Introduction to Logical Foundations of an Adaptive Security Infrastructure*

Leo Marcus
The Aerospace Corporation
Los Angeles
marcus@aero.org

June 23, 2004

Abstract

We give an introduction to questions relating to the logical underpinnings of an adaptive security infrastructure.

1 Introduction

The goals of this paper are to introduce the Adaptive Security Infrastructure concept, discuss issues of assurance and logical formalization, and state some tentative definitions and theorems.

The term “adaptive security” is intended to indicate that security policies and mechanisms can change in some automated or semi-automated fashion in response to events. Of course, adaptation is a matter of degree; all security architectures and devices are adaptive to *some* degree.

The need (or “use”; of course, as in many such technological “advances”, sometimes it is a case of “invention is the mother of necessity” instead of the other way round) for (more adaptive) adaptive security stems from two considerations: short term and long term:

1. standard “static” security architectures do not cope well with rapidly changing security environments, including physical parameters, threats, attacks, policies, and mission goals.

2. At the other end of the spectrum, systems designed for extended many-decade life cannot predict and handle all future threats and attacks by *ab initio* built-in non-flexible mechanisms.

Appropriate adaptive architectures and mechanisms should be chosen according to which aspects of the short-term or long-term need are being addressed.

The term “infrastructure” was added on to “adaptive security”, obtaining Adaptive Security Infrastructure (ASI), in order to indicate the approach that sees adaptive security as an integral, fundamental, functional component underlying any system, rather than an ill- (or nil-) structured collection of security devices.

While this need is becoming increasingly recognized – one could even say that over the last few years there has been a paradigm shift toward adaptivity – systems are still being specified, designed, and built without a good method for architecting system-wide adaptive security mechanisms.

---

*This paper is intended as an introduction to WOLFASI, the Workshop on Logical Foundations of an Adaptive Security Infrastructure, held in conjunction with FCS’04, LICS’04, and ICALP’04, July 12-13, 2004, in Turku, Finland. As such, it is hereby declared to be exempt from some traditional requirements of professional papers, for example, to present actual results.

Much work is currently being focused on detailed aspects of the related fields of intrusion detection, sensor networks, architectures, and security policies. Much less work is devoted towards putting together those pieces[1]. In particular, there does not appear to be a currently accepted good method for gaining confidence that the mechanisms to be employed will work together to deliver what, and only what, is needed. The hard part is "only" to decide what is wrong (security-wise) with the current state of affairs, what to do about it, and how to do that, with the resources available. Without a system-wide perspective, mechanisms can interfere with each other, be counter-productive, and create new vulnerabilities. Indeed, without the assurance that comes from rigorous specification leading to an enhanced likelihood of real verification, the cure may be worse than the disease.

Perhaps reflecting the author's personal bias, the first step toward true assurance requires some formalization of an ASI that *could, eventually* lead to the verification that proposed adaptive security mechanisms will perform as hoped (specified).

Enough about the need for adaptive security and formalization. In any case, we hope to show that there are some interesting logical questions relating to ASIs that have not really been addressed until now[2]. It is a hope of this workshop to help remedy that.

2 Components of an ASI

In order to be able to satisfy the stated goals, i.e., to coordinate detection of security-relevant input, security policy, user input, analysis, and then be able to formulate and execute a response, if needed, a natural approach is to isolate the three conceptual components of sensor, analysis, and response.

Taking this approach to the extreme, one can imagine a system which is constantly monitoring, analyzing, and responding, in order to maintain security invariants or to evolve the system to satisfy new security properties, taking into account current security policy, severity of environmental effects, temporal and geographic aspects of attacks and responses[3].

The skeptical reader may be wondering how we can hope to prove anything about such a complicated system, when we can barely prove the most rudimentary security properties of the most rudimentary devices and mechanisms[4]? The answer is hierarchy! In other words, *assuming* the building blocks (protocols, algorithms, devices, interfaces) work as advertised, how do they function together? What properties need to be defined in order to even formulate theorems? What properties must components and interfaces have in order that their cooperative effect satisfies some desired property?

3 Formalization: Principles and Issues

What kind of "formalization" are we interested in? Some vague basic principles:

1. Use a mathematical logical framework

2. Abstract from realistic scenarios

[1] Over the past few years I have accumulated approximately 600 research papers on relevant topics. The bibliography section lists some of them.

[2] There are certainly connections with the related fields of "collaborative enterprises", "self-healing systems", "auto-adaptive systems", "reconfigurable systems," and the like.

[3] It is tempting at this point to jump straight to modeling biological defenses, immune systems, etc. But we prefer a more logical framework.

[4] If the skeptical reader would really say that, he or she is obviously not aware of some very good "historical" work in proving security properties of "evaluated products," or more recently security properties of protocols. Nevertheless, the point that we are far away from proving properties of an ASI is correct.

3. Don't be concerned with usability or current technology (of course, at a deeper level, we recognize that current technology has an undeniable, if unmeasurable, influence on our imagination)
4. Long term goal should be a common, uniform, inter-interpretable semantics to allow rigorous specifications and verifications of architectures, properties, and capabilities that can connect policy, detection, analysis, and response.

The basic assumption:

- ASI exists in a temporal and spatial world. If we accept the temporal and distributed nature of the whole system in its full generality, we get arbitrary architectural structures (patterns of connectivity, e.g. generalized networks) existing within the system and the ASI, and these structures may be dynamically changing. Any aspect of policy, specification, detection, analysis, or response can be considered in a version relativized to any definable structure. We call this the Pervasive Hierarchy Assumption (PHA).

The following research issues may appear to be rather grandiose in scope. Of course, they are, but part of the fun is to break them up into smaller bite-size, or at least meal-size, chunks.

1. What are the appropriate semantics of a dynamic, adaptive security policy, and how should that be specified?
2. How should we take into account the global-local nature of all components of an ASI according to the PHA?
3. How should we specify the "security-relevant resources" available so that at any time the analyzer can choose an appropriate response?
4. How do we specify the capabilities of responses (including trade-offs?)
5. How should we unify the temporal-spatial reasoning aspects?
6. What are the decidability or complexity issues in such a system?
7. What is the role of "approximate security"?

3.1 Research Issues: Spatial

Some of the interesting research issues pertaining to the spatial dimension are:

1. Specification of hierarchical architectures
2. Central (local) and distributed (global) detection, analysis, and response coordination
3. Smooth transition between hierarchies
4. Testability of policy satisfaction
5. Enforceability of response

3.2 Research Issues: Temporal

Some research issues pertaining to the temporal dimension are:

1. Duration of response
2. Synchronization
3. Relative speeds of changing environment, detection, analysis, communication, response
4. Incorporation of time in policy
5. Acknowledgments, success reports

4 Adaptive Security Policy

The goal for specifying adaptive security is twofold: to provide an umbrella guide for deciding if future events, actions, or responses are to be permitted under current policy; and to allow new security goals to be stated, in order to initiate system responses to enforce that policy, if necessary.

For example, we want to be able to reason about policy change within the context of larger policy or policy hierarchy[5]. We want to be able to test, prove, and implement security policies. We also want to be able to analyze combinations of security policies, for example, if the union of two security policies contains a contradiction.

We have used the term "security policy" without definition until now, which is dangerous since it might mean a lot of different things to different people, or to the same person at different times (as in the case of the author.) But what we mean here and now can be stated intuitively as follows:

- a security policy is (a specification of) what is allowed.

More precisely, in purely semantic terms, a security policy is a set of computer systems, namely those computer systems that satisfy that policy. Thus, if a computer system is identified with a set of computation sequences (the set of its permitted computation sequences), then a security policy is a family of sets of computation sequences. It is hard to get more general than that[6]. The general definition can be refined a bit by defining a *primitive* security policy to be a set of computations (so , e.g. "non-interference" or "non-deducibility" are not primitive), and an *enforceable* security policy to be a primitive policy that can be monitored.

Exactly which of these security policies are "static" and which are adaptive (or dynamic, if you prefer), is not a question with an objective answer.

However, as an example of a simple adaptive policy consider the following:

- System initially satisfies policy P_1
- At the first occurrence of condition C, system switches to policy P_2.

So this immediately raises the issue: what does satisfying a policy P in an interval (from one time/event t_1 to another time/event t_2) mean?

Answer?: *non-contradicting* the policy, i.e., that there is some continuation of the computation, or in the case of non-primitive policies, some enlargement of the set of computations (within some larger context of admissible computations), that explicitly satisfies the policy.

If we represent the above situation by $\langle P_1; C \rightarrow P_2 \rangle$ then we can easily generalize the notation to, for example:

[5] As a simple example, a structured policy hierarchy can specify the system policy with regard to the security/performance tradeoff. At certain times confidentiality may be more important, and at others, availability.

[6] This generality appears to dilute totally the use of the word "security" in ASI. Security is certainly the motivation, and source of examples, but there may not be a technical (logical) reason to limit these considerations to the conventional security concerns of confidentiality, integrity, availability, etc.

1. $\langle P; C_1 \rightarrow P_1, C_2 \rightarrow P_2, \ldots, C_n \rightarrow P_n \rangle$ Branching Policies

2. $\langle P; C_1 \rightarrow \langle P_1; C_2 \rightarrow P_2 \rangle \rangle$ Compound Policies

with the obvious intended meanings.

4.1 Incremental Policy

An incremental policy change is when we know what aspect we want to change, but don't know or don't care about the rest of the policy as expressed in its complete system-wide specification. For example, changing one user's access rights could/should be expressible as an increment affecting only that user. This raises the question of dependencies among policies that may appear to be local: perhaps the change to one user's access rights, via some admissible interaction with other users, changes those other users' rights as well.

An increment can be a "weakening" (allowing more computations) represented by set union of the previous policy with the new policy, or a "strengthening" (allowing fewer computations) represented by set intersection of the previous policy with the new policy.

A policy increment can be indicated by: $\langle P; C \rightarrow (+P_1 - P_2) \rangle$, where P_1, P_2 are themselves policies, meaning: strengthen by P_1 and then weaken by P_2. Such an increment could be a complex combination of strengthenings and weakenings.

4.2 Local Policy

Let H be a hierarchy description, A an ASI specification (as opposed to an individual instantiation), and P a policy.

Intuitively, we want

- *P is local with respect to H in A*

to mean something like

- the satisfaction of P in A is dependent only on the satisfaction of some (perhaps other, "test") policy in all subsystems satisfying H.

In certain situations we may want to define locality differently, by playing with the quantifiers and saying

1. "*For all* instantiations of A *there is* a test policy for P such that ..." or

2. "*There is* a test policy for P such that *for all* instantiations of A ..." or

3. "... in *some* subsystems satisfying H"

5 Specification, Derivation, and Verification of Response

One of the more challenging questions is how to specify and reason about responses, their relation to resources, and their capabilities. As examples, in current 2004 technology, some kinds of (defensive) responses that would be appropriate for certain security-relevant tasks include, in random order:

1. allocate resources (e.g. power; turning devices on or off)

2. adjust routing (include or exclude nodes)

3. change access rights

4. change crypto algorithms, keys, protocols
5. change sensor networks
6. change auditing
7. change strength of authentication
8. adjust intrusion detection system settings (altering the false positive/negative ratio)
9. install patches
10. destroy data or devices
11. install new hardware or software

In the general formal context of an ASI we can define a "response" to be simply a distributed program/algorithm running concurrently with the ongoing ASI and system operation. Of course, intuitively, common responses have more specific properties, like changing the state and terminating.

In order to incorporate responses into a formal framework, we need to

1. Specify and evaluate responsive resources
    - including communication channels, if needed
    - and including current (and projected) strength and location
2. Coordinate response with analysis
3. Plan appropriate action in time and space; consider temporary and local "fixes" while long-term global solution-response is being worked on

6 Detection and Analysis

The detection and analysis components are very closely related.

Typical detection data and mechanisms currently employed include:

1. intrusion detection methods of various kinds (e.g. signature and anomaly)
2. network statistics
3. system usage statistics
4. insider threat statistics
5. electronic background data

Who knows what other kinds of environmental information may be useful in the future? In coordinating this information, lessons from the field of sensor networks are very relevant here. Obviously, the possible connection between the nature of data collected, the nature of the policy implemented, and the nature of the analysis engine, and how these connections themselves can be made adaptive, is a wide open question.

7 Other Topics

Other issues that could easily be relevant to the formalization of an ASI are

1. Approximate security, that is:
   - How to specify *achievable* security goals
   - Allow statistical properties in security policies
2. Game-theoretic view, that is:
   - Consider adaptive security to be a game between the environment and the ASI
   - The goal is to (assume minimal restriction on the environment and) design the ASI so the adversary (environment) does not have a winning strategy

8 Future Theorem

A typical theorem to be proved in some distant future verification of an ASI could look like:

Theorem:

1. For any system *S* implementing the specification S
2. for any ASI *A* implementing the specification A
3. for any adaptive security policy *P* of type P
4. for any environment *E* satisfying conditions E:

S + *A* satisfies *P* in *E*.

The ASI architect's problem: Given E, P, and S, find A, as above. As E gets more "realistic", P has to get weaker in order for there to be any hope of finding an appropriate A. This weakening can be in the temporal axis (allow for longer "lapse" of security) or the approximation axis (allow for less rigorous security conditions.)

9 Bibliography

Here we present a brief, definitely not comprehensive, list of documents that may provide the interested reader with some good starting points, before beginning his or her own directed internet search.

1. N. Aguirre, T. Maibaum, A temporal logic approach to the specification of reconfigurable component-based systems, Proceedings of the 17th International Conference on Automated Software Engineering ASE 2002, Edinburgh, UK, September 2002

2. M. Aksit, Z. Choukair, Dynamic, adaptive, and reconfigurable systems: overview and prospective vision, DARES - The International Workshop on Distributed Auto-Adaptive and Reconfigurable Systems, in 23rd International Conference on Distributed Computing Systems Workshops (ICDCS 2003 Workshops), 19-22 May 2003, Providence, RI, USA

3. X. Ao, N. Minsky, T. Nguyen, A hierarchical policy specification language, and enforcement mechanism, for governing digital enterprises, Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02)

4. I. Aron, S. Gupta, Analytical comparison of local and end-to-end error recovery in reactive routing protocols for mobile ad hoc networks, Proceedings of the 3rd ACM international Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Boston, Massachusetts, 2000, pp 69 - 76

5. A. Arora, S. Kulkarni, Detectors and correctors: a theory of fault-tolerance components, 18th International Conference on Distributed Computing Systems, pp: 436–443, IEEE Computer Society, Amsterdam, The Netherlands, 1998

6. L. Bauer, J. Ligatti, D. Walker, More enforceable security policies. In Foundations of Computer Security, Copenhagen, Denmark, July 2002

7. L. Bauer, J. Ligatti, D. Walker, A calculus for composing security policies Technical Report TR-655-02, Princeton University, 2002. http://citeseer.ist.psu.edu/bauer02calculus.html

8. M. Becker, L. Gilham, D. Smith, Planware II: Synthesis of schedulers for complex resource systems, Kestrel Institute, submitted for publication, 2003

9. A. Belokosztolszki, Ken Moody and David M. Eyers, A formal model for hierarchical policy contexts, Policy 2004: IEEE 5th International Workshop on Policies for Distributed Systems and Networks, Yorktown Heights, U.S.A. June 2004.

10. R. Bharadwaj, Secure middleware for situation-aware naval C2 and combat systems, 9th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2003), 28-30 May 2003, San Juan, Puerto Rico

11. C. Bidan, V. Issarny, Dealing with multi-policy security in large open distributed systems, Proceedings of 5th European Symposium on Research in Computer Security, pages 51–66, September 1998

12. J. Burns, A. Cheng, P. Gurung, S. Rajagopalan, P. Rao, D. Rosenbluth, A. Surendran, D. Martin, Automatic management of network security policy DARPA Information Survivability Conference and Exposition (DISCEX II). Volume II. Pages 12-26. Anaheim, CA. June 2001. Pub. IEEE Computer Society Press, Los Alamitos, California. ISBN: 0769512127

13. Michael Carney and Brian Loe, A comparison of methods for implementing adaptive security policies (carney.pdf) Proceedings of the 7th USENIX Security Symposium San Antonio, Texas, January 26-29, 1998

14. A. Cass, B. Lerner, E. McCall, L. Osterweil, A. Wise, Logically central, physically distributed control in a process runtime environment, Technical Report 99-65, University of Massachusetts at Amherst, Nov. 1999

15. B. Charron-Bost, C. Delporte-Gallet, H. Fauconnier, Local and temporal predicates in distributed systems, ACM Transactions on Programming Languages and Systems (TOPLAS) Volume 17, Issue 1 (January 1995), pp: 157 - 179

16. Craig M. Chase, Vijay K. Garg, Detection of global predicates: techniques and their limitations, Workshop on Distributed Algorithms, France, September 1995

17. S. Cheng, D. Garlan, B. Schmerl, J. Sousa, B. Spitznagel, P. Steenkiste, Using architectural style as a basis for system self-repair, Proceedings of the IFIP 17th World Computer Congress - TC2 Stream / 3rd IEEE/IFIP Conference on Software Architecture: System Design, Development and Maintenance, 2002, pp: 45-59

18. D. Chess, C. Palmer, S. White, Security in an autonomic computing environment, IBM Systems Journal, Vol 42, No. 1, 2003, pp: 107-118

19. K. Cheverst, C. Efstratiou, N. Davies, A. Friday, Architectural ideas for the support of adaptive context-aware applications, Workshop on Infrastructure for Smart Devices - How to Make Ubiquity an Actuality HUC 2K, Bristol September 27, 2000

20. J. Cobleigh, L. Osterweil, A. Wise, B. Lerner, Containment units: a hierarchically composable architecture for adaptive systems, ACM SIGSOFT 2002, Charleston, SC, November 2002

21. R. Corin, A. Durante, S. Etalle, P. Hartel, A trace logic for local security properties, International Workshop on Software Verification and Validation (SVV), Mumbai, India, Dec. 2003

22. J. Doyle, I. Kohane, W. Long, H. Shrobe, P. Szolovits, Event recognition beyond signature and anomaly, In Proceedings of the Second IEEE SMC Information Assurance Workshop. IEEE, IEEE Computer Society, June 2001

23. C. Efstratiou, A. Friday, N. Davies and K. Cheverst, Utilising the event calculus for policy driven adaptation in mobile systems, Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY 2002), Monterey, Ca., U.S., J. Lobo, B. J. Michael and N. Duray (eds.), IEEE Computer Society, pp. 13-24, June, 2002

24. D. Evans, A. Twyman, Flexible policy-directed code safety, in IEEE Symposium on Security and Privacy, May 1999

25. R. Feiertag, S. Rho, L. Benzinger, S. Wu, T. Redmond, K. Levitt, D. Peticolas, M. Heckman, Intrusion detection inter-component adaptive negotiation Proceedings of the RAID 99: Recent Advances in Intrusion Detection, West Lafayette, Indiana, USA, September 7-9, 1999

26. R. Filman, T. Linden, SafeBots: a paradigm for software security controls, 1996 ACM New Security Paradigms Workshop, Lake Arrowhead

27. M. Fitzi, U. Maurer, From partial consistency to global broadcast, In Proc. 32nd Annual ACM Symposium on Theory of Computing (STOC), pages 494-503, 2000

28. D. Gabelaia, R. Kontchakov, A. Kurucz, F. Wolter, M. Zakharyaschev, On the computational complexity of spatio-temporal logics, To appear in the proceedings of FLAIRS, 2003

29. V. Garg, J. Mitchell, Distributed predicate detection in a faulty environment, Proceedings IEEE International Conference on Distributed Computing Systems, Amsterdam, Netherlands, 1998

30. D. Garlan, B. Schmerl, J. Chang, Using gauges for architecture-based monitoring and adaptation, Working Conference on Complex and Dynamic Systems Architecture, December 2001, Brisbane, Australia

31. C. Geib, R. Goldman, Information modeling for intrusion report aggregation, Proceedings of the DARPA Information Survivability Conference and Exposition II (DISCEX-II), 2001

32. C. Gill, D. Levine, Quality of service management for real-time embedded information systems, Technical Report, Center for Distributed Object Computing, Dept. of Computer Science, Washington University, St. Louis, MO, 2000

33. C. Gunter, T. Jim, Design of an application-level security infrastructure, DIMACS Workshop on Design and Formal Verification of Security Protocols, September, 1997

34. J. Guttman, Filtering postures: local enforcement for global policies, 1997 IEEE Symposium on Security and Privacy

35. U. Halfmann, W. Kuhnhauser, Embedding security policies into a distributed computing environment, Operating Systems Review, Vol. 33, No. 2 (April 1999)

36. B. Hashii, S. Malabarba, R. Pandey, M. Bishop, Supporting reconfigurable security policies for mobile programs Computer Networks 33 (2000), 77-93

37. D. Hollingworth, T. Redmond, Enhancing operating system resistance to information warfare, MILCOM 2000, 21st Century Military Communications Conference Proceedings

38. T. Jensen, D. Le Metayer, T. Thorn, Verification of control flow based security policies, IEEE Symposium on Security and Privacy, pp 89-103, 1999

39. S. Jiang, R. Kumar, Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications, IEEE Transactions on Automatic Control, 2001, submitted

40. Anita Jones, Cyber security in open systems, in Computer Systems: Theory, Technology and Applications edited by A. Herbert and K. S. Jones, Springer-Verlag, December 2003

41. Z. Kalbarczyk, S. Bagchi, K. Whisnant, R. Iyer, Chameleon: a software infrastructure for adaptive fault tolerance, IEEE Transactions on Parallel and Distributed Systems, Vol. 10, No. 6, June 1999

42. John Keeney, Vinny Cahill, Chisel: A policy-driven, context-aware, dynamic adaptation framework, Policy 2003, Como Italy, June 2003

43. A. Keromytis, J. Parekh, P. Gross, G. Kaiser, V. Misra, J. Nieh, D. Rubenstein, S. Stolfo, A holistic approach to secure survivability, ACM Workshop on Survivable and Self-Regenerative Systems (SRS), held in conjunction with the 10th ACM International Conference on Computer and Communications Security (CCS). October 2003, Fairfax, VA.

44. F. Kerschbaum, E. Spafford, D. Zamboni, Using embedded sensors for detecting network attacks, Journal of Computer Security Volume 10, Issue 1-2, 2002, pp: 23 - 70

45. J. Knight, D. Heimbigner, A. Wolf, A. Carzaniga, J. Hill, P. Devanbu, M. Gertz, The willow survivability architecture, Proceedings of the 4th Information Survivability Workshop, 2001

46. C. Ko, P. Brutch, J. Rowe, G. Tsafnat, K. Levitt, System health and intrusion monitoring (SHIM) using a hierarchy of constraints, Proceedings of 4th International Symposium, RAID, 2001

47. J. Kong, H. Luo, K. Xu, D. Gu, M. Gerla, S. Lu, Adaptive security for multilevel ad hoc networks, Wireless Communications and Mobile Computing, Special Issue on Mobile Ad Hoc Networking, 2002

48. C. Krishna, I. Koren, A. Ganz, C. Moritz, Security tradeoffs in NEST, DARPA presentation, Dec. 2003

49. O. Kupferman, M. Vardi, Synthesizing distributed systems, Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science, 2001

50. W. Lee, J. B. D. Cabrera, A. Thomas, N. Balwalli, Y. Zhang, Performance adaptation in real-time intrusion detection, RAID 2003

51. W. Lee, W. Fan, M. Miller, S. Stolfo, E. Zadok, Toward cost-sensitive modeling for intrusion detection and response, Journal of Computer Security Volume 10, Issue 1-2 2002

52. R. de Lemos, J. Fiadeiro, An architectural support for self-adaptive software for treating faults, Workshop on Self-healing Systems, November 2002

53. Y. Liu, S. Smith, A component security infrastructure Foundations of Computer Security Workshop (FCS'02), Copenhagen, Denmark, July 2002

54. J. Lobo, R. Bhatia, S. Naqvi, A policy description language, in Proc. of AAAI, Orlando, FL, July 1999

55. O. Maler, Control from computer science, IFAC Annual Reviews in Control, 2003

56. D. Malkhi, M. K. Reiter, An architecture for survivable coordination in large distributed systems, Knowledge and Data Engineering, vol 12(2), 2000

57. L. Marcus, Semantics of static, adaptive, and incremental security policies, First Symposium on Requirements Engineering for Information Security (SREIS) March 2001, Indianapolis, Technical Report ATM 2001(8104-05)-1, The Aerospace Corporation July 2001

58. L. Marcus, Local and global requirements in an adaptive security infrastructure, International Workshop on Requirements for High Assurance Systems (RHAS 2003), Sept. 2003

59. S. Merz, M. Wirsing, J. Zappe, A spatio-temporal logic for the specification and refinement of mobile systems, Fundamental Approaches to Software Engineering (FASE 2003), April 2003, Warsaw, Poland

60. J. Millen, Local reconfiguration policies, Proceedings of the 1999 IEEE Symposium on Security and Privacy

61. N. Mittal, V. Garg, On detecting global predicates in distributed computations, Proceedings of the 21st International Conference on Distributed Computing Systems (ICDCS), pp 3-10, April 2001

62. P. Nathan, A trajectory for the evolution of security infrastructure management systems (SIMS) architecture, Symbiot, Inc., Austin, TX, Dec. 2003

63. P. Pal, R. Schantz, J. Loyall, Timeliness in auto-adaptibve distributed systems, Proceedings. 24th International Conference on Distributed Computing Systems Workshops, March 2004.

64. D. Pavlovic, Towards semantics of self-adaptive software, In P. Robertson, R. L., and Shrobe, H., eds., Self-Adaptive Software. Springer-Verlag. 2000, pp: 50-64

65. X. Qie, R. Pang, and L. Peterson, Defensive programming: using an annotation toolkit to build DOS-resistant software, ACM SIGOPS Operating Systems Review Volume 36, Issue SI Winter 2002

66. D. Ragsdale, C. Carver, J. Humphries, U. Pooch, Adaptation techniques for intrusion detection and intrusion response systems, www.itoc.usma.edu/ragsdale/pubs/adapt.pdf

67. R. Rivest, B. Lampson, SDSI – a simple distributed security infrastructure, USENIX 96 and CRYPTO 96, April 30, 1996

68. R. Roshandel, Coupling static and dynamic semantics in an architecture description language, Working Conference on Complex and Dynamic Systems Architecture, December 2001, Brisbane, Australia

69. T. Ryutov, C. Neuman, The specification and enforcement of advanced security policies, Proceedings of the Conference on Policies for Distributed Systems and Networks (POLICY 2002), June 5-7, 2002, Monterey, California

70. J. Salasin, The DARPA ISO DASADA project: Dynamic assembly for systems' adaptability, dependability, and assurance, http://www.rl.af.mil/tech/programs/dasada/program-overview.html

71. R. Sandhu, Decentralized management of security in distributed systems, Procceedings IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, October 15-16, 1991, Santa Barbara, CA

72. F. Schepers, A framework for adaptive security management systems, http://tmf.studentenweb.org/papers/infosec/msc-thesis/html/msc-thesis.html

73. D. Schnackenberg, K. Djahandari, D. Sterne, Infrastructure for intrusion detection and response, Proceedings of the DARPA Information Survivability Conference and Exposition 2000.

74. F. Schneider, Enforceable security policies, Cornell University Technical Report TR98-1664. Jan 1998; also ACM Transactions on Information and System Security Vol 3, No 1, pp 30-50, February 2000

75. D. Scott, A. Beresford, A. Mycroft, Spatial security policies for mobile agents in a sentient computing environment, IEEE POLICY 2003 (4th International Workshop on Policies for Distributed Systems and Networks).

76. L. Semini, C. Montangero, A logic with modalities for asynchronous distributed systems, tp.di.unipi.it/pub/techreports 99-23.ps.Z

77. D. Stewart, P. Khosla, Real-time scheduling of sensor-based control systems, Proceedings of the IFAC/IFIP Workshop, May 15-17, 1991. pp. 139-144.

78. O. Strichman, R. Goldring, Testing, monitoring, and controlling real-time systems with temporal invariants, http://iew3.technion.ac.il/Home/Users/SECOND.php?ofers+Ofer+Strichman+4+4

79. B. Tung, Common intrusion detection framework, 1999, http://www.isi.edu/gost/cidf/

80. P. Venables, Security monitoring in heterogeneous globally distributed environments, Information Security Technical Report, Volume 3, Issue 4, 1998, pp: 15-31

81. R. Venkatesan, S. Bhattacharya, Threat-adaptive security policy, IEEE Journal, 1997, pp: 525-531

82. L. Wagner, Byzantine agreements in secure communication, 5th Operations Research Conference, ASOR 2003

83. D. Wijesekera, S. Jajodia, Policy algebras for access control – the predicate case, Proceedings of the 9th ACM conference on computer and communications security Washington, DC, USA, 2002 pp: 171 - 180

84. D. Wile, Towards a synthesis of dynamic architecture event languages, WOSS ’02 (Workshop on Self-healing Systems, November 2002)

85. T. Wu, M. Malkin, D. Boneh, Building intrusion tolerant applications Proceedings of USENIX Security Symposium, August 1999

86. Q. Zhang, R. Janakiraman, Indra: a distributed approach to network intrusion detection and prevention, Washington University Tech. Report # WUCS-01-30, 2001

87. B. Zhao, A. Joseph, J. Kubiatowicz, Locality aware mechanisms for large-scale networks, Proceedings of Workshop on Future Directions in Distributed Computing (FuDiCo), Bertinoro, Italy, June, 2002