, that is, information cannot flow from high to low [18]. Causality applies to probabilistic systems as well as nonprobabilistic models. It is in fact a very strong requirement: if causality holds for the low-level user of a system then the system satisfies probabilistic noninterference. Unfortunately, causality may be too strong and is not an adequate property when encryption is used.

Formal methods generally tend to verify or prove that the system is secure without estimating how secure it might be.

Intrusion injection and detection. In all fault-tolerant systems, the complexity of the coordination protocols and the number of replicas required depend on the number and severity of faults to be tolerated [25, 26]. The simplest case corresponds to crash failures, where a failed server simply stops to produce answers. The most complex case corresponds to Byzantine failures, where no assumption is made about faulty servers, which can exhibit arbitrary behaviors. Other classes of faults include various kinds of omission and timing failures at the node or network level. Tolerance to Byzantine failures requires active replication and can be achieved only under strong assumptions about the communication network [27]. Less expensive approaches such as transaction-based systems or primary-backup architectures are of greater practical interest in common distributed systems. The basic assumption behind all server-replication techniques is that the same fault does not affect all replicas. Server nodes are assumed to fail independently. Fault isolation is also required, the failure of a server or client site must not cause the failure of other sites.

Intrusion tolerance may have several objectives, depending on the system's dependability requirements. In many systems, the objectives might be to preserve information confidentiality and integrity after an attacker has penetrated the system. In other applications, ensuring the continuity and quality of service might be more important than confidentiality issues. Simple replication schemes are of little use for ensuring confidentiality and data integrity after intrusion. Existing intrusion-tolerant systems that ensure confidentiality and integrity rely on the distribution of trust and data and use threshold encryption schemes [28,29]. The authentication mechanisms used in such systems can also provide some protection against denial of service.

Ensuring fault isolation is also an important related issue. Initial intrusion at a node should not enable easy access to the rest of the system. Although illicit access to a server can have a more immediate effect on the service, intrusion tolerance must also take into account possible attacks mounted from a client site. The server replication techniques described above are intended to protect against server failure but for such schemes to work in practice, the servers should be designed so that failures of a client do not propagate to servers. Protecting servers against rogue clients is even more a concern in the intrusion tolerance context.

The approaches used to protect against accidental faults in distributed systems can provide solutions to intrusion tolerance in case the objective is to maintain service after intrusions. For such solutions to be effective, one must ensure that the underlying assumptions about fault independence and isolation are still valid in the case of intentional faults. These assumptions are more difficult to justify in the case of deliberate attacks than in the case of accidental faults. Proper security measures, such as strong user authentication or separation of duties must be in place.

In addition, it is not clear whether the classes of faults considered in usual replication schemes, such as Byzantine failures crashes, or omission failures, are adequate for dealing with intrusions. In the worst case, an intruder can gain full control of a node. The behavior of a compromised node is then arbitrary and intrusions lead to Byzantine failures. Unfortunately, protection against such failures requires costly solutions and is difficult to implement in large distributed systems, when not theoretically impossible [30]. On the other hand, an assumption that penetrated nodes do not generate any new malicious activity or only exhibit omission or timing failures is too optimistic. A better understanding and classification of the behavior of nodes after an intrusion is still needed.

Tests: Most of the testing metrics described in the literature are designed at the unit or source code level. There are just few objective measures of coverage that are independent of the implementation. Traditional program mutation analysis [31] is a code-based method for developing a test set that is sensitive to any small syntactic change to the structure of a program. A mutant program is produced by applying a single mutation operator exactly once to the original program. The rationale is that if a test set can distinguish the original program from a mutant, the test set exercises that part of the program adequately. Applying the set of operators systematically generates a set of mutants. Some of these mutants may still be equivalent to the original program. A test set is mutation adequate if at least one test in the test set distinguishes each nonequivalent mutant. There are test data generation systems that, except for the ever-present undecidability problem, attempt to automatically generate mutation adequate test inputs [32]. Very little work on mutation analysis for specifications is reported in the literature; however, Woodward did apply mutation analysis to algebraic specifications [33] and so did Ammann and Black [34], who categorize mutations of temporal logic formulae with respect to specification coverage analysis and define expounding, in which implicit aspects of a model checking specification are made explicit, and then describe how to symbolically evaluate a test set for mutation adequacy.

IV. HOW GOOD ARE THOSE METHODS?

Research in computer security has led to the development of a rich theory of secure systems. The search for a mathematically precise yet practical definition of security based on information flow was a central motivation for most of the formal methods work. The results have produced a collection of possible characterizations of security for several classes of systems. The situation is reasonably clear for deterministic systems where noninterference provides an almost perfect security criterion, though sometimes too strong. On the other hand, a practical notion of information flow security for nondeterministic systems has proved much harder to obtain. Probabilistic models provide convincing security definitions but are fairly complex, and few examples of successful applications are available. Nonprobabilistic models are simpler and seem easier to apply, but they can be both too weak to ensure real security and too strong in many applications. As a whole, the various security models remain mostly of theoretical interest, and practical applications are scarce. These security models have had little impact on the way computer systems are built in practice.

Some of the difficulties in applying information flow models in practice are due to their "all or nothing" philosophy. In other words, information flow models aim at defining perfect and absolute security that practically cannot be achieved. Most models are not adequate for more realistic security goals, such as tolerating covert channels of low capacity. Only the more complex probabilistic models are able to characterize such properties. Another limitation of many existing information flow models is that they promote a centralized view of computer systems. The associated analysis method requires first the construction of a model of the global system's behavior and proof that this global model satisfies the expected properties. Because of noncompositionality, no clear alternative to this naive approach has emerged. Security models make worst-case assumptions about system users and environment. Systems have to resist deliberate attack by a skilled opponent. The attacker is often assumed able to exploit covert channels and to insert Trojan horses in the system. One must then make sure that no information flow exists between two users who are colluding and trying their best to communicate. The properties required to ensure information confidentiality under such pessimistic assumptions are very strong and hard to implement in practice. In the safety domain, the objective is usually to protect against accidental rather than deliberate faults, and one can then make more favorable assumptions. For example, faulty components may be fail-silent or fail-stop or may be assumed to fail independently. In non-security related applications, it becomes easier to define a tolerable degree of interference. Essentially, interference can be tolerated as long as the top-level property of interest still holds. For example, the authors [35] consider bounded interference in the temporal domain. There is no need for sophisticated probabilistic models and information-theoretic notions for obtaining noninterference notions that can be achieved in practice. This contrasts with the unrealistic "all or nothing" properties proposed in security models.

Products may be delivered with default settings that intruders can exploit. System administrators and users may neglect to change the default settings, or they may simply set up their system to operate in a way that leaves the network vulnerable. An example of a faulty configuration that has been exploited is anonymous File Transfer Protocol (FTP) service. Secure configuration guidelines for this service stress the need to ensure that the password file, archive tree, and ancillary software are separate from the rest of the operating system, and that the operating system cannot be reached from this staging area. When sites misconfigure their anonymous FTP archives, unauthorized users can get authentication information and use it to compromise the system.

One can conclude that it is not enough to look at just the system or even the system and its intended operating environment. Formal methods need to be integrated with other methods that can address issues, some of which are beyond the scope of formalization raised by examples like the one above. These analyses include risk analysis, hazard analysis, fault analysis, and intrusion detection analysis. Formal methods also need to be better integrated into the entire software development lifecycle such as during requirements analysis, testing, and simulation.

There have been developed numerous tests in order to decide if the system is subject to attack, been compromised, intruded, information been stolen, etc. A few tests can produce numerical results, even less are able to produce some characteristics directly characterizing the “degree” of security, at least in some aspects. However, everyone recognizes that there exist some relationship between the test results and security characteristics of the system. Other difficulties in evaluating security of unbounded systems include:

- the need for thousands to tens of thousands of nodes to be analyzed,
- lack of linguistic mechanisms in conventional methods and programming languages for making incomplete and imprecise specification,
- the inability of object oriented computations to describe and reason about the real world,
- the need to combine information about a system from multiple knowledge domains,
- management of multiple simultaneous beliefs of the various stakeholders in an infrastructure,
- integration among separately developed simulations, and

exponential increases in computational cost that accompany linear increments in the granularity or number of nodes in a simulation.

Finally, the human factor, which in principle is part of the system's environment, must be introduced. Research in modeling human behavior, human-computer interaction, and management of processes and organizations can all complement the more formal nature of security measurement methods.

V. HOW IS SOFT COMPUTING APPLIED NOW?

Soft computing community has already accumulated a number of achievements of applying fuzzy, neural networks and evolutionary methodologies in computer security issues, mainly monitoring with intrusion detection systems (IDS) and algorithms. In IDS design the successful attempts have been made to move from a pure statistical approach to detect anomaly [36] to the application of fuzzy methodologies in algorithm modification [37], to the development of fuzzy decision systems [38-40] and to design more sophisticated intrusion methods integrating fuzzy classifiers with genetic algorithms [41, 42] and neural networks [43]. All these works have laid a wonderful foundation for an expansion of monitoring systems to evaluation systems, which will require development of the more comprehensive decision systems. Those systems might combine behavior monitoring and prediction, practical test results with an application of formal methods enriched by incorporating soft computing techniques.

VI. WHAT FEATURES OF MODELS ARE REQUERED?

There are no security metrics, which produce accurate measurements and predictions of the behavior of unbounded systems. By definition, unbounded systems are incompletely and imprecisely defined. Most current models, however, require complete information and thus are always built with assumptions of inaccurate information. The ability to operate on abstract specifications and simulate at various levels of abstraction is a long-standing need of many applications, but is not provided as a feature of existing systems.

Equally important, all object based models (both physical and computerized) are inherently inaccurate because they are based on complete representations as objects. This might be acceptable when dealing with small numbers of nodes or when great care is taken to differentiate between which modeling results are likely to be valid. Such remedies seldom if ever succeed in differentiating inaccurate results when modeling complex or large scale system. Furthermore, as the number of subsystems in a model increases, the inaccuracies of each subsystem become dominant after a few iterations and guarantee that all results will be inaccurate. This may account for the pervasive failure of large scale simulations to produce accurate results. These problems are aggravated in unbounded systems where the numbers of components are very large and a primary purpose of simulation is to accurately predict the global effects from local activities.

Because accuracy and completeness are not simultaneously achievable when describing the physical world, an accurate simulation is feasible only if the simulator can guarantee accurate results from accurate but incomplete specifications. Koopman [4] states desirability to have a measurement tool or methodology that does not require much detailed information about the system whose security is measured, does not attempt to measure quantities that deviate only very slightly from perfection, uses novel tests or differs measurement from defect correction and is seen as a constructive activity. The assessment concern is that there is often significant disparate evidence that augurs well (or ill) for a given system, but currently there is no methodology, which allows advancing from a project engineer's expert opinion to a repeatable scientific exercise [10].

VII. CONCLUSION: WHICH MODELS SHOULD BE USED?

Alternative models applied for uncertainty evaluation should satisfy the following conditions:

- they must possess sufficient features to describe the uncertainty components (based on expert's opinion or judgments, model-dependent variability sources and components resulting from intelligent techniques application),
- they should be supported by a well developed theory providing the methods for model developing and processing,
- they should allow an easy implementation in modern measuring systems,
- they should allow for a joint application and processing along with the statistical models,
- in practice, they should not lead to any unreasonable difference in the numerical value of the measurement result or of the uncertainty assigned to that result because of the difference in the point of view or the formalism of the methods.

Thus a universal model based on fuzzy logic with application of probabilistic and statistical methods to deal with uncertainty sources where applicable should be considered feasible and further investigated as the underlying hierarchical philosophy and methodology to deal with the problem of computer security measurement. One of the approaches for producing measurement results based on fusion of information coming from a variety of sources, including direct measurements, statistics and expert's estimates, and having different nature is presented in [44-48].

REFERENCES

[1] Fisher D.A. and H.F. Lipson, "Emergent Algorithms - A New Method for Enhancing Survivability in Unbounded Systems," Proceedings of the 32nd Annual HawaiiInternational Conference on System Sciences, Maui, Hawaii, January 5-8, 1999 (HICSS-32), IEEE Computer Society, 1999

[2] Hinton H.M., "Under-Specification, Composition and Emergent Properties," Proceedings of the 1997 New Security Paradigms Workshop, Langdale, Cumbria UK, September 23-26, 1997, ACM, 1998

[3] Ellison R.J., D.A. Fisher, R.C. Linger, H.F. Lipson, T.A. Longstaff, N.R. Mead, "Survivable Systems: An Emerging Discipline," Proceedings of the 11 th Canadian Information Technology Security Symposium (CITSS), Ottawa, Ontario Canada, May 10-14, 1999, Communications Security Establishment, 1999

[4] Koopman, P. "Toward a scalable method for quantifying aspects of fault tolerance, software assurance, and computer security", Proceedings Computer Security, Dependability, and Assurance: From Needs to Solutions , p. 103-31, 1999, Los Alamitos, CA, USA

[5] Hartz M, Walker E. and Mahar D. "Introduction to software reliability: A state of the art review", Reliability Analysis Center Report SWREL, Rome NY, December 1996

[6] Lampson B. Protection. In Proceedings of the Fifth Princeton Symposium on Information Science and Systems, 1971. Reprinted in ACMU Operating Systems Review, Vol. 8, 1974

[7] McLean J. A Comment on the 'Basic Security Theorem' of Bell and LaPadula, Information Processing Letters. v.20(2):p. 67-70, February 1985

[8] McLean J. Proving Noninterference and Functional Correctness Using Traces, Journal of Computer Security, 1(1), p.37-57,1992

[9] National Computer Security Center, Department of Defense, Trusted Computer Security Evaluation Criteria. Technical Report DoD 5200.28.STD, NCSC, 1985

[10] Amoroso E.. Fundamentals of Computer Security Techology. AT&T Bell Laboratories, 1994

[11] Boyer R. and J. Moore. A Computational Logic, ACM monograph series, Academic Press, New York, 1979

[12] Goguen J.A. and J. Meseguer Security Policies and Security Models. Proceedings of the IEEE Symposium on Reasoning in Security and Privacy, p. 11-20, Oakland, CA, 1982

[13] McCullough D. Specifications for Multi-Level Security and a Hook-Up Property, Proceedings of the IEEE Symposium on Research in Security and Privacy, p. 161-166, Oakland, CA, 1987

[14] Sutherland D. A Model of Information, Proceedings of the 9th National Computer Security Conference, p. 175-183, Gaithersburg, MD, September 1986

[15] Bieber P. and F. Cuppens. A Logical View of Secure Dependencies, Journal of Computer Security, v.1(1):p.99-129, 1992

[16] Roscoe P. CSP and Determinism in Securily Modelling, Proceedings of the IEEE Symposium on Research in Security and Privacy, p. 114-127, Oakland, CA, May 1995

[17] Wittbold J.T. and D.M. Johnson. Information Flow in Nondeterministic Systems, Proceedings of the IEEE Symposium on Research in Security and Privacy, p. 144-161, Oakland, CA, May 1990

[18] Gray J.W. Toward A Mathematical Foundation for Information Flow Security, Journal of Computer Security, v.3-4(1), p.255-294, 1992

[19] Stavridou V. and B. Dutertre "From security to safety and back", In: Proceedings on Computer Security, Dependability and Assurance: From Needs to Solutions, 1998, p. 182 - 195 7-9 July 1998 & 11-13 November 1998 York, UK & Williamsburg, VA, USA, 1999

[20] McLean J. Security Models, Encyclopedia of Software Engineering, J. Marciniak, editor. Wiley and Sons. 1994

[21] Simpson A., J. Woodcock, and J. Davies. Safety through Security, Proceedings of the Ninth International Workshop on Software Specifications and Design, p. 18-24, Ise-Shima, Japan, April 1998

[22] Millen J.K. Covert Channel Capacity. Proceedings of the IEEE Symposium on Research in Security and Privacy, p. 60-66, Oakland, CA, April1987

[23] McLean J. Security Models and Information Flow, Proceedings of the IEEE Symposium on Research in Security and Privacy, p. 180-187, Oakland, CA, May 1990

[24] Gray J.W. Probabilistic Interference, Proceedings of the IEEE Symposium on Research in Security and Privacy, p. 170-179. Oakland. CA. May 1990

[25] Lynch N. Distributed Algorithms, Morgan Kaufmann, 1996

[26] Budhiraja N., K. Mazrullo, F. Schneider and S. Toueg. The Primary-Backup Approach. In: Distribuud Systems, Mullender/ Editor, p. 199-216, Addison-Wesley, 1993

[27] Lamport L., R. Shostak. and M. Pease. The Byzantine General Problem. Journal of the ACM, v.4(3):p. 382-401. July 1982

[28] Deswarte Y, L. Blain, and J.-C. Fabre. Intrusion Tolerance in Distributed Computing Systems, Proceedings of 'he IEEE Symposium on Research in Security and Privacy, p. 110-121, Oakland, CA. May 1991

[29] Fabre J.-C., V. Deswane, and B. Randell. Designing Secure and Reliable Applications using Fragmentation- Redundancy-Scattering: an Object-Oriented Approach, First European Dependable Computing Conference (EDCC-1), p. 21-38, Berlin, Germany, Oclober 1994. Springer-Verlag

[30] Fischer M., N. Lynch, and M Merrin Easy Impossibility Proofs for Distributed Consensus, Distributed Computing, *v.*1(1), p. 26-39, January 1986

[31] De Millo R.A., R. J. Lipton, and F. G. Sayward. Hints on test data selection: Help for the practicing programmer, IEEE Computer, v.11(4): p. 34-41, April 1978

[32] De Millo R.A. and A. J. Offutt. Constraint-based automatic test data generation, IEEE Transactions on Software Engineering, v.17(9), p.900-910, September 1991

[33] Woodward M.R. Errors in algebraic specifications and an experimental mutation testing tool, Software Engineering Journal, p. 211-224, July 1993

[34] Ammann P.E., Black P.E. "A specification-based coverage metric to evaluate test sets", In: High-Assurance Systems Engineering, 1999, Proceedings of the 4th IEEE International Symposium on 17-19 Nov. 1999, Washington, DC, USA p: 239 – 248

[35] Dutertre B. and S. Schneider. Using a PVS embedding of CSP to verify authenticalion protocols. In Theorem Proving in Higher Order Logics, p. 121-136, August 1997, LNCS 1275

[36] Denning D.E. An intrusion detection model. In: Proc. of the 1986 IEEE Computer Society Symposium on Research in Security and Privacy, p.118-131, 1986

[37] Florez G., Bridges S.M., Vaughn R.B. "An improved algorithm for fuzzy data mining for intrusion detection" In: Fuzzy Information Processing Society, 2002 Proceedings NAFIPS., p. 457 – 462, 27-29 June 2002, New Orleans, LA, USA, 2002

[38] Kohout L.J., A.Yasinsac, E.McDuffie "Activity profile for intrusion detection" In: Fuzzy Information Processing Society, 2002 Proceedings NAFIPS., p. 463-468, 27-29 June 2002, New Orleans, LA, USA, 2002

[39] Dasgupta D. and Gonzalez F.A. "An intelligent decision support system for intrusion detection and response" In: Proceedings of the International workshop on Mathematical Methods, Models and Architectures for Computer Networks Security, May 21-23, 2001, St.Petersburg, Russia

[40] Dickerson J.E., Juslin J., Koukousoula O., Dickerson J.A. "Fuzzy intrusion detection" In: Joint 9th IFSA World Congress and 20th NAFIPS International Conference, 2001, vol.3, p. 1506 – 1510, 25-28 July 2001 Vancouver, BC, Canada 2001

[41] Gomez J., Dasgupta D., Nasraoui O., Gonzalez F. "Complete expression trees for evolving fuzzy classifier systems with genetic algorithms and application to network intrusion detection" In 2002 Proceedings NAFIPS, p. 469-474, 27-29 June 2002, New Orleans, LA, USA, 2002

[42] Gomez J., Dasgupta D. "Evolving fuzzy classifiers for intrusion detection" Proceedings of the 2002 IEEE Workshop on Information Assurance, US Military Academy, West Point, NY, June 2001

[43] Liu Z., Florez G., Bridges S.M. "A comparison of input representations in neural networks: a case study in intrusion detection" In: Proceedings of the 2002 International Joint Conference on Neural Networks, vol.2, p. 1708 – 1713, 12-17 May 2002, Honolulu, HI, USA, 2002

[44] Bloch I. "Information combination operators for data fusion: a comparative review with classification", IEEE Transactions on Systems, Man and Cybernetics, part A, vol. 26. No.1, pp.52-67

[45] Reznik L. and G.Solopchenko "Use of a priori information on functional relations between measured quantities for improving accuracy of measurement", Measurement, 1985, vol. 3, No. 3, pp. 98-106

[46] Reznik L., G. Solopchenko "Method of Complexing of Measurements", The USSR Patent: Bulletin of Patents, 1986, No. 47

[47] Lin C.T., Lee Y.C., Pu H.C. "Satellite sensor image classification using cascaded architecture of neural fuzzy network", IEEE Transactions on Geoscience & Remote Sensing. vol. 38, No. 2, part 2, 2000, pp.1033-1043

[48] Reznik L., V. Kreinovich and S.A.Starks "Use of Fuzzy Expert's Information in Measurement and What We Can Gain from Its Application in Geophysics (to appear in these proceedings)