Annotated Bibliography for Computer Systems Security

Jon A. Solworth

April 21, 2010

1 Introduction

This is annotated bibliography for computer systems security. Not all annotations are by the author. Moreover, the author is always interested in any corrections of facts or suggestions for improvement. Also, many of the papers have been difficult to find: if there is an annotation of "–missing–", the author would appreciate a copy of the paper if you have one. The email address of the author is solworth at cs.uic.edu.

The best part of this bibliography, is that you can download the papers by clicking on the on the boxes after the citation.

- If you're client is on the UIC network, click on UIC.
- If you're not on the UIC network, click on Public, although some of these require subscription.

2 Click for category

- Role-Based Access Control
- Distributed protection
- Certificates
- Networks
- Firewall
- Distributed Denial of Service (DDOS)
- Separation of Duty
- Internet RFCs
- Cryptography
- Programming Languages
- Miscellaneous
- Attacks
- Unsorted
- New Paradigms

3 Books

[adams02pki] Carlisle Adams and Steve Lloyd. *Understanding PKI: Concepts, Standards, and Deployment Considerations*. Addison-Wesley, 2nd edition, 2002. UIC

Annotation: This is a good overview of PKI from the implementer's point of view. Relatively high level, implementation independent. But does not give a lot of academic references.

[anderson01securityEngineering] Ross Anderson. *Security Engineering: A Guide to Dependable Distributed Systems*. Wiley, 2001. UIC

Annotation: This is a very, very nice book. Very general. Is not at the textbook level (doesn't go through the details in depth). But I think that the book is enormously precise and carefully written, and there are more details than appear.

[gollmann99computerSecurity] Dieter Gollmann. *Computer Security*. Wiley, 1999. UIC

[pfleeger02security] Charles P. Pfleeger and Shari Lawrence Pfleeger. *Security in Computing*. Pearson Professional Education, 3rd edition, 2002. UIC

Annotation: Broad overall view of security.

[?] [?] [?] [?]

4 Foundations

[abadi93composing] Martn Abadi and Leslie Lamport. Composing specifications. *ACM Transactions on Computing Systems (TOCS)*, 15(1):73–132, 1993. UIC Public

Abstract: A rigorous modular specification method requires a proof rule asserting that if each component behaves correctly in isolation, then it behaves correctly in concert with other components. Such a rule is subtle because a component need behave correctly only when its environment does, and each component is part of the others' environments. We examine the precise distinction between a system and its environment, and provide the requisite proof rule when modules are specified with safety and liveness properties.

[abadi95cojoining] Martn Abadi and Leslie Lamport. Conjoining specifications. *ACM Transactions on Computing Systems (TOCS)*, 17(3):507–535, 1995. UIC Public

Abstract: We show how to specify components of concurrent systems. The specification of a system is the conjunction of its components' specifications. Properties of the system are proved by reasoning about its components. We consider both the decomposition of a given system into parts, and the composition of given parts to form a system.

[abadi99types] Martín Abadi. Secrecy by typing in security protocols. *Journal of the ACM*, 46(5):749–786, 1999. UIC Public

Abstract: We develop principles and rules for achieving secrecy properties insecurity protocols. Our approach is based on traditional classification techniques, and extends those techniques to handle concurrent processes that use shared-key cryptography. The rules have the form of typing rules for a basic concurrent language with cryptographic primitives, the spi calculus. They guarantee that, if a protocol typechecks, then it does not leak its secret inputs.

Annotation: A preliminary version of this paper appeared as [**?**]

[crampton02authorizations] Jason Crampton. *Authorizations and Antichains*. PhD thesis, Birkbeck College, Univ. of London, UK, 2002. UIC

Annotation: Ph.D. thesis, includes proofs of undecidability for both RBAC96 and ARBAC97.

[denning76lattice] Dorothy E. Denning. A lattice model of secure information flow. *Communications of the ACM (CACM)*, 19(5):236–243, 1976. UIC

Annotation: Describes security as a lattice (an extension of a partial order, that is reflexive, transitive, and anti-symmetric)–meaning that it is finite, has a top and a bottom. Forms the basis for Role-based access control.

[harrison75protection] Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. On protection in operating system. In *Symposium on Operating Systems Principles*, pages 14–24, 1975. UIC

Annotation: Paper which showed that general purpose DACs were undecidable in terms of the *safety property*, or whether a subject can can access a given object.

[harrison76protection] Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D. Ullman. Protection in operating systems. *Communications of the ACM (CACM)*, 19(8):461–471, 1976. UIC

Abstract: A model of protection mechanisms in computing systems is presented and its appropriateness is argued. The "safety" problem for protection systems under this model is to determine in a given situation whether a subject can acquire a particular right to an object. In restricted cases, it can be shown that this problem is decidable; i.e., there is an algorithm to determine whether a system in a particular configuration is safe. In general, and under surprisingly weak assumptions, it cannot be decided if a situation is safe. Various implications of this fact are discussed.

[jones75security] Anita K. Jones and Richard J. Lipton. The enforcement of security policies for computation. In *Proceedings of the fifth symposium on Operating Systems Principles*, pages 197–206, 1975. UIC

Annotation: Original separation of protection vs. security.

[jones76takeGrant] A. K. Jones, R. J. Lipton, and L. Snyder. A linear time algorithm for deciding security. In *Proc. 17th Annual Symp. on Foundations of Computer Science*, pages 33–41, 1976. UIC

Annotation: First time that the Take-Grant model for authorizations is introduced. This DAC model is not only decidable, its linear.

[jones78enforcement] Anita K. Jones and Richard J. Lipton. The enforcement of security policies for computation. *Journal of Computer and System Sciences (JCSS)*, 17(1):35–55, 1978. UIC

[koch02decidability] Manuel Koch, Luigi V. Mancini, and Francesco Parisi-Presicce. Decidability of safety in graph-based models for access control. In *Proc. European Symp. Research in Computer Security (ESORICS)*, pages 229–243. LNCS, Springer-Verlag, 2002. UIC

[koch02graph] Manuel Koch, Luigi V. Mancini, and Francesco Parisi-Presicce. A graph-based formalism for RBAC. *ACM Transactions on Information and System Security (TISSEC)*, 5(3):332–365, 2002. UIC

Abstract: Role-Based Access Control (RBAC) is supported directly or in a closely related form, by a number of products. This article presents a formalization of RBAC using graph transformations that is a graphical specification technique based on a generalization of classical string grammars to nonlinear structures. The proposed formalization provides an intuitive description for the manipulation of graph structures as they occur in information systems access control and a precise specification of static and dynamic consistency conditions on graphs and graph transformations. The formalism captures the RBAC models published in the literature, and also allows a uniform treatment of user roles and administrative roles, and a detailed analysis of the decentralization of administrative roles.

Annotation: safety

[lampson74protection] Butler Lampson. Protection. In *ACM Operating Systems Review*, volume 8, pages 18–24. ACM, 1974. UIC

Annotation: This is the paper in which the access matrix is first defined, capabilities and access control lists are also discussed.

[landwehr81formal] Carl E. Landwehr. Formal models for computer security. *ACM Computing Surveys*, 13(3):247–278, 1981. UIC

Annotation: A good overview of security models circa 1981, written from someone on the military security side of things. Puts into perspective a lot of the early work on security and early OS work. Describes military security needs lucidly. Discusses both information flow and access matrix.

[lipton77linearTime] R. J. Lipton and L. Snyder. A linear time algorithm for deciding subject security. *Journal of the ACM*, 24(3):455–464, 1977. UIC

[munawer99atam] Qamar Munawer and Ravi Sandhu. Simulation of the augmented typed access matrix model (ATAM) using roles. In *INFOSECU99: International Conference on Information Security*, 1999. UIC

Annotation: This shows that RBAC can be used to simulated ATAM, thus showing that RBAC is undecidable.

[rushby92noninterference] John Rushby. Noninterference, transitivity, and channel-control security policies. Technical Report CSL-92-02, SRI International, 1992. UIC

Abstract: We consider noninterference formulations of security policies in which the "interferes" relation is intransitive. Such policies provide a formal basis for several real security concerns, such as channel control, and assured pipelines. We show that the appropriate formulation of noninterference for the intransitive case is that developed by Haigh and Young for "multidomain security" (MDS). We construct an "unwinding theorem" for intransitive polices and show that it differs significantly from that of Haigh and Young. We argue that their theorem is incorrect. An appendix presents a mechanically-checked formal specification and verification of our unwinding theorem. We also consider the relationship between transitive and intransitive formulations of security. We show that the standard formulations of noninterference and unwinding correspond exactly to our intransitive formulations, specialized to the transitive case. We show that transitive polices are precisely the "multilevel security" (MLS) polices, and that any MLS secure system satisfies the conditions of the unwinding theorem. In addition, we consider the relationship between noninterference formulations of security and access control formulations, and

we identify the "reference monitor assumptions" that play a crucial role in establishing the soundness of access control implementations.

[rushby93criticalSystems] John Rushby. Critical system properties: Survey and taxonomy. Technical Report SRI-CSL-93-1, Computer Science Laboratory, SRI International, 1994. UIC Public

Abstract: Computer systems are increasingly employed in circumstances where their failure (or even their correct operation, if they are built to flawed requirements) can have serious consequences. There is a surprising diversity of opinion concerning the properties that such "critical systems" should possess, and the best methods to develop them. The dependability approach grew out of the tradition of ultra-reliable and fault-tolerant systems, while the safety approach grew out of the tradition of hazard analysis and system safety engineering. Yet another tradition is found in the security community, and there are further specialized approaches in the tradition of real-time systems. In this report, I examine the critical properties considered in each approach, and the techniques that have been developed to specify them and to ensure their satisfaction. Since systems are now being constructed that must satisfy several of these critical system properties simultaneously, there is particular interest in the extent to which techniques from one tradition support or conflict with those of another, and in whether certain critical system properties are fundamentally compatible or incompatible with each other. As a step toward improved understanding of these issues, I suggest a taxonomy, based on Perrow's analysis (C. Perrow. Normal Accidents: Living with High Risk Technologies. Basic Books, New York, NY, 1984), that considers the complexity of component interactions and tightness of coupling as primary factors.

Annotation: A very lucid description of different approaches towards formal verification of systems with strong properties, including dependability analysis, safety analysis, security, and real-time systems. Describes a taxonomy to enables a (partial) integration of the techniques although some seem to be in conflict with others.

[sandhu92typed] Ravi S. Sandhu. The typed access matrix model. In *Proc. IEEE Symp. Security and Privacy*, pages 122–136, 1992. UIC

Abstract: The typed access matrix (TAM) model is defined by introducing the notion of strong typing into the Harrison, Ruzzo, and Ullman model (HRU) (M. H. Harrison et al., 1978). It is shown that

monotonic TAM (MTAM) has decidable, but NP-hard, safety for its acyclic creation cases. It is further shown that ternary MTAM has polynomial time safety analysis for its acyclic cases, even though it is, in general, equivalent to MTAM. Ternary MTAM thus has strong safety properties. The expressive power of ternary MTAM has been shown to be equivalent to MTAM in general. The results establish that strong typing is crucial to achieving a useful demarcation between decidable and undecidable safety, and ternary monotonic commands are critical for tractable safety analysis.

Annotation: The typed access matrix (TAM) model adds a type to each object and subject in the access matrix which does not change. The model has a decidable and tractable case but if this doesn't hold is undecidable.

[schaefer93foundations] M. Schaefer. We need to think about the foundations of computer security. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 120–125. IEEE, 1993. UIC

[snyder81formalCapability] Larry Snyder. Formal models of capability-based protection systems. *IEEE Trans. Comput.*, C-30(3):172–181, 1981. UIC

[solworth04dacDecidability] Jon A. Solworth and Robert H. Sloan. A layered design of discretionary access controls with decidable properties. In *Proc. IEEE Symp. Security and Privacy*, pages 56–67, 2004. UIC

Annotation: The Discretionary Access Control models of Osborne, Sandhu, and Munawer are shown to be implementable in a model whose safety property is decidable. Also introduces a novel group structure and a first class relabel permission/operation.

[soshi00dynamic] Masakazu Soshi. Safety analysis of the dynamic-typed access matrix model. In *Proc. European Symp. Research in Computer Security (ESORICS)*, volume 1895 of *Lecture Notes in Computer Science*, pages 106–121. Springer-Verlag, 2000. UIC

Annotation: A dynamic, decidable subset (if certain runtime conditions hold) of TAM.

5 Security Policies

[bell73secure] D. E. Bell and L. J. LaPadula. Secure computer systems: Mathematical foundations and model. Technical Report M74-244, Mitre Corporation, Bedford MA, 1973. UIC

> **Annotation:** The original multi-level security paper, defining the simple security property (ss-property) which is that a principle cannot read above his level (also called *no read up*) and the *-property, which means cannot write below the high level sources *no write down*.

[bell76unified] D. Bell and L. LaPadula. Secure computer systems: Unified exposition and Multics interpretation. Technical Report MTR-2997, MITRE Corp., Bedford, MA, July 1976. UIC

> **Annotation:** The big tech report, with everything in it: models, proof, and Multics implementation.

[berger90cmw] Jeffrey L. Berger, Jeffrey Picciotto, John P. L. Woodward, and Paul T. Cummings. Compartmented mode workstation: Prototype highlights. *IEEE Transactions on Software Engineering*, 16(6):608–618, 1990. Special Section on Security and Privacy. UIC

> **Annotation:** A Compartmented Mode Workstation (CMW) is a late 1980s design to extend COTS workstation technology to high security levels. Window system is well integrated with security, offering trusted path, labeling of window security, line-by-line security labeling for terminal windowing, and ability to launch programs at different security levels. Each object is labeled by a fixed sensitivity label and a floating information label enabling flexibility without having labels float arbitrarily high.

[beznosov98healthcare] Konstantin Beznosov. Requirements for access control: US healthcare domain. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 43–46, New York, 1998. ACM Press. UIC

> **Annotation:** 1 page description of healthcare protections. Interesting that location plays a part in determining access (eg. nurse on the same floor in a hospital).

[biba77integrity] K. Biba. Integrity considerations for secure computer systems. Technical Report TR-3153, MITRE Corp, Bedford, MA, 1977. UIC

[boebert85practical] W. E. Boebert and R. Kain. A practical alternative to hierarchical integrity policies. In *8th National Computer Security Conference*, pages 18–27, 1985. UIC

Annotation: This is the original *Type enforcement* paper.

[brewer89chinese] D. F. C. Brewer and M. J. Nash. The Chinese Wall security policy. In *Proc. IEEE Symp. Security and Privacy*, pages 206–214, 1989. UIC

Annotation: Chinese Wall is a security policy for an investment bank. An investment bank has confidential information on different corporate clients. It is illegal for any individual to have access to confidential information on more than one company in a given industry. Initially, an individual can access any company, but once a company is accessed no other company can be accessed. Moreover, confidential information from one company cannot be put in another company's objects.

[clark87commercial] David D. Clark and David R. Wilson. A comparison of commercial and military computer security policies. In *Proc. IEEE Symp. Security and Privacy*, pages 184–194, 1987. UIC

Annotation: Argues that the mechanisms for commercial needs rest more on integrity and is best characterized by well formed transactions (double entry bookkeeping) and separation of duties.

Requirements are (1) Authentication (2) require data to be manipulated only by certain programs (3) Associate users with the programs they can run (proper assignment ensures separation of duty), (4) Audit log. In addition, administrative controls are needed to ensure the configuration changes only in well-defined ways.

To ensure correctness, Integrity Verification Procedure (IVP) are used to ensure the current state is sound and Transformation Procedures to change it.

Talks about TP ensuring, for example, time of day restrictions. Other examples is the order of TPs.

It separates enforcement (application independent) from certification (application dependent issues).

[lampson91computerSecurity] Butler Lampson. *Computers at Risk*, chapter Requirements and Technology for Computer Security, pages 74–191. National Academy Press, Washington, D.C., 1991. UIC

[longstaff00healthcare] J. J. Longstaff, M. A. Lockyer, G. Capper, and M. G. Thick. A model of accountability, confidentiality and override for healthcare and other

applications. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 71–76. ACM Press, 2000. UIC

[mcdaniel02securityPolicyReconciliation] Patrick McDaniel and Atul Prakash. Methods and limitations of security policy reconciliation. In *Proc. IEEE Symp. Security and Privacy*, pages 73–87, 2002. UIC

[meadows90multilevel] C. Meadows. Extending the Brewer-Nash model to a multilevel context. In *Proc. IEEE Symp. Security and Privacy*, pages 95–102, 1990. UIC

Abstract: In this paper we show how the Brewer-Nash Chinese Wall model can be extended to a policy for handling the aggregation problem in a multilevel context. We derive a lattice-based information flow policy that can be integrated into both the multilevel and Brewer-Nash context. We use this information flow policy to develop a security policy described in terms of labeled subjects accessing labeled objects that will allow us to construct a system that prevents users from accessing aggregates that they are not cleared to see.

Annotation: The idea here is that aggregates may have more information than any of their individual components, so that it is necessary to associate security labels with the aggregate rather than use the default Bell-LaPadula max of inputs.

[minear00secureOs] Spence Minear. Secure OS: A working example of a secure BSD OS. Technical report, Secure Computing, 2000. UIC

Annotation: Very readable, but too brief, description of Type Enforcement (TE) on top of BSD.

[saltzer75protection] J. H. Saltzer and M. D. Schroeder. The protection of information in computer system. *Proceedings of the IEEE*, 63(9):1278–1308, 1975. UIC

Annotation: Defines least privilege.

[woodward87cmw] John P. L Woodward. Security requirements for system high and compartmented mode workstations. Technical Report MTR 9992, Revision 1, The MITRE Corporation, Bedford, MA, November 1987. Also published by the Defense Intelligence Agency as document DDS-2600-5502-87. UIC

6 Kernel Protection

[badger95practical] Lee Badger, Daniel F. Sterne, David L. Sherman, Kenneth M. Walker, and Sheila A. Haghihat. Practical domain and type enforcement for UNIX. In *Proc. IEEE Symp. Security and Privacy*, pages 66–77, Oakland, CA, 1995. UIC

Abstract: Type enforcement is a table-oriented mandatory access control mechanism well-suited for confining applications and restricting information flows. Although both flexible and strong, type enforcement alone imposes significant administrative costs and has not been widely adopted. Domain and Type Enforcement (DTE) is an enhanced version of type enforcement designed to provide needed simplicity and compatibility. Two primary techniques distinguish DTE from simple type enforcement: DTE policies are expressed in a high-level language that includes file security attribute associations as well as other access control information; and during system execution, DTE file security attributes are maintained using a concise human-readable format in a runtime DTE policy database, thus removing the need for security-specific low-level data formats. Such formats are a major source of incompatibility for security-enhanced systems. A DTE UNIX prototype system has been implemented to evaluate these primary DTE concepts. This paper presents experiences gained and preliminary results indicating that DTE can provide cost effective security increases to UNIX systems while maintaining a high degree of compatibility with existing programs and media.

Annotation: Describes the ideas behind Domain and Type Enforcement, based on Type Enforcement. The idea is that the access matrix is partitioned into equivalence classes, with domains partitioning subjects and types partitioning objects. These seem to be layered on top of Unix DAC.

The *domain* that a user is in is determined by rules which specify the domain for `init`, automatic transition rules (via exec), and process-induced changes of domain: The last notably for login.

The *types* of objects define from which domains they can be accessed. The types are *assigned* to points in the directory tree, and this is inherited by all subtrees until there is a new assignment. Assignments can also be added dynamically, and it is necessary to disambiguate multiple assignments for multilinked files. Assignment removes the per file need for attributes.

Describes in future work the possibility of putting in predicates to test information flow. –jas

[badger95prototype] Lee Badger, Daniel F. Sterne, David L. Sherman, Kenneth M. Walker, and Sheila A. Haghihat. A domain and type enforcement UNIX prototype. In *Proc. of the USENIX Security Symposium*, Salt Lake City, 1995. UIC

[chen02mops] Hao Chen and David Wagner. Mops: an infrastructure for examining security properties of software. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 235–244. ACM Press, 2002. UIC

[corbato72multics] F. J. Corbato, J. H. Saltzer, and C. T. Clingen. Multics – the first seven years. In *Spring Joint Computer Conference*, pages 571–583. AFIPS Press, 1972. UIC

Annotation: Overview of Multics. See more also on `http://www.multicians.org`. Notable features of Multics (extract from above site): - Segmented memory - Virtual memory (unified with filesystem using segments?) - High-level language implementation (PL/I) - Shared memory multiprocessor - Multi-language support - Relational database - Security (protection rings, MAC, first (for long time) to get rated B2) - On-line (hard- and software) reconfiguration (to run 7/24) - Software Engineering (disciplined way: design before code, high level language, design and code review, structured programming, modularization and layering were all employed extensively to manage the complexity of the system, which was one of the largest software development efforts of its day.) - Dynamic linking

[cox02plan9Security] Russ Cox, Eric Grosse, Rob Pike, Dave Presotto, and Sean Quinlan. Security in Plan 9. In *Proc. of the USENIX Security Symposium*, pages 3–16, 2002. UIC Public

Annotation: Best conference paper award.

Describes the design and use of a per user agent called `factotum` which holds a copy of the user keys and negotiates authentication protocols.

[foley96dynamicLabeling] Simon Foley, Li Gong, and Xiaolei Qian. A security model of dynamic labeling providing a tiered approach to verification. In *Proc. IEEE Symp. Security and Privacy*, pages 142–154, 1996. UIC

Annotation: Describes the basis for a MLS kernel design which has rewrite rules on labels.

[foley97canonicalUpgrade] Simon N. Foley. Supporting secure canonical upgrade policies in multilevel secure object stores. In *Proc. of the Annual Computer Security Applications Conference (ACSAC)*, pages 69–80, 1997. UIC

Annotation: This is the second in a series, positioned not at kernels but at object stores.

[foley98kernelized] Simon N. Foley. A kernelized architecture for multilevel secure application policies. In *Proc. European Symp. Research in Computer Security (ESORICS)*, pages 33–49, 1998. UIC

Annotation: The third in a series. Back to kernels

[gibson02multiDomain] Timothy J. Gibson. An architecture for flexible multi-security domain networks. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, pages 63–72. The Internet Society, 2001. UIC

[jaeger01accessControl] Trent Jaeger. Managing access control complexity using metrics. In *Proceedings of the Sixth ACM Symposium on Access control models and technologies*, pages 131–139. ACM Press, 2001. UIC

[jaeger02accessSpaces] Trent Jaeger, Antony Edwards, and Xiaolan Zhang. Managing access control policies using access control spaces. In *Seventh ACM Symposium on Access Control Models and Technologies*, pages 3–12. ACM Press, 2002. UIC

[jaeger94fileSystemSecurity] Trent Jaeger and Atul Prakash. Support for the file system security requirements of computational e-mail systems. In *Proceedings of the 2nd ACM conference on Computer and Communications Security*, pages 1–9. ACM Press, 1994. UIC

[koch01accessControlPolicies] M. Koch, L. V. Mancini, and F. Parisi-Presicce. On the specification and evolution of access control policies. In *Proceedings of the Sixth ACM Symposium on Access control models and technologies*, pages 121–130. ACM Press, 2001. UIC

[lampson83hints] Butler W. Lampson. Hints for computer system design. In *Proceedings of the 9th Symposium on Operating Systems Principles, Operating Systems Review*, pages 33–48, 1983. Published as Proceedings of the 9th Symposium on Operating Systems Principles, Operating Systems Review, volume 17, number 5. UIC

Annotation: An excellent paper touting the need for simplicity, functionality, fault-tolerance, and speed. It is the the operating systems equivalent to Alan Perlis' "Epigrams on Programming." Another paper of a similar vein from Xerox is Lauer's 1981 SOSP paper on Operating systems development. Also reproduced in "IEEE Software" (USA), v1, n1, pp. 11-28, 55 REFS. Treatment PRACTICAL, Jan. 1984, digital computers, computer system design.

[obrien91lockApplications] R. O'Brien and C. Rogers. Developing applications on LOCK. In *Proc. 14th NIST-NCSC National Computer Security Conference*, pages 147–156, 1991. UIC

Abstract: The Logical Coprocessing Kernel (LOCK) system is a highly assured INFOSEC system that can be used as a platform to develop countermeasures to current and future security threats. In this paper we discuss the manner in which applications are developed on LOCK and the features of the LOCK system that allow these applications to be developed quickly and securely. The paper focuses on the design of such applications using LOCK's type enforcement and the implementation of these applications using the current LOCK software development environment.

Annotation: This is a very coherent description of Type Enforcement (TE) and was written some time after the original paper by Boebert-Kain [**?**]. There is a table of subjects vs. objects. The subjects are associated with "modules". Each object has a type and subjects have a domain. Roles enable users to execute within domains.

TE uses a lattice and allows trusted users to violate the lattice for write (but not read) permission.

[posix1] IEEE/ANSI Std. 1003.1. Portable Operating System Interface (POSIX) Part 1: System Application Program Interface (API) [C Language], 1996. UIC

Annotation: The original POSIX standard which describes the base DAC model.

[posix1003-1e] IEEE/ANSI Draft Std. 1003.1e. Draft Standard for Information Technology–POSIX Part 1: System API: Protection, Audit and Control Interface, 1997. UIC

Annotation: This standard effort was eventually abandoned, but was still very influential and parts of it are widely implemented. This includes ACLs, capabilities, support of MLS.

[rushby81secure] John Rushby. Design and verification of secure systems. In *8th ACM Symposium on Operating System Principles*, volume 15, pages 12–21, 1981. UIC

[rushby89safety] John Rushby. Kernels for safety? In T. Anderson, editor, *Safe and Secure Computing Systems*, chapter 13, pages 210–220. Blackwell Scientific Publications, 1989. (Proceedings of a Symposium held in Glasgow, October 1986). UIC

Annotation: This paper contains a very lucid and early understanding of what it is that kernel-based protections can achieve. It points out that kernels cannot ensure that "good things happen" but rather can ensure that "bad things don't happen".

[somayaji98immuneSystem] Anil Somayaji, Steven Hofmeyr, and Stephanie Forrest. Principles of a computer immune system. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 75–82. IEEE, 1998. UIC

[thomsen90comparison] D. J. Thomsen and J. T. Haigh. A comparison of type enforcement and unix setuid implementation of well-formed transactions. In *In Proceedings of Sixth Annual Computer Security Applications Conference*, pages 304–312, 1990. UIC Public

[tidswell00integratedConstraints] Jonathan F. Tidswell and Trent Jaeger. Integrated constraints and inheritance in DTAC. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 93–102, 2000. UIC

Annotation: Describes an extension to DTAC to support constraints and inheritence. DTAC (Dynamic Typed Access Control) is, by itself, an extension of DTE.

[tidswell97dynamicDte] Jonathon Tidswell and John Potter. An approach to dynamic domain and type enforcement. In *Australian Conference on Information Security and Privacy*, pages 26–37, 1997. UIC

Annotation: DTE partitions processes into domains, and objects into types. For entry in such a table it provides a set of permissions.

[zelemZpSecurity] Marek Zelem and Milan Pikula. Zp security framework. available at http://medusa.fornax.sk/. UIC Public

Annotation: Implemented in Linux, available on the web.

7 Application Security

[ioannidis01secureWebBrowser] Sotiris Ioannidis and Steven M. Bellovin. Building a secure web browser. In *Proceedings of the FREENIX Track (FREENIX-01)*, pages 127–134, Berkeley, CA, 2001. The USENIX Association. UIC

8 Covert channels and non-interference

[goguen82noninterference] J. A. Goguen and J. Meseguer. Security policies and security models. In *Proc. IEEE Symp. Security and Privacy*, pages 11–20, 1982. UIC

Annotation: The original paper which defined non-interference and hence made the idea of covert channels significantly more precise. Two principals don't interference with each other if nothing one does effects the other.

[gray90probabilistic] III. J. W. Gray. Probabilistic interference. In *Proc. IEEE Symp. Security and Privacy*, pages 170–179, 1990. UIC

Abstract: Probabilistic interference in nondeterministic machines can be exploited by trojan horses to reliably leak information to unauthorized users. This problem has been noted by other researchers but has not previously been addressed. We extend McCullough's restrictiveness to additionally prevent probabilistic interference. Then, to illustrate the use of our extension, we develop a nondeterministic system that solves a denial of service problem and use our definition to prove that the system is secure.

[kang98nrlPump] Myong H. Kang, Andrew P. Moore, and Ira S. Moskowitz. Design and assurance strategy for the NRL pump. *Computer*, 31(4):56–64, 1998. UIC Public

Annotation: The NRL Pump is an almost "one way" buffer from low security to high security, with the intention of preventing covert flows from high to low. It succeeds only partially in this since the low application is informed about problems both via acknowledgements and reliability. The NRL Pump also provides some primitive firewall capability, allowing the specification of what types of connections to accept.

[lampson73note] Butler W. Lampson. A note on the confinement problem. *Communications of the ACM (CACM)*, 16(10):613–615, 1973. UIC

Annotation: First paper describing protection goal (in terms of secrecy) and the first paper defining a covert channel. Describes the need to confine (confidentiality) information to a computer system and the mechanism for doing it. Describes both covert and overt channels.

[mclean85basicSecurityTheorem] John McLean. A comment on the 'basic security theorem' of Bell and LaPadula. *Information Processing Letters*, 20(2):67–70, 1985. UIC

[mclean92noninterference] John McLean. Proving noninterference and functional correctness using traces. *Journal of Computer Security*, 1(1), 1992. UIC

[mclean94general] John McLean. A general theory of composition for trace sets closed under selective interleaving functions. In *Proc. IEEE Symp. Security and Privacy*, pages 79–93, 1994. UIC

[mclean94securityModels] John McLean. Security models. In J. Marciniak, editor, *Encyclopedia of Software Engineering*. Wiley & Sons, 1994. UIC

Annotation: A theoretical discussion of security models based on information flow. While providing a finer grain view of information than access controls, it does not allow for encryption as reducing information flow. Since it is in general concerned with backdoors, and encryption is not provably secure, this is a fundamental consequence.

[millen99twentyYears] J. Millen. Twenty years of covert channel modeling and analysis. In *Proc. IEEE Symp. Security and Privacy*, pages 20–114, 1999. UIC

[rushby92noninterference] John Rushby. Noninterference, transitivity, and channel-control security policies. Technical Report CSL-92-02, SRI International, 1992. UIC

Abstract: We consider noninterference formulations of security policies in which the "interferes" relation is intransitive. Such policies provide a formal basis for several real security concerns, such as channel control, and assured pipelines. We show that the appropriate formulation of noninterference for the intransitive case is that developed by Haigh and Young for "multidomain security" (MDS). We construct an "unwinding theorem" for intransitive polices and show that it differs significantly from that of Haigh and Young. We argue that their theorem is incorrect. An appendix presents a mechanically-checked formal specification and verification of our unwinding theorem. We also consider the relationship between transitive and intransitive formulations of security. We show that the standard formulations of noninterference and unwinding correspond exactly to our intransitive formulations, specialized to the transitive case. We show that transitive polices are

precisely the "multilevel security" (MLS) polices, and that any MLS secure system satisfies the conditions of the unwinding theorem. In addition, we consider the relationship between noninterference formulations of security and access control formulations, and we identify the "reference monitor assumptions" that play a crucial role in establishing the soundness of access control implementations.

[ryan01noninterference] P. Ryan, J. McLean, J. Millen, and V. Gligor. Non-interference, who needs it? In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW)*, pages 237–240. IEEE, 2001. UIC

Annotation: Non-interference is difficult to understand, no agreed upon definition is sufficient, but it may be increasing in importance.

[shieh90covertStorage] S. W. Shieh and V. D. Gligor. Auditing the use of covert storage channels in secure systems. In *Proc. IEEE Symp. Security and Privacy*, pages 285–295, 1990. UIC

Abstract: In this paper we define requirements for auditing covert storage channels, and illustrate some fundamental problems which appear in most computer systems. We argue that audit subsystems designed to minimally satisfy the TCSEC requirements [1,2] are unable to detect many instances of covert channel use, and hence require major design and implementation changes before they are able to detect all use of covert storage channels. Finally, we present the design of a Secure Xenix(*) tool for covert-channel audit that has been in operation since July 1989. Results of experiments indicate that the tool is able to detect all use of covert storage channels without raising false alarms.

[zakinthinos97general] A. Zakinthinos and E. S. Lee. A general theory of security properties. In *Proc. IEEE Symp. Security and Privacy*, pages 94–102, 1997. UIC

9 Role-Based Access Control

[barkley97roleAcl] John Barkley and Anthony Cincotta. Comparing simple role based access control models and access control lists. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 73–80, 1997. UIC

Annotation: Very nice construction, using POSIX 1e capabilities, of role based accesses control using POSIX groups.

[chen95constraints] F. Chen and Ravi S. Sandhu. Constraints for role-based access control. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, Gaithersburg, Maryland, 1995. ACM. UIC

[ferraiolo92rbac] David F. Ferraiolo and Richard Kuhn. Role based access control. In *15th National Computer Security Conference*, pages 554–563, Baltimore, MD, 1992. UIC

Annotation: This is the original role-based access control. A user acts in a capacity which is a role. Roles correspond to groups of users which can then be assigned permissions. The permissions are similar to Clark-Wilson, the ability to perform TP on some data sets. Talks also about role inheritence and also about administration of roles.

[friberg97support] Christian Friberg and Achim Held. Support for discretionary role-based access control in ACL-oriented operating systems. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 83–94, Fairfax, VA, 1997. ACM. UIC

[giuri96natural] Luigi Giuri. Role based access control: A natural approach. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages II 33–37. ACM, 1996. UIC

Annotation: "A role is defined as a named set of protection domains." Defines a simultaneous set of roles enabled and an exclusive set. Very vaguely defined, only other reference to it is in Italian.

[gligor96characteristics] Virgil Gligor. Characteristics of role-based access control. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages II 9–14. ACM, 1996. UIC

[jaeger00rebuttal] Trent Jaeger and Jonathon E. Tidswell. Rebuttal to the NIST RBAC model proposal. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 65–66, 2000. UIC

Annotation: A rebuttal to the above paper. Proposal does not provided a means to specify models, and the concepts are vaguely defined. The levels, are not levels, but orthogonal properties (level 2 is not needed for level 3). Formalization of permissions, users into groups which are assigned to roles, and constraints. Missing is the hierarchical role administration.

[jaeger95rbacCollaborative] Trent Jaeger and Atul Prakash. Requirements of role-based access control for collaborative systems. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 16–27. ACM Press, 1995. UIC Public

[jaeger99constraints] Trent Jaeger. On the increasing importance of constraints. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 33–42, 1999. UIC

Annotation: Describes why constraints are increasingly important, although the can lead to computability and efficiency issues. Many of the constraints he suggest are because of a lack of labels on on data elements.

[lupu97reconciling] Emil C. Lupu and Morris Sloman. Reconciling role-based management and role-based access control. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 135–142. ACM Press, 1997. UIC

Annotation: Discusses some issues in role based management, a superset of RBAC, since it contains obligations as well as permissions. (It discussed negative obligations, but this seem to me to be the same as a lack of privileges.) It seems to me that RBAC has to do with the form of what can be done, while RBM with the substance and so the latter is much more difficult.

[moffett98control] Jonathan D. Moffett. Control principles and role hierarchies. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, 1998. UIC

Annotation: Describes how inheritence in Role Hierarchy can conflict with separation of duties. This probably can be handled either through constraints or limiting inheritence.

[moffett99hierarchy] J. Moffett and E. Lupu. The uses of hierarchies in access control. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, 1999. UIC

Annotation: Follow on to [**?**] about problems of role hierarchies. Interesting discussion about delegations.

[osborn00MacDac] Sylvia Osborn, Ravi Sandhu, and Qamar Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security (TISSEC)*, 3(2):85–106, 2000. UIC

Annotation: Constructions for Bell-LaPadula and various DACs are described and proved correct.

[osborn00modelingUsers] Sylvia Osborn and Yuxia Guo. Modeling users in role-based access control. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 31–38. ACM Press, 2000. UIC

Annotation: Describes how a group hierarchy, which includes individual users can be integrated with a role hierarchy.

[osborn97rbacMac] Sylvia Osborn. Mandatory access control and role-based access control revisited. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 31–40. ACM Press, 1997. UIC

Annotation: Analyzes role graphs to check if they meet Bell-LaPadula mandatory access controls.

[sandhu00nist] Ravi Sandhu, David Ferraiolo, and D. Richard Kuhn. The NIST model for role-based access control: Towards a unified standard. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 47–64, 2000. UIC

Annotation: Describes a National Institute of Standards proposal for what constitutes role-based access control. The structure is either Flat, Hierarchical, or Constraint-Based or Symmetric.

[sandhu92chineseWall] Ravi Sandhu. Lattice-based enforcement of Chinese Walls. *Computers and Security*, 11(8):753–763, 1992. UIC

Annotation: Journal version of [**?**].

[sandhu92chineseWallconference] Ravi Sandhu. A lattice interpretation of the chinese wall security policy. In *Proc. 15th NIST-NCSC National Computer Security Conference*, pages 329–339, Washington, D.C., 1992. US Govt. Printing Office. UIC

Abstract: The Chinese Wall policy was identified and so named by Brewer and Nash. This policy arises in the segment of the commercial sector which provides consulting services to other companies. Consultants naturally have to deal with confidential company information for their clients. The objective of the Chinese Wall policy is to prevent information flows which cause conflict of interest for individual consultants. Brewer and Nash develop a mathematical model of the Chinese Wall policy, on the basis of which they claim that this policy "cannot be correctly represented by a Bell-LaPadula model." In this paper we demonstrate that the Brewer-Nash model is too restrictive to be employed in a practical system. This is due to their treatment of users and subjects as synonymous concepts, with the consequence that they do not distinguish security policy as applied to human users versus security

policy as applied to computer subjects. By maintaining a careful distinction between users, principals and subjects, we show that the Chinese Wall policy is just another lattice-based information policy which can be easily represented within the Bell-LaPadula framework.

[sandhu93latticeBased] Ravi S. Sandhu. Lattice-based access control models. *IEEE Computer*, 26(11):9–19, 1993. UIC

Annotation: Describes how Denning's Lattice Models can be used to model Bell-LaPadula, Chinese Wall, and the Biba model. The first two are confidentiality models, the last is an integrity model.

Points out some of the limitations of Lattice-Based Access control models including transitivity required and hence cannot force intermediate states.

[sandhu96roleBased] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996. UIC

Annotation: This is a preview of the NIST unified model for RBACs, and similar to the 1993 computer based model.

[sandhu99arbac] Ravi Sandhu, Venkata Bhamidipati, and Qamar Munawer. The ARBAC97 model for role-based administration of roles. *ACM Transactions on Information and System Security (TISSEC)*, 2(1):105–135, 1999. UIC

Annotation: Roles to administer roles. A collection of rules for administering the changing system as a result of adding/removing users to roles, new roles, ... The article discusses the definitions, and some of their rationale but no proofs of properties.

[tidswell99dynamicRights] J. E. Tidswell, G. H. Outhred, and J. M. Potter. Dynamic rights: Safe extensible access control. In *Proc. 4th ACM Workshop on Role-Based Access Control*, pages 113–120, 1999. UIC

Abstract: Extensible systems such as micro-kernels and component architectures push current security models to the limit. A number of dynamic access control models have been developed but all fail to ensure safety, especially of large scale configurations. In previous work we have developed a dynamic typed access control (DTAC) model that supports generalized security configuration descriptions based on subject and object types. This model includes a security invariant to ensure safety in the presence of change. In this paper we investigate the use of structured

subject types, structured object types and structured rights to simplify both modeling and safety enforcement within DTAC. Structuring all aspects of the access control relation is both promising and novel.

10 Objects

[jajodia90multilevel] S. Jajodia and B. Kogan. Integrating an object-oriented data model with multilevel security. In *Proc. IEEE Symp. Security and Privacy*, pages 76–85, 1990. UIC

Abstract: We present a new security model for object-oriented database systems. This model is a departure from the traditional security models based on the passive object — active subject paradigm. Our model is a flow model whose main elements are objects and messages. An object combines the properties of a passive information repository with those of an active agent. Messages are the main instrument of information flow. The chief advantages of the proposed model are its compatibility with the object-oriented data model and the simplicity with which security policies can be stated and enforced.

[sterne99scalable] Daniel F. Sterne, Gregg W. Tally, C. Durvard McDonald, David L. Sherman, David L. Sames, Pierre X. Pasturel, and E. John Sebes. Scalable access control for distributed object systems. In *Proc. of the USENIX Security Symposium*, pages 201–214. Usenix Association, 1999. UIC

Annotation: Describes how DTE protection is added to a CORBA framework to provide security at the level of object methods. The DTE model has types, which is the security tags and domains which are the roles. Then, methods are labeled with types, and domains are allowed to access certain types.

11 Rainbow Series

[blakely97tcb] Bob Blakley and D. M. Kienzle. Some weaknesses of the tcb model. In *Proc. IEEE Symp. Security and Privacy*, pages 3–5, 1997. UIC

Annotation: The problems seem to revolve around two issues: "What constitutes trust?" and "What to do about integrity?".

[chokhani92trustedEvaluation] Santosh Chokhani. Trusted products evaluation. *Communications of the ACM (CACM)*, 35(7):64–76, 1992. UIC

[dod85trustedComputer] Department of Defense. Trusted computer system evaluation criteria. Technical Report DOD 5200.28–STD, U. S. Department of Defense, 1985. UIC

[gligor93covertChannel] Virgil Gligor. A guide to understanding covert channel analysis of trusted systems. Technical Report NCSC-TG-030, National Computer Security Center, Ft. George G. Meade, Maryland, U.S.A., November 1993. Approved for public release: distribution unlimited. UIC

Abstract: This is the official NSA guide to the identification and elimination of covert channels in multilevel secure systems. Military multilevel secure systems – at least at the higher levels of evaluation – should limit covert channel bandwidth to about one bit per second. The techniques involved include both channel elimination and noise insertion.

[mclean97tcb] John McLean. Is the trusted computer base fundamentally flawed? In *Proc. IEEE Symp. Security and Privacy*, page 2, 1997. UIC

[shockley97tcb] William R. Shockley and James P. Downey. Is the reference monitor fundamentally flawed? a case for the negative. In *Proc. IEEE Symp. Security and Privacy*, pages 6–7, 1997. UIC

12 OS implementation

[arbaugh97reliableBootstrap] William A. Arbaugh, David J. Farber, and Jonathan M. Smith. Reliable bootstrap architecture. In *Proc. IEEE Symp. Security and Privacy*, pages 65–71, 1997. UIC

Abstract: In a computer system, the integrity of lower layers is treated as axiomatic by higher layers. Under the presumption that the hardware comprising the machine (the lowest layer) is valid, integrity of a layer can be guaranteed if and only if: (1) the integrity of the lower layers is checked, and (2) transitions to higher layers occur only after integrity checks on them are complete. The resulting integrity "chain" inductively guarantees system integrity. When these conditions are not met, as they typically are not in the bootstrapping (initialization) of a computer system, no integrity guarantees can be made. Yet, these guarantees are increasingly important to diverse applications such as Internet commerce, intrusion detection systems, and "active networks." In this paper, we describe the AEGIS architecture for initializing a computer system. It validates integrity at each layer transition in the bootstrap process. AEGIS also includes a recovery process for integrity check failures, and we show how this results in robust systems. We discuss our prototype implementation for the IBM personal computer (PC) architecture, and show that the cost of such system protection is surprisingly small.

Annotation: This is the basis for the paladium process, Microsoft's new attempt to be big brother.

[chen02setuid] Hao Chen, David Wagner, and Drew Dean. Setuid demystified. In *Proc. of the USENIX Security Symposium*. USENIX, 2002. UIC Public

Abstract: Access control in Unix systems is mainly based on user IDs, yet the system calls that modify user IDs (uid-setting system calls), such as setuid, are poorly designed, in-sufficiently documented, and widely misunderstood and misused. This has caused many security vulnerabilities in application programs. We propose to make progress on the setuid mystery through two approaches. First, we study kernel sources and compare the semantics of the uid-setting system calls in three major Unix systems: Linux, Solaris, and FreeBSD. Second, we develop a formal model of user IDs as a Finite State Automaton (FSA) and develop new techniques for automatic construction of such models. We use the resulting FSA to uncover pitfalls in the Unix API of the uid-setting system calls, to identify differences in the semantics of these calls among various Unix systems, to detect inconsistency in the handling of user IDs within an OS kernel, and to check the proper usage of these calls in programs automatically. Finally, we provide general guidelines on the proper us-age of the uid-setting system calls, and we propose a high-level API that is more comprehensible, usable, and portable than the usual Unix API.

[cowan00subdomain] Crispin Cowan, Steve Beattie, Greg Kroah-Hartman, Calton Pu, Perry Wagle, and Virgil Gligor. Subdomain: Parsimonious security server. In *14th Systems Administration Conference (LISA 2000)*, pages 355–367, New Orleans, LA, 2000. UIC

Annotation: Subdomains for domain and type enforcement.

[fraser00lomac] Timothy Fraser. LOMAC–low water-mark integrity protection for COTS environments. In *Proc. IEEE Symp. Security and Privacy*, 2000. UIC

Annotation: Low water-mark protection lowers the protection level of a subject based on the level of objects it observes. The author uses it to make a security monitor for Linux (using a Loadable Kernel Module), which does not require any changes to applications. The security monitor prevents damage to system binaries and configuration files by putting them at a higher level then "ordinary" root objects.

The paper seems to go through a lot of trouble to say that the level of a subject (process) is the minimum level of the objects it observes transitively.

Problems with the approach: 1) Protects only system objects and difficult to extend to user objects, 2) Difficult to protect logs, which must reside at most restricted level, 3) User behavior changes require, for example, cannot edit level 1 and level 2 in the same session and then save level 2.

[fraser01lomac] Timothy Fraser. LOMAC: MAC you can live with. In *Freenix Track: 2001 USENIX Annual Technical Conference*, Boston, Mass., 2001. UIC

Annotation: This is a very implementation oriented description of LOMAC, and appears to have a few changes relative to the '01 paper. Notably, trusted demons and does not talk about support for shared memory.

[fraser99lomac] Timothy Fraser. LOMAC–low water-mark mandatory access control for Linux. In *Proc. of the USENIX Security Symposium*, Washington D.C., 1999. UIC

Annotation: LOMAC uses low water-mark security integrity policy to divide root level privileges into two integrity levels, thus protecting the most important core from attack.

[gligor99twentyYears] Virgil D. Gligor. 20 years of operating system security. In *Proc. IEEE Symp. Security and Privacy*, 1999. UIC

[grimm97security] R. Grimm and B. Bershad. Security for extensible systems. In IEEE, editor, *The Sixth Workshop on Hot Topics in Operating Systems: May 5–6, 1997, Cape Cod, Massachusetts*, pages 62–66, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1997. IEEE Computer Society Press. UIC

Annotation: A very vague description of security in extensible OS. Seems like a very hard problem.

[grimm99providing] Robert Grimm and Brian N. Bershad. Providing policy-neutral and transparent access control in extensible systems. In *Secure Internet Programming*, pages 317–338, 1999. UIC

Annotation: This is a better paper. Most of the mechanism is from DTOS, with a separation of security manager (policy) and kernel-level enforcer, a security context mapped into a security ID, caching of results in the kernel. Has a two level security in that the core is protected in a different way from the extensible portion.

[hurwitz01abolishRoot] Carol Hurwitz and Scott McPeak. Abolish root daemons! http://www.cs.berkeley.edu/ ~smcpeak/cs261/paper.pdf, February 2001. UIC Public

[ioannidis02subOperatingSystem] Sotiris Ioannidis, Steven M. Bellovin, and Jonathan Smith. Sub-operating systems: A new approach to application security. In *SIGOPS European Workshop*, 2002. UIC Public

Abstract: Encapsulated code has become an increasingly common technique for extending the functionality of applications such as e?mail systems and web browsers. An unfortunate consequence has been the use of the same encapsulation techniques for malicious purposes, such as e?mailing viruses or active content embedded in HTML. We believe that the core problem is that encapsulated code must have a default security policy distinct from that of the application, rather than inheriting the application's privileges. Current operating systems are unable to protect their users from this kind of attack, since the hostile software is running with the user's privileges and permissions. Our approach is a novel protection mechanism we call the Sub?Operating System or SubOS. The SubOS tags all arriving content with unique non?removable sub?user IDs, under the presumption that the content may become active. If it does become active, the ID is used as the basis for creating a privilege environment, in a fashion analogous to setuid. This environment has a default security policy, which is typically quite restrictive. In this way, active code is prevented from inheriting the full privileges and permissions of the

user. We have implemented a SubOS prototype under OpenBSD and report on its use for supporting a mail system based on mh, as well as a simple web browser application.

Annotation: See also [**?**]

[jaeger98fineGrained] Trent Jaeger, Jochen Liedtke, and Nayeem Islam. Operating system protection for fine-grained programs. In *Proc. of the USENIX Security Symposium*. USENIX, 1998. UIC

Annotation: Describes an architecture which provides kernel-based support for a fine-grained programs (java) under certain circumstances. Capabilities based, and reasonably efficient (claiming 30% overhead).

[jaeger99download] Trent Jaeger, Atul Prakash, Jochen Liedtke, and Nayeem Islam. Flexible control of downloaded executable content. *ACM Transactions on Information and System Security (TISSEC)*, 2(2):177–228, 1999. UIC

Abstract: We present a security architecture that enables system and application a ccess control requirements to be enforced on applications composed from downloaded executable content. Downloaded executable content consists of messages downloaded from remote hosts that contain executables that run, upon receipt, on the downloading principal's machine. Unless restricted, this content can perform malicious actions, including accessing its downloading principal's private data and sending messages on this principal's behalf. Current security architectures for controlling downloaded executable content (e.g., JDK 1.2) enable specification of access control requirements for content based on its provider and identity. Since these access control requirements must cover every legal use of the class, they may include rights that are not necessary for a particular application of content. Therefore, using these systems, an application composed from downloaded executable content cannot enforce its access control requirements without the addition of application-specific security mechanisms. In this paper, we define an access control model with the following properties: (1) system administrators can define system access control requirements on applications and (2) application developers can use the same model to enforce application access control requirements without the need for ad hoc security mechanisms. This access control model uses features of role-based access control models to enable (1) specification of a single role that applies to multiple application instances; (2) selection of a content's access

rights based on the content's application and role in the application; (3) consistency maintained between application state and content access rights; and (4) control of role administration. We detail a system architecture that uses this access control model to implement secure collaborative applications. Lastly, we describe an implementation of this architecture, called the Lava security architecture.

[kamp00jails] Poul-Henning Kamp and Robert N. M. Watson. Jails: Confining the omnipotent root. In *SANE 2000*. NLUUG, 2000. UIC

Annotation: FreeBSD Project. Jails are an extension of chroot to not only limit filesystem access but also access to other processes and network resources. Each root application is run in a separate jail, with a separate IP address, and can only communicate with its descendents which forever share the same jail. Pretty cool, but all at the process level.

[kiriansky02shepherding] Vladimir Kiriansky, Derek Bruening, and Saman Amarasinghe. Secure execution via program shepherding. In USENIX, editor, *Proc. of the USENIX Security Symposium*, pages 191–206, Berkeley, CA, USA, 2002. USENIX. UIC Public

[loscocco01flexible] Peter Loscocco and Stephen Smalley. Integrating flexible support for security policies into the linux operating system. In *Proceedings of the FREENIX Track (FREENIX-01)*, pages 29–42, Berkeley, CA, 2001. The USENIX Association. UIC

Annotation: Based on the Fluke work, uses a kernel-based module and a very fine degree of acccess control to make a secure, extensible system according to the policy implemented in the module.

[loscocco01securityObjectives] Peter Loscocco and Stephen Smalley. Meeting critical security objectives with security-enhanced linux. In *Proceedings of the Ottawa Linux Symposium*, Berkeley, CA, 2001. The USENIX Association. UIC

[loscocco98inevitability] Peter Loscocco, Stephen D. Smalley, Patrick A. Muckelbauer, Ruth C. Taylor, S. Jeff Turner, and John F. Farrel. The inevitability of failure: The flawed assumption of security in modern computing environments. In *21st National Information System Security Conference*, pages 303–314, 1998. UIC

[olawsky96policyNeutral] D. Olawsky, T. Fine, E. Schneider, and R. Spencer. Developing and using a "policy neutral" access control policy. In *Proc. of the New Security Paradigms Workshop (NSPW)*, 1996. UIC

[peterson02containment] David S. Peterson, Matt Bishop, and Raju Pandey. A flexible containment mechanism for executing untrusted code. In USENIX, editor, *Proc. of the USENIX Security Symposium*, pages 207–225, Berkeley, CA, USA, 2002. USENIX. UIC Public

[provos02systemCallPolicies] N. Provos. Improving host security with system call policies. In *Proc. Usenix Security Symposium*, August 2002. UIC Public

[schroeder75secureMultics] Michael D. Schroeder. Engineering a security kernel for Multics. In *Proceedings of the fifth symposium on Operating systems principles*, pages 25–32, 1975. UIC

[shapiro00eros] Jonathan S. Shapiro and Samuel Weber. Verifying the EROS confinement mechanism. In *Proc. IEEE Symp. Security and Privacy*, pages 166–176, 2000. UIC

Abstract: Capability systems can be used to implement higher-level security policies including the property if a mechanism exists to ensure confinement. The implementation can be efficient if the "weak" access restriction described in this paper is introduced. In the course of developing EROS, a pure capability system, it became clear that verifying the correctness of the confinement mechanism was necessary in establishing the security of the operating system. This paper presents a verification of the EROS confinement mechanism with respect to a broad class of capability architectures (including EROS). We give a formal statement of the requirements, construct a model of the architecture's security policy and operational semantics, and show that architectures covered by this model enforce the confinement requirements if a small number of initial static checks on the confined subsystem are satisfied. The method used generalizes to any capability system.

Annotation: See `http://www.eros-os.org` for more info on EROS.

[shapiro99erosCapabilities] Jonathan S. Shapiro, Jonathan M. Smith, and David J. Farber. EROS: a fast capability system. In *Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP'99)*, pages 170–185, 1999. UIC

Annotation: Evolution of GNOSIS/KeyKOS [**?**] written in C++. capability-based with hardware monitor (contrary to e.g., amoeba; reason being mandatory access control). transparent persistence. See `http://www.eros-os.org/` for more info on EROS.

[smalley02seLinuxLsm] Stephen Smalley, Chris Vance, and Wayne Salamon. Implementing SELinux as a Linux security module. Report #01-043, NAI Labs, December 2001. Revised April 2002. UIC Public

Abstract: This technical report describes the implementation of the LSM-based SELinux security module. The report begins by providing an overview of LSM and a review of the SELinux basic concepts. It then provides a summary of how the LSM-based SELinux security module differs from the original SELinux kernel patch. Several aspects of the SELinux security module are then described, including its internal architecture, its initialization and exit code, its support for stacking with other security modules, and its approach for implementing the new SELinux system calls. The remainder of the report is then spent documenting the SELinux hook function implementations, organized into sections for each grouping of LSM hooks.

Annotation: See also [loscocco01flexible, spencer99flask] for more info on SELinux and its underpinning Flask.

[smalley02selinuxPolicy] Stephen Smalley. Configuring the SELinux policy. Report #02-007, NAI Labs, February 2002. Revised April 2002. UIC Public

Abstract: This technical report describes how to configure the SELinux security policy for the example security server. It begins by explaining concepts defined by the Flask architecture that are important to configuring the policy. The report then describes the security model implemented by the example security server. The policy language and the example policy configuration are then described. The report explains how the policy is built and applied to the system. Security-aware applications and their configurations are discussed. Finally, the report describes how to customize the policy for various purposes.

Annotation: See also SmaFra2001 [loscocco01securityObjectives, spencer99flask].

[spencer99flask] Ray Spencer, Stephen Smalley, Peter Loscocco, Mike Hibler, David Andersen, and Jay Lepreau. The Flask security architecture: System support for diverse security policies. In *Proc. of the USENIX Security Symposium*, 1999. UIC

Annotation: Describes a security architecture in which the (micro) kernel enforces security while the security policy server makes security decisions. Uninterpreted security decision (security contexts)

are described by fixed sized identifiers (security IDs), which are then transmitted to the security policy server.

[swift01nt] Michael M. Swift, Peter Brundrett, Cliff Van Dyke, Praerit Garg, Anne Hopkins, Shannon Chan, Mario Goertzel, and Gregory Jensenworth. Improving the granularity of access control in Windows NT. In *SACMAT*, pages 87–96, 2001. UIC

[walter75securityKernel] K. G. Walter, S. I. Schaen, W. F. Ogden, W. C. Rounds, D. G. Shumway, D. D. Schaeffer, K. J. Biba, F. T. Bradshaw, S. R. Ames, and J. M. Gilliganp. Structured specification of a security kernel. In *Proceedings of the international conference on Reliable software*, pages 285–293, 1975. UIC

[watson00trustedOs] Robert Watson. Introducing supporting infrastructure for trusted operating system support in FreeBSD. In *BSDCon 2000*, Monterey, CA, 2000. UIC

Annotation: This is from the TrustedBSD project at NAI, which adds Extended Attributes (customizable), and Posix.1e capabilities to FreeBSD.

[watson01trustedBsd] Robert Watson. TrustedBSD: Adding trusted operating system features to FreeBSD. In *USENIX Technical Conference*, Boston, MA, 2001. UIC

Annotation: This paper describes more of the philosophy, and is less detailed on features than the 2000 bsdcon paper.

[wright02lsm] Chris Wright, Crispin Cowan, Stephen Smalley, James Morris, and Greg Kroah-Hartman. Linux Security Modules: General security support for the Linux Kernel. In *Proc. of the USENIX Security Symposium*, San Francisco, Ca., 2002. UIC

Annotation: Scheme to add in additional checks–i.e. restrictions–(beyond unix DACs) to Linux. Adds hooks so that new kernel protection models can be built without making changes (patches) all over the linux kernel. Also supports capabilities, which are privileges (ie additive).

13 Distributed protection

[abadi93calculus] Martín Abadi, Michael Burrows, Butler Lampson, and Gordon Plotkin. A calculus for access control in distributed systems. *ACM Transactions on Computing Systems (TOCS)*, 15(4):706–734, 1993. UIC

[belani98crisis] Eshwar Belani, Amin Vahdat, Thomas Anderson, and Michael Dahlin. The CRISIS wide area security architecture. In USENIX, editor, *Proc. of the USENIX Security Symposium*, pages 15–30, Berkeley, CA, USA, 1998. USENIX. UIC Public

Annotation: Describes a vary large, distributed infrastructure for authentication, which is in essence a very souped up version of Kerberos.

[blaze96decentralizedTrust] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proc. IEEE Symp. Security and Privacy*, 1996. UIC

Annotation: Verification of identities is distinct from assessing the trust accorded to them. Existing systems tend to combine the two into certificate management. PolicyMaker is a system for evaulating trust. It is essentially a rule-based system. The rules or "assertions" are of the form: *source* ASSERTS *AuthorityStruct* WHERE *filter* where *source* is either a key-id or the keyword POLICY. Assertions install filters which can check conditions under which trust from *source* is transferred to the key-ids that satisfy AuthorityStruct. *AuthorityStruct* is a function on key-ids (it can be a simple key-id or a program that takes key-ids as parameters) and *filter* is program (described below). Queries to PolicyMaker are of the form: *key-id list* REQUESTS *ActionString ActionString* is a string constructed by the application. It is not interpreted by the PolicyMaker except that the filters of assertions will take action strings as input and returns a boolean status (filters may also append annotations to the action strings). Assertions are either local (called policies, and having the POLICY keyword as the *source*) or signed binding of an authority structure to a filter (called certificates). Input to filters consist of the action string and the "environment" consisting of information like the current time and the calling application's name. PolicyMaker does not do any crypto – signature verification etc. are assumed to be done elsewhere.

[desmedt93keyboard] Y. Desmedt. Computer security by redefining what a computer is. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 160–166, Washington - Brussels - Tokyo, 1993. IEEE. UIC

[gilmore99secureRemoteWeb] Christian Gilmore, David Kormann, and Aviel D. Rubin. Secure remote access to an internal web server. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, San Diego, CA, 1999. Internet Society. UIC Public

Annotation: How to build a (secure) web-proxy to allow external users to access internal web-pages. proxy is split in two halves, one before and one behind the firewall. proxy authentication to user with SSL. Initial user authentication with S/KEY or OPIE for, authentication of requests with URLs containing a MAC (all URLs are rewritten by the proxy, so this goes easily).

[gligor01accessControlPolicies] Virgil Gligor, Himanshu Khurana, Radostina Koleva, Vijay G. Bharadwaj, and John Baras. On the negotiation of access control policies. In *Cambridge Security Protocols Workshop*, Cambridge, England, 2001. UIC

[goldberg96secure] Ian Goldberg, David Wagner, Randi Thomas, and Eric A. Brewer. A secure environment for untrusted helper applications (confining the wily hacker). In *Proc. of the USENIX Security Symposium*, San Jose, Ca., 1996. UIC

Annotation: Provides a mechanism for sandboxing helper applications using system call tracing and interception facilities. Requires a per process configuration file (although doesn't really tell why one is needed per process). System is called Janus.

[greenwald96distributedResourceModel] Steven J. Greenwald. A new security policy for distributed resource management and access control. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 74–86. ACM Press, 1997. UIC

[kohl91kerberosEvolution] John T. Kohl, B. Clifford Neuman, and Theodore Y. Ts'o. The evolution of the kerberos authentication service. In *Proceedings of the Spring EurOpen'91 Conference*, Tromso, 1991. UIC

Abstract: Reprinted in F. M. T. Brazier, D. Johansen, eds.

[lampson91authentication] Butler Lampson, Martín Abadi, Michael Burrows, and Edward Wobber. Authentication in distributed systems: Theory and practice. *ACM Transactions on Computer Systems*, pages 165–182, 1991. UIC

Annotation: Cite the journal version [lampson92authentication].

[lampson92authentication] Butler Lampson, Martín Abadi, Michael Burrows, and Edward Wobber. Authentication in distributed systems: Theory and practice. *ACM Transactions on Computer Systems*, 10(4):265–310, 1992. UIC

Annotation: A very nice paper on distributed systems security. Describes: revocation (and why its needed); caches with timeouts (as a means of trading of efficiency vs. security); Principals, Objects, and Reference Monitors; Trusted Programs; Roles and Groups.

[minsky98heterogeneous] Naftaly H. Minsky and Victoria Ungureanu. Unified support for heterogeneous security policies in distributed systems. In USENIX, editor, *Proc. of the USENIX Security Symposium*, pages 131–142, Berkeley, CA, USA, 1998. USENIX. UIC

[na00roleDelegation] Sang Yeob Na and SuhHyun Cheon. Role delegation in role-based access control. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 39–46. ACM Press, 2000. UIC

[neuman88unknownAuthorization] B. Clifford Neuman and Jennifer G. Steiner. Authentication of unknown entities on an insecure network of untrusted workstations. In *Proceedings of the Usenix Workshop on Workstation Security*, 1988. UIC

[orman97positiveFeedback] Hilarie Orman and Richard Schroeppel. Positive feedback and the madness of crowds. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 134–138. ACM Press, 1997. UIC

[perlman99securePassword] Radia Perlman and Charlie Kaufman. Secure password-based protocol for downloading a private key. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, San Diego, CA, 1999. Internet Society. UIC Public

Annotation: Describe how to use strong protocols to access remotely stored key pairs. Their protocols are based on EKE and SPEKE and they primarily optimize (simplify) them for their specific need: Their minimal two-flow protocol is A-¿B: $E_{h_1(pwd)}(g^x)$ B-¿A: $E_{h_1(pwd)}(g^y), E_{g^{xy}}(Y)$ where B stores $h_1(pwd)$, y, $E_{h_1(pwd)}(g^y)$ (ie, B can reuse the secret exponent y) and $Y = E_{h_2(pwd)}$(A's secret-key). See [**?**] for (unjustified) critic.

[ramm95delegation] Karl Ramm and Michael Grubb. Exu: A system for secure delegation of authority on an insecure network. In *Proceedings of the Ninth Systems Administration Conference*, pages 89–94, Berkeley, 1995. Usenix Association. UIC

[reiter96distributedTrust] Michael Reiter. Distributing trust with the rampart toolkit. *Communications of the ACM (CACM)*, 39(4):71–74, 1996. UIC

[reiter99authenticationMetric] Michael K. Reiter and Stuart G. Stubblebine. Authentication metric analysis and design. *ACM Transactions on Information and System Security (TISSEC)*, 2(2):138–158, 1999. UIC Public

[rivest96sdsi] Ronald L. Rivest and Butler Lampson. SDSI — a simple distributed security infrastucture. Technical report, MIT, April 1996. UIC

Annotation: Proposes a very nice idea for infrastructure based on using public keys, which combines together domain-based security (ala NIS) with universal identifiers which are public keys. Also describes how to build groups and many other relationships.

[samar96unifiedPamLogin] Vipin Samar. Unified login with Pluggable Authentication Modules (PAM). In Clifford Neuman, editor, *Proc. ACM Conference on Computer and Communications Security (CCS)*, pages 1–10. ACM Press, 1996. UIC

Annotation: API and SPI for pluggable authentication modules. decomposition in an authentication, account, session and password management part. stacking and password-reuse/mapping allows for single sign on of multiple parallel authentication schemes. separate configuration(-file). user-interaction is controlled by caller of API through conversation structure allowing therefore for adaption to whaterever UI/GUI used. implementation for UNIX, ONC+, DCE, Kerberos, S/Key, rlogin, RSA. PAM API is selected by CDE vendors as official API. see also `http://www.sun.com/software/solaris/pam/`

[sandhu98dacRbac] Ravi Sandhu and Qamar Munawer. Control principles and role hierarchies. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 47–54, 1998. UIC

[satyanarayanan89integrating] M. Satyanarayanan. Integrating security in a large distributed system. *ACM Transactions on Computer Systems*, 7(3):247–280, 1989. UIC

Annotation: Describes security in the Andrew File System. This is an early paper on distributed security. Although there are many limitations to the security it provides, it represented a serious step forward in the 80s. Also of interest describes permissions on directories, and their mapping into unix files.

[shands00enclaves] Deborah Shands, Richard Yee, Jay Jacobs, and E. John Sebes. Secure virtual enclaves: Supporting coalition use of distributed application technology. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, San Diego, CA, 2000. Internet Society. UIC

[steiner88kerberos] Jennifer G. Steiner, B. Clifford Neuman, and J. I. Schiller. Kerberos: An authentication service for open network systems. In *Winter 1988 USENIX Conference*, pages 191–201, Dallas, TX, 1988. UIC

Abstract: In an open network computing environment, a workstation cannot be trusted to identify its users correctly to network services. Kerberos provides an alternative approach whereby a trusted third-party authentication service is used to verify users' identities. This paper gives an overview of the Kerberos authentication model as implemented for MIT's Project Athena. It describes the protocols used by clients, servers, and Kerberos to achieve authentication. It also describes the management and replication of the database required. The views of Kerberos as seen by the user, programmer, and administrator are described. Finally, the role of Kerberos in the larger Athena picture is given, along with a list of applications that presently use Kerberos for user authentica tion. We describe the addition of Kerberos authentication to the Sun Network File System as a case study for integrating Kerberos with an existing application.

Annotation: Describes Kerberos v4 ticket granting authority and algorithms.

[weiss85yellowPages] P. Weiss. *Yellow Pages Protocol Specification*. Sun Microsystems, Inc., 1985. UIC

[wobber93authentication] Edward Wobber, Martín Abadi, Michael Burrows, and Butler Lampson. Authentication in the Taos operating system. In *ACM Symposium on Operating Systems Principles*, pages 256–269, 1993. UIC

Annotation: Cite the journal paper [wobber94authentication].

[wobber94authentication] Edward Wobber, Martín Abadi, Michael Burrows, and Butler Lampson. Authentication in the Taos operating system. *ACM Transactions on Computer Systems*, 12(1):3–32, 1994. UIC

[woo93distributedAuthorization] Thomas Y.C. Woo and Simon S. Lam. A framework for distributed authorization. In *Proc. ACM Conference on Computer and Communications Security (CCS)*. ACM, 1993. UIC

14 Certificates

[aura98delegationNetworks] T. Aura. On the structure of delegation networks. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW)*, pages 14–26, Washington - Brussels - Tokyo, 1998. IEEE. UIC

[borisov02delegationFramework] Nikita Borisov and Eric Brewer. Active certificates: A framework for delegation. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*. Internet Society, 2002. UIC

[ellison00pkiRisks] Carl Ellison and Bruce Schneier. Ten risks of PKI: What you're not being told about Public Key Infrastructure. *Computer Security Journal*, 16(1):1–7, 2000. UIC

> **Abstract:** Public-key infrastructure has been oversold as the answer to many network security problems. We discuss the problems that PKI doesn't solve, and that PKI vendors don't like to mention.
>
> **Annotation:** risks: - Who do we trust, and for what? (who gave authority to CA over what) - Who is using my key? (physical/logical protection of key) - How secure is the verifying computer? (integrity root keys and SW in general) - Which John Robinson is he? (does a distinguished name mean anything?) - Is the CA an authority? (SSL certifications contain DNS name but not issued by authority of DNS names ..) - Is the user part of the security design? (In browser user cannot (reasonably) be expected to make correct decision; SSL-closed key/lock in browser not sufficient!) - Was it one CA or a CA plus a Registration Authority? (RA+separate CA model is weaker than all-in-one-CA. [really? physical protection ..]) - How did the CA identify the certificate holder (unsecure basis of identification at registration time (e.g., through credit bureaus ..)) - How secure are the certificate practices (are CPS designed with solid security reasons?) - Why are we using the CA process anyway?

[grigg06pkiHarmful] Ian Grigg. Pki considered harmful. `http://iang.org/ssl/pki_considered_harmful.html`, 2006. UIC

[josang99algebra] A. Josang. An algebra for assessing trust in certification chains. In J.Kochmar, editor, *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, 1999. UIC

[khurana00chainedRevocation] Himanshu Khurana and Virgil D. Gligor. Review and revocation of access privileges distributed with PKI certificates. In *Proc. of the Security Protocols Workshop*, volume 2133 of *Lecture Notes in Computer Science*, Cambridge, UK, 2000. Springer-Verlag. UIC

[kortesniemi00delegation] Yki Kortesniemi, Tero Hasu, and Jonna Särs. A revocation, validation and authentication protocol for SPKI based delegation systems. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, pages 85–101. Internet Society, 2000. UIC

Annotation: Argues that for authorization certificates online checking (e.g., CRLs don't work in distributed settings and for some resource management usage-tracking is needed) is necessary and propose necessary changes to SPKI (deprecate CRLs, online test query format, add 'renew', introduce 'limit' online test).

[micali96efficientRevocation] Silvio Micali. Efficient certificate revocation. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996. UIC

[naor98certificateRevocation] Moni Naor and Kobbi Nissim. Certificate revocation and certificate update. In *Proc. of the USENIX Security Symposium*, pages 217–228. Usenix Association, 1998. UIC

[nochta02revocation] Zoltn Nochta, Peter Ebinger, and Sebastian Abeck. Pamina: A certificate based privilege management system. In *Proc. IEEE Symp. Security and Privacy*, 2002. UIC

Annotation: Discusses revocation.

15 Network

[blaze01ipsecTrust] M. Blaze, J. Ioannidis, and A. D. Keromytis. Trust management for IPsec. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, San Diego, CA, 2001. Internet Society. UIC

Abstract: IPsec is the standard suite of protocols for network-layer confidentiality and authentication of Internet traffic. The IPsec protocols, however, do not address the policies for how protected traffic should be handled at security endpoints. This paper introduces an efficient policy management scheme for IPsec, based on the principles of trust management. A compliance check is added to the IPsec architecture that tests packet filters proposed when new security associations are created for conformance with the local security policy, based on credentials presented by the peer host. Security policies and credentials can be quite sophisticated (and specified in the trust-management language), while still allowing very efficient packet-filtering for the actual IPsec traffic. We present a practical,

portable implementation of this design, based on the KeyNote trust-management language, that works with a variety of Unix-based IPsec implementations.

Annotation: For journal version see [blaze02ipsecTrust]

[blaze02ipsecTrust] Matt Blaze, John Ioannidis, and Angelos D. Keromytis. Trust management for IPsec. *ACM Transactions on Information and System Security (TISSEC)*, 5(2):95–118, 2002. UIC Public

Abstract: IPsec is the standard suite of protocols for network-layer confidentiality and authentication of Internet traffic. The IPsec protocols, however, do not address the policies for how protected traffic should be handled at security end points. This article introduces an efficient policy management scheme for IPsec, based on the principles of trust management. A compliance check is added to the IPsec architecture that tests packet filters proposed when new security associations are created for conformance with the local security policy, based on credentials presented by the peer host. Security policies and credentials can be quite sophisticated (and specified in the trust-management language), while still allowing very efficient packet-filtering for the actual IPsec traffic. We present a practical portable implementation of this design, based on the KeyNote trust-management language, that works with a variety of UNIX-based IPsec implementations. Finally, we discuss some applications of the enhanced IPsec architecture.

Annotation: Journal version of [**?**]. See also closely related papers on distributed firewall papers [**?**, **?**] and KeyNote [**?**]

[blaze99networkLayer] John Ioannidis Matt Blaze and Angelos D. Keromytis. Trust management and network-layer security protocols. In *Cambridge Protocols Workshop*, Cambridge, 1999. UIC

[casella95openNetwork] Karen A. Casella. Security administration in an open networking environment. In *Proceedings of the Ninth Systems Administration Conference*, pages 67–74, Berkeley, 1995. Usenix Association. UIC

[choo99vaultedVpn] Tse-Huong Choo. Vaulted VPN: Compartmented virtual private networks on trusted operating systems. In *Proc. of the USENIX Security Symposium*, pages 35–46, Berkeley, CA, 1999. Usenix Association. UIC

Annotation: Describes a process-level design for augmenting the protocol stack with a process per sensitivity level to support Bell-LaPadula security model in HP/UX 10.24. Claims some security advantage in moving the protocol stack out of the kernel, but this is not persuasive.

[fu01webAuthentication] Kevin Fu, Emil Sit, Kendra Smith, and Nick Feamster. The dos and don'ts of client authentication on the Web. In USENIX, editor, *Proc. of the USENIX Security Symposium*, pages 251–270, Berkeley, CA, USA, 2001. USENIX. UIC

[helme99offlineDelegation] Arne Helme and Tage Stabell-Kulø. Offline delegation. In *Proc. of the USENIX Security Symposium*, pages 25–33. USENIX, 1999. UIC Public

Abstract: This article describes mechanisms for offline delegation of access rights to files maintained by a distributed "File Repository". The mechanisms are designed for a target environment where personal machines are used at times when critical services, such as authentication and authorization services, are not accessible. We demonstrate how valid delegation credentials can be transferred verbally without the use of shared secrets. Our main result shows that delegation of access rights can be accomplished in a system that uses public-key encryption for secrecy and integrity, without forcing the user to rely on a trusted third party, and without requiring connection to the infrastructure. The implementation runs on a contemporary Personal Digital Assistant (PDA); the performance is satisfactory.

[hill96dissemination] Brian C. Hill. Priv: Secure and flexible privileged access dissemination. In USENIX, editor, *Proceedings of the Tenth Systems Administration Conference (LISA X), September 29–October 4, 1996, Chicago, IL, USA*, pages 1–8, Berkeley, CA, USA, 1996. USENIX. UIC

[paxson99bro] Vern Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks*, 31(23-24):2435–2463, 1999. UIC

Annotation: Describes a network monitoring program which captures interesting network events and then analyzes them for security problems. Also logs. Interesting discussion of the types of attacks, mostly about how intent is disguised at the packet level and how their system deals with that. Emphasis is on detection, not prevention as is done with firewalls.

[raadt99cryptographyOpenbsd] Theo de Raadt, Niklas Hallqvist, Artur Grabowski, Angelos D. Keromytis, and Niels Provos. Cryptography in OpenBSD: An overview.

In *Proceedings of the FREENIX Track (FREENIX-99)*, pages 93–102, Berkeley, CA, 1999. USENIX Association. UIC

[tatu96ssh] Tatu Ylonen. SSH—secure login connections over the Internet. In *Proc. of the USENIX Security Symposium*, pages 37–42, San Jose, California, 1996. UIC

16 Firewall

[bellovin99distributedFirewalls] Steven M. Bellovin. Distributed firewalls. *;login: magazine*, pages 37–39, 1999. special issue on security. UIC

Annotation: This is the justification paper. A much more detailed and well developed architecture is in the implementation paper, see [ioannidis00distributedFirewalls]

[cheswick94firewalls] W. Cheswick and S. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley, 1994. UIC

[firmato99firewall] A. Firmato, Yair Bartal, Alain Mayer, Kobbi Nissim, and Avishai Wool. A Novell firewall management toolkit. In *Proc. IEEE Symp. Security and Privacy*, 1999. UIC

[ioannidis00distributedFirewalls] Sotiris Ioannidis, Angelos D. Keromytis, Steve M. Bellovin, and Jonathan M. Smith. Implementing a distributed firewall. In *Proceedings of the 7th ACM conference on Computer and Communications Security*, pages 190–199. ACM Press, 2000. UIC

17 Distributed Denial of Service (DDoS)

[chang02ddosTutorial] Rocky K. C. Chang. Defending against flooding-based, distributed denial-of-service attacks: A tutorial. *IEEE Communications Magazine*, 40(10):42–51, 2002. UIC

Annotation: This paper describes several "flooding" type DDOS attacks either consuming victim memory resources (SYN attacks) or victim bandwidth resources (bogus packets resulting in RST). The attacks can either be direct or reflector based, causing routers or hosts which are neither victims or attackers to participate in the attacks (eg. by issuing replies to bogus IP addresses). Finally, defenses can either be at the source (bogus IP), at the destination (DDOS attack) or in the middle. The last is the most complex.

[ioannidis02ddos] John Ioannidis and Steven M. Bellovin. Implementing pushback: Router-based defense against DDoS attacks. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, San Diego, California, 2002. The Internet Society. UIC

[moore01inferringDos] David Moore, Geoffrey Voelker, and Stefan Savage. Inferring Internet denial of service activity. In *Proc. of the USENIX Security Symposium*, pages 9–22. USENIX, 2001. UIC Public

Annotation: Uses "backscattering" to infer DDoS. The assumption is that attack packets are spoofed and randomly (uniformly) generated. Hence, by oberving a fraction of the address space, it is possible to tell what is happening in the entire internet. The authors used 1/256 of the address space.

[snoeren02ipTraceback] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Beverly Schwartz, Stephen T. Kent, and W. Timothy Strayer. Single-packet IP traceback. *IEEE/ACM Transactions on Networking (TON)*, 10(6):721–734, 2002. UIC

Annotation: Describes a means of tracing individual IP packets by means of hashing techniques. The scheme masks out most fields that change as a packet is routed, as well as the data field and then computes a hash code which ultimately reduces to a single bit in a bit vector (called a Bloom hash). Special care must be taken to account for various ways that packets are mangled in the network such as NAT.

[sung02defendingDdos] Minho Sung and Jun Xu. IP traceback-based intelligent packet filtering: A novel technique for defending against Internet DDoS attacks. In *Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 302–311. IEEE Computer Society, 2002. UIC

Annotation: Proposes dropping packets from paths along which attack is occurring. This frees up bandwidth at a "perimeter", but also drops legitimate traffic.

18 Separation of Duty

[ahn99rsl] Gail-Joon Ahn and Ravi Sandhu. The RSL99 language for role-based separation of duty constraints. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 43–54. ACM Press, 1999. UIC

Annotation: Discussion focusses on needs of Separation of Duty (SoD) within role-based access controls, not the inherent needs of SoD.

[ferraiolo95rbacFeatures] David Ferraiolo, Janet Cugini, and Rick Kuhn. Role based access control (RBAC): Features and motivations. In *Annual Computer Security Applications Conference*. IEEE Computer Society Press, 1995. UIC

Annotation: Describes an access control method based on the roles a user has within an organization. In terms of separation of duty, describes operational separation of duty as "A role can be associated with an operation of a business function only if the role is an authorized role for the subject and the role had not be assigned previously to all of the other operations."

[gligor98separation] V. D. Gligor, S. I. Gavrila, and D. Ferraiolo. On the formal definition of separation-of-duty policies and their composition. In *Proc. IEEE Symp. Security and Privacy*, pages 172–185, 1998. UIC

[kuhn97separation] D. Richard Kuhn. Mutual exclusion of roles as a means of implementing separation of duty in role-based access control systems. In *Proc. of the ACM Workshop on Role-Based Access Controls (RBAC)*, pages 23–30. ACM Press, 1997. UIC

[nash90conundrums] M. J. Nash and K. R. Poland. Some conundrums concerning separation of duty. In *Proc. IEEE Symp. Security and Privacy*, pages 201–207, 1990. UIC

Abstract: This paper examines some questions concerning commercial computer security integrity policies. We give an example of a dynamic separation of duty policy which cannot be implemented by TCSEC based mechanisms alone, yet occurs in the real commercial world, and can be implemented efficiently in practice. We examine and describe a commercial computer security product in wide use for ensuring the integrity of

financial transactions, show that it implements a well defined and sensible integrity policy that includes separation of duty, yet fails to meet either the TCSEC criteria or the Clark and Wilson rules.

Annotation: Introduces object-base separation of duty in which each object can be dynamically assigned its own separation of duty. Suggests a mechanism by which individual object track what users have accessed them.

[sandhu88separation] Ravi Sandhu. Transaction control expressions for separation of duties. In *Fourth Aerospace Security Applications Conference*, pages 282–286, 1988. UIC

Annotation: Interesting discussion of various Separation of Duty policies. Policies can have requirements that different individuals or the same individuals perform various steps. Policy can be differentiated by, for example, amount of money involved. Policy can require less individuals to approve if the individuals are more trusted (higher rank). Discusses anomalies which can occur and how they might be dealt with.

[simon97separation] Richard T. Simon and Mary Ellen Zurko. Separation of duty in role-based environments. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW)*, pages 183–194. IEEE, 1997. UIC

19 Internet RFCs

[rfc1631] K. Egevang and P. Francis. RFC 1631: The IP network address translator (NAT), May 1994. Status: INFORMATIONAL. UIC Public

Annotation: Network Address Translation is a method of sharing an IP address among several machines by rewriting the network address as it goes through a NAT router.

[rfc1825] R. Atkinson. RFC 1825: Security architecture for the Internet Protocol, August 1995. Obsoleted by RFC2401 [rfc2401]. Status: PROPOSED STANDARD. UIC Public

[rfc1826] R. Atkinson. RFC 1826: IP authentication header, August 1995. Obsoleted by RFC2402 [rfc2402]. Status: PROPOSED STANDARD. UIC Public

[rfc1827] R. Atkinson. RFC 1827: IP encapsulating security payload (ESP), August 1995. Obsoleted by RFC2406 [rfc2406]. Status: PROPOSED STANDARD. UIC Public

[rfc2246] T. Dierks and C. Allen. RFC 2246: The TLS protocol version 1, January 1999. Status: PROPOSED STANDARD. UIC Public

[rfc2367] D. McDonald, C. Metz, and B. Phan. RFC 2367: PF_KEY key management API, version 2, July 1998. Status: INFORMATIONAL. UIC Public

[rfc2401] S. Kent and R. Atkinson. RFC 2401: Security architecture for the Internet Protocol, November 1998. Status: PROPOSED STANDARD. UIC Public

[rfc2402] S. Kent and R. Atkinson. RFC 2402: IP authentication header, November 1998. Status: PROPOSED STANDARD. UIC Public

[rfc2406] S. Kent and R. Atkinson. RFC 2406: IP Encapsulating Security Payload (ESP), November 1998. Status: PROPOSED STANDARD. UIC Public

[rfc2408] D. Maughan, M. Schertler, M. Schneider, and J. Turner. RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP), November 1998. Status: PROPOSED STANDARD. UIC Public

[rfc2409] D. Harkins and D. Carrel. RFC 2409: The Internet Key Exchange (IKE), November 1998. Status: PROPOSED STANDARD. UIC Public

[rfc2459] R. Howsley, W. Ford, W. Polk, and D. Solo. RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile, January 1999. UIC Public

[rfc2693] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian Thomas, and Tatu Ylonen. RFC 2693: SPKI certificate theory, September 1999. UIC Public

[rfc2704] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. RFC 2704: The KeyNote Trust-Management System Version 2, September 1999. UIC Public

[rfc791] J. Postel. RFC 791: Internet Protocol, September 1981. Obsoletes RFC076. See also STD0005. Status: STANDARD. UIC Public

20 Cryptography

[anderson94whyCryptosystems] Ross Anderson. Why cryptosystems fail. *Communications of the ACM (CACM)*, 37(11):32–41, 1994. UIC Public

[bellovin96problem] Steven M. Bellovin. Problem areas for the IP security protocols. In *Proc. of the USENIX Security Symposium*, San Jose, California, 1996. UIC

Annotation: A rather tersely written paper which describes many attacks which can occur on IPsec when a) a single encrypted computer to computer communication channel is used, and b) users can insert arbitrary packets into the communication.

[bellovin98cryptography] Steven M. Bellovin. Cryptography and the Internet. *Lecture Notes in Computer Science*, 1462:46–??, 1998. UIC Public

[ellison96identity] Carl Ellison. Establishing identity without certification authorities. In USENIX, editor, *Proceedings of the sixth annual USENIX Security Symposium, focusing on applications of cryptography*, pages 67–76, San Jose, California, 1996. USENIX. UIC Public

[gutmann02cryptoLessons] Peter Gutmann. Lessons learned in implementing and deploying crypto software. In USENIX, editor, *Proc. of the USENIX Security Symposium*, pages 315–325, Berkeley, CA, USA, 2002. USENIX. UIC Public

[rivest78rsa] Ronald Rivest, Adi Shamir, and L. Adleman. On digital signatures and public key cryptosystems. *Communications of the ACM (CACM)*, 21:120–126, 1978. UIC

[schneier97whyCrypto] Bruce Schneier. Why cryptography is harder than it looks. Technical report, Counterpane Systems, 101 East Minnehaha Parkway, Minneapolis, MN 55419, 1997. UIC

[schneier98secureLogs] Bruce Schneier and John Kelsey. Cryptographic support for secure logs on untrusted machines. In *Proc. of the USENIX Security Symposium*, pages 53–62. USENIX, 1998. UIC Public

Abstract: In many real-world applications, sensitive information must be kept in log files on an untrusted machine. In the event that an attacker captures this machine, we would like to guarantee that he will gain little or no information from the log files and to limit his ability to corrupt the log files. We describe a computationally cheap method for making all log entries generated prior to the logging machine's compromise impossible for the attacker to read, and also impossible to undetectably modify or destroy.

Annotation: Lacks security analysis and is unlikely to work under the general assumption " one-way hash" due to the partial information leaked about pre-images. However, using MD5/SHA-1 as proposed as implementation and assuming some pseudo-random function (or maybe even random-oracle?) behaviour probably ok. See [**?**] for a slightly more general and a more systematic approach security-wise (although also only partially analyzed). See also their 1997 Technical Report (`http://citeseer.nj.nec.com/bellare97forward.html`, `ftp://www.cs.ucsd.edu/pub/bsy/pub/fi.ps`)

21 Programming Languages

[gong98jdk12] L. Gong and R. Schemers. Implementing protection domains in the Java Development Kit 1.2. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, pages 125–134, San Diego, CA, 1998. UIC Public

[myers00protecting] Andrew C. Myers and Barbara Liskov. Protecting privacy using the decentralized label model. *Software Engineering and Methodology*, 9(4):410–442, 2000. UIC

[myers97decentralizedModel] Andrew C. Myers and Barbara Liskov. A decentralized model for information flow control. In *Proceedings of the sixteenth ACM Symposium on Operating System Principles*, pages 129–142, 1997. Appeared in ACM Operating Systems Review volume 31, number 5. UIC

Abstract: This paper presents a new model for controlling information flow in systems with mutual distrust and decentralized authority. The model allows users to share information with distrusted code (e.g, downloaded applets), yet still control how that code disseminates the shared information to others. The model improves on existing multilevel security models by allowing users to declassify information in a decentralized way, and by improving support for fine-grained data sharing. The paper also shows how static program analysis can be used to certify proper information flows in this model and to avoid most run-time information flow.

Annotation: Journal version [**?**]. Information leakage across channels, covert or otherwise, in a programming language (called Jif) based

on Java. Checks the propagation of information in a programming language, so that the containment is more specific than in Lampson.

[wallach97extensibleJavaSecurity] Dan S. Wallach, Dirk Balfanz, Drew Dean, and Edward W. Felten. Extensible security architectures for Java. In *16th Symposium on Operating Systems Principles*, pages 116–128, 1997. UIC Public

22 Miscellaneous

[anderson01informationSecurityHard] Ross Anderson. Why information security it hard: An economic perspective. In *17th Annual Computer Security Applications Conference*, 2001. UIC

Annotation: Describes why it is not always in organizations best interest to provide high securities. Examples include banks, governments, and software companies.

[bonatti00composingPolicies] Piero A. Bonatti, Sabrina De Capitani di Vimercati, and Pierangela Samarati. A modular approach to composing access control policies. In Sushil Jajodia and Pierangela Samarati, editors, *Proc. ACM Conference on Computer and Communications Security (CCS)*, pages 164–173, N.Y., 2000. ACM Press. UIC

[ganger01security] Gregory R. Ganger and David Nagle. Better security via smarter devices. In *HotOS-VIII (IEEE Workshop on Hot Topics in Operating Systems*, 2001. UIC

[meadows93taxonomy] Catherine Meadows. An outline of a taxonomy of computer security research and development. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 33–35. IEEE, 1993. UIC

[rooker93referenceMonitor] T. Rooker. The reference monitor: An idea whose time has come. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 192–197. IEEE, 1993. UIC

[sterne94redraw] Dan Sterne, Glen Benson, and Homayoon Tajalli. Redrawing the security perimeter of a trusted system. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW)*, pages 162–177. IEEE, 1994. UIC

[venema96murphys] Wietse Venema. Murphy's law and computer security. In *Proc. of the USENIX Security Symposium*, 1996. UIC

Annotation: Apocryphal stories of what can go wrong. Describes lessons from computer security including: Don't use easily guessable "random" number, beware of malicious data, don't put secrets in user memory, watch out in depending on other programs (eg. they may have shell escapes).

[williams97sick] Jeff Williams. Just sick about security. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 139–147. ACM Press, 1997. UIC

23 Attacks

[berghel01codeRed] Hal Berghel. Digital village: The Code Red worm. *Communications of the ACM (CACM)*, 44(12):15–19, 2001. UIC

[seely88worm] Donn Seely. A tour of the worm. Technical report, Department of Computer Science, University of Utah, 1988. UIC Public

Abstract: On the evening of November 2, 1988, a self-replicating program was released upon the Internet 1 . This program (a worm) invaded VAX and Sun-3 computers running versions of Berkeley UNIX, and used their resources to attack still more computers 2 . Within the space of hours this program had spread across the U.S., infecting hundreds or thousands of computers and making many of them unusable due to the burden of its activity. This paper provides a chronology for the outbreak and presents a detailed description of the internals of the worm, based on a C version produced by decompiling.

Annotation: A description of how Robert Morris's worm worked. It used the following techniques of attack (1) Network discovery techniques through network tools and system configuration files (2) Used a buffer overrun vulnerability in finger to download vax binary and overwrite the return address on the stack, tricking finger into invoking a shell (3) password guessing attack (4) sendmail executables which were compiled with a debug flag would execute a shell upon request.

It used the following defenses: (1) respawing, periodically sleeping, and rewriting argv to fool ps (2) encoding string which needed to be xored with x81; (3) reducing debug info to make it less tracable.

[thompson84turing] Ken Thompson. Reflections on trusting trust. *Communications of the ACM (CACM)*, 27(8):761–763, 1984. UIC

Annotation: This is a small article on the event of Ken Thompson's turing award in which he reflects on the difficulty of detecting security flaws, in this particular case a trojan horse. This recounts an alleged exploit in which Thompson, who was preparing a tape for NSA created a trojan horse which was undetectable in the source code.

24 Unsorted

[carney98adaptive] Michael Carney and Brian Loe. A comparison of methods for implementing adaptive security policies. In *Proc. of the USENIX Security Symposium*, pages 1–14, 1998. UIC

Annotation: Describes several methods for changing security policy in an external security monitor. Good examples of why this is desirable: time sensitive material, time restrictions on when policies are enforced, reactions to threats, and role based security mechanisms.

[dietrich00shaft] Sven Dietrich, Neil Long, and David Dittrich. Analyzing distributed denial of service tools: The shaft case. In *Proceedings of the Fourteenth Systems Administration Conference (LISA-00)*, pages 329–340, Berkeley, CA, 2000. The USENIX Association. UIC

[needham94denial] Roger M. Needham. Denial of service: an example. *Communications of the ACM (CACM)*, 37(11):42–46, 1994. UIC

[wedde01modularAuthorization] Horst F. Wedde and Mario Lischka. Modular authorization. In *Seventh ACM Symposium on Access Control Models and Technologies*, pages 97–108, 2001. UIC

[wu98secureRemotePassword] Thomas Wu. The secure remote password protocol. In *Proc. of the Symp. on Network and Distributed Systems Security (NDSS)*, pages 97–111, San Diego, California, 1998. Internet Society. UIC Public

Annotation: This is a variation of password-authenticated Diffie-Hellman (see [BelMer92]). While a general construction is given the only instantiation presented is a slight variation of B-EKE [Jablon97]

which safes an exponentiation on each side by combining the "proof-of-knowledge" of the password-derived (secret !) "public key" g^pwd in the key generation itself. Find below "optimal" version for two-way authentication.
init: user chooses password pwd, salt s and computes $x = h(pwd, s), v = g^x$ B stores s, v
A: choose a random, $A = g^a$
$A- > B$: "$Alice''$, A
B: get s, v, choose b random, $B = g^b$
$B- > A : s, u, B$
A: $x = h(pwd, s), S = (B - g^x)^{a+ux}, K = H(S), M_1 = H(A, B, K)$
$A- > B : M_1$
A: $S = (Av^u)^b, K = H(S)$, check $M_1 == H(A, B, K), M_2 = H(A, M_1, K)$
$B- > A : M_2$
B: check $M_2 == H(A, M_1, K)$
Not very formal security analysis. Some early versions of SRP posted to sci.crypt in 1997 were attacked and broken. Submitted as input to P1363.

[wulf97newModel] William A. Wulf, Chenxi Wang, and Darrell Kienzle. A new model of security for distributed systems. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 34–44. ACM Press, 1997. UIC

Annotation: Describes a security model based on object methods for large scale heterogeneous objects. Intended to support user defined security.

[zurko97userCentered] Mary Ellen Zurko and Richard T. Simon. User-centered security. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 27–33. ACM Press, 1997. UIC

25 New Paradigms

[baker97sand] Dixie B. Baker. Fortresses built upon sand. In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 148–153, New York, 1997. ACM Press. UIC

Annotation: Discusses some of the issues that arise from having systems composed of multiple heterogeneous organizations.

[dobson93otherConcepts] J. Dobson. New security paradigms: What other concepts do we need as well? In *Proc. of the New Security Paradigms Workshop (NSPW)*, pages 7–18, Washington - Brussels - Tokyo, 1993. IEEE. UIC

[witten01openSource] Brian Witten, Carl Landwehr, and Michael Caloyannides. Does open source improve system security? *IEEE Software*, 18(5):57–61, 2001. UIC Public

Annotation: Looks at the efficacy of open source for security. Points out that trust of closed source also means trust of compilers and elimination of randomizing defenses for buffer overflow. Some types of threats best discovered by code review such as backdoors and race conditions. Commercial software goal is to generate upgrades which may indirectly increase security issues. Notes that open source responsiveness to security advisories (measured by Red Hat) are faster than proprietary and could be further cut in half.

Quote: "The recent Digital Millennium Copyright Act and UCITA legislation seem designed to discourage law abiding citizens from reconstructing source code to improve its security but are unlikely to deter those with baser motives."