The PEI Framework for Application-Centric Security

Ravi Sandhu
Executive Director and Endowed Professor
Institute for Cyber Security
University of Texas at San Antonio
ravi.sandhu@utsa.edu
www.profsandhu.com

***Abstract*—This paper motivates the fundamental importance of application context for security. It then gives an overview of the PEI framework for application-centric security and outlines some of the lessons learned in applying this framework. PEI stands for Policy, Enforcement and Implementation, signifying three distinct layers at which security policy and design decisions need to be made. The framework was introduced by this author in 2006 [35]. It is closely related to the earlier OM-AM framework also introduced by this author in 2000 [32].**

I. Introduction

There can be no security without application context. This is a fundamental premise which seems almost self-evident. Nonetheless there has been serious and consequential disagreement on this issue, so it is worth our while to spend some time discussing this premise. We will begin with some history.

A. Orange Book Era

The early era of computer security was in fact based on the opposite premise, which might be stated as follows: application context makes security impossible to achieve. Alternately, application context is bad for security. This era culminated in the Orange Book [10], so named for the color of its cover. The Orange Book definition of security was largely centered on information flow in a lattice of security labels [5], [7], [31], commonly known as multilevel security. The Orange Book was particularly concerned with the vulnerability of covert channels that could be exploited by cooperating Trojan Horses. Other system capabilities such as discretionary access control, authentication and auditing were also covered. The central concept was that of a security kernel that would enforce the information flow in terms of Operating System objects such as files independent of application context.

The Orange Book was followed by numerous continuations which came to be known as the Rainbow Series due to the variety of colors used (indeed in excess of the seven colors of the physical rainbow and with some duplicates). The current Wikipedia entry on the Rainbow Series lists thirty-four titles from 1986-1994 identified as the "most significant Rainbow Series books." Remarkably only one of these deals with an application technology.

The Trusted Database Interpretation of the Trusted Computer Systems Evaluation Criteria [12] commonly known as the TDI was written to extend the evaluation classes of the Orange Book to "trusted applications in general, and database management systems in particular." The TDI had to confront the situation that the relational data model requires integrity constraints that cut across security levels. An example of this arises with foreign key integrity. A tuple in one relation may reference a tuple in a second relation via the primary key of the second relation. Thus tuples in a Professor relation may reference a tuple in a Department relation. Foreign key integrity requires that the referenced tuple in the Department relation should exist, otherwise we have a dangling reference and integrity of the data base is compromised. Simply put, Professors cannot belong to non-existent Departments. It follows that before a Department is deleted all Professors belonging to it must de-reference the Department by assignment to another Department or possibly no Department. In a multilevel secure database some Professors may be secretly assigned to a Department. A user who is unprivileged to see this secret assignment may incorrectly conclude that the Department has no Professors and is eligible for deletion. Deletion would leave the secret Professor with a dangling reference. Preventing the deletion would leak information about the existence of a secret Professor in the Department. The amount of information leaked (or integrity compromised) may seem innocuous in this example but it has been known since the earliest days [24] that malicious programs, also known as Trojan Horses, can exploit such covert channels to rapidly move large amounts of secret information to unprivileged users. Likewise, compromise of integrity can spread and damage the utility of the database.

The TDI underscored the difficulties in reconciling prohibited information flows across security levels with the requirements of data integrity. Attempts to reconcile this conflict without consideration of application semantics [9] turned out to be infeasible [18]. It should be mentioned that leading Operating System and Database Management System vendors and several research groups invested considerable effort in bringing Orange Book and TDI based systems to market [17], [28]. Nevertheless, this overall approach can be said to have largely failed if only due to the scarce presence of such products in the market today.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8382
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8382

B. Fundamental Flaws of the Orange Era

It is beyond the scope of this paper to analyze in depth the reasons for failure of the Orange Book agenda. However, if our discipline of cyber security is to advance we must strive to learn from failures of the past. In my opinion there are three major reasons why the Orange Book was doomed to failure, given below in no particular order.

The possible exploitation of Trojan Horse driven covert channels was the wrong problem to focus on. While the dangers of malware are apparent to an informed user of the Internet today the exploitation of covert channels remains a distant threat in the current ecosystem. Attackers have much easier means to attack. They do exploit malware extensively but to date there are no documented covert channel based attacks in the wild. Moreover, the technical challenge of closing covert channels, especially high-speed covert channels available in low-level hardware mechanisms invented to increase performance, turned out to be almost impossible [15], at least without seriously compromising performance. The TDI further demonstrated the difficulty of achieving strict information flow controls in the context of Database Management Systems. Eliminating covert channels obscured the semantics of the underlying data since integrity constraints that crossed multiple labels became unenforceable. The resulting data models [19], [33], while clever and elegant, never received significant traction in practice. Last but not least we have the inference problem wherein users can infer sensitive information from non-sensitive information often in combination with other information they may possess [13]. Inference is not covered by the classic Orange Book controls.

The enforcement of information flow in a lattice of security labels was irrelevant for most applications. It is easy to see that operations that are equivalent in information flow terms are often very different from an application perspective. Credit or debit operations on an account both require read and write access to the account. Thereby they are equivalent from an information flow perspective. However, the simple intuitive distinction between these operations and the need to distinguish authorization for these is evident to any consumer who owns a bank account. Most of us would naturally consider debits more sensitive than credits. Many, if not most, applications need to distinguish the purpose of the read-write and the means used to carry out the read-write which goes beyond the information flow focus of the Orange Book. The read-write is not arbitrary but is carried out by abstract operations, sometimes called transformation procedures [6], which enforce integrity, confidentiality and usage properties with respect to the data and application context.[1] Further, in many applications separation of duty, which requires multiple users to effect a sensitive operation, is a dominant concern relative to information flow. While it is possible to enforce special forms of Separation of Duty such as Chinese Walls using an information flow lattice [30], [31], there are fundamental aspects of this principle that have nothing to do with information flow [3], [29], [36].

The exclusion of cryptography from the Rainbow Series was unsustainable in secure distributed systems. In the Orange Book era there was a strong separation between cryptography and computer security. The reasons for this separation and the long battles by the computer industry "rebels" to break it are well chronicled [25]. Suffice it to say that in context of the Rainbow Series this separation led to some absurd situations wherein the Trusted Network Interpretation [11] speaks to security of networks without any mention of encryption. Instead packets carry security labels thereby advertising to the attacker which are the more attractive ones! Researchers today recognize the intertwining of cryptography and access control not only on the network (data in motion) but also on the disk (data at rest) and even during computation (data in use) wherein cryptographic keys are only usable by approved software [1], [2].

Subsequent to the Rainbow Series the Common Criteria was developed as an ISO standard.[2] While the Common Criteria arguably fixes some of the problems of the Rainbow Series it has significant problems of its own and has received only grudging attention from vendors who market to the Government [16].

C. Post Orange Era

The failure of the Orange Book agenda and the tremendous growth of the Internet in the 1990's led to several important developments which are briefly reviewed below, in no particular order.

- *The emergence and dominance of Role-Based Access Control.* Role-Based Access Control (RBAC) was first formalized in a family of models in 1996 [34] which subsequently evolved into a NIST/ANSI standard [14]. It was the first serious alternative to the Mandatory and Discretionary Access Control Models codified in the Orange Book. RBAC rapidly became the dominant form of access control in commercial products and is likely to remain with us for a long time. RBAC has a natural affinity to be application oriented, since roles and their permissions, users and constraints are ultimately application driven.
- *The emergence of perimeter protection and the patch cycle.* Firewalls and software patches have dominated the lives of security professionals for the past two decades. The paradigm of a hard exterior and soft interior has become codified as best practice. At the same time the impossibility of completely hardening the exterior along with the necessity to harden some of the interior has come to be appreciated.

[1] As a consequence the application programs that implement these abstract operations become part of the Trusted Computing Base in Orange Book parlance and thereby make it difficult to achieve high assurance. Hence the notion that application context makes security impossible to achieve. This is a serious concern and there may well be limits to the assurance of application-centric security. However, in principle, applications are able to mitigate the risks by compensating mechanisms at the application layer, and by providing the right level of abstraction to address mission risk.

[2] http://www.commoncriteriaportal.org/thecc.html

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8382
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8382

- *The emergence of intrusion detection and prevention.* Intrusion detection [8] and later intrusion prevention [37] systems brought in the detect-and-react paradigm into cyber security. Although these techniques have their fundamental limits [4] they have become an essential part of the cyber defense arsenal. At the application layer these techniques are often called fraud detection and have been successfully use in the global credit card system including e-commerce.
- *The emergence of highly motivated and sophisticated attackers.* The criminal enterprise and its innovative and rapidly evolving infrastructure has emerged in the 2000's as the most significant recent event in cyber security. The recent FBI takedown of the Dark Market criminal network[3] is but one example of this phenomena. Criminals will attack at the weakest points including at the application layer. The insidious nature of modern attacks has led to the notion that a persistent embedded insider is possibly present in every network.

D. Emerging Application-Centric Era

I believe that after this post-Orange era we are now moving into a new era that is best described by my newly coined term application-centric.[4] The Orange Book and post-Orange eras can be described as the eras of enterprise security. Security concerns were driven largely by the priorities and needs of individual organizations, notwithstanding their presence on the Internet. The applications provided were cyber analogs of previously existing applications, such as banking, brokerage, retail, auctions etc. These applications are principally provided by enterprises for its customers. The applications of the future are just taking shape and they will surprise us. While my crystal ball does not reveal the precise form these might take, it is my strong belief that these applications will have three significant characteristics from a cyber security perspective.

1) First these application will need to reconcile differing and competing concerns of multiple parties. It is already true that the interests of a consumer of a banking service, for example, are often different from those of the bank. In the future we can expect there to be a larger number of parties involved each of whose security concerns need to be reconciled and satisfied.
2) Second the security concerns will be much fuzzier than in the past. Traditional access control sees authorization as binary. Either you are authorized or not. Many familiar systems today limit the rate and total amount of access in order to contain damage. For example, an ATM machine will limit the number of withdrawals, the amount of each withdrawal and the total amount to protect the bank and the consumer against attacks. Authorization then becomes a function of usage.
3) Third the nature of the attacks and threats protected against will need to made much more explicit. Rate-limiting activity to human speed is a simple and effective defense against machine-based attacks. The use of captchas has become commonplace as a means of distinguishing humans from automated bots, but the criminals have responded by outsourcing captcha solutions to low-cost labor in real-time!

As new applications emerge their security needs will need to be analyzed in light of the above three security characteristics. The ultimate problem is not so much sloppy coding or subtle bugs, but rather a simple and effective understanding of the real security needs of the application and its users. Hence the anticipation of a new era of application-centric security. It is our firm belief that in the future only application context can drive the policy requirements.

It should be mentioned that traditional security models were not built with these three characteristics in mind. In the past five years or so academic research has developed the notion of usage control [26], [27] to provide additional sophistication beyond traditional access control. These notions provide a sound basis for capturing the security needs of future applications. However, in addition to models we need a framework and methodology to develop application-centric security models. This brings us to our next topic.

II. The PEI Framework

We have developed the PEI framework illustrated in figure 1. This framework clearly separates three layers in the design process [35]. These are the Policy (P), Enforcement (E) and Implementation (I) layers. At each layer we need formal models to express and analyze the security policy. The PEI framework focusses on the three inner layers. Hence its name.[5]

The topmost objectives layer is deliberately informal and seeks to elicit the high level security and system goals. It is the layer at which the business or missions owners provide input and judgement about the major tradeoffs within competing security and functional needs and desires of individual and multiple parties. I am firmly convinced that attempts to formalize this layer are mistaken and we must leave it deliberately informal and non-mathematical. It remains an open question of how to do this. The bottom layer produces actual running code, using some flavor of trusted computing technology. The correspondence of this code to the formal and quasi-formal models of the I layer poses a challenging and important problem, but is not the main problem of application-centric security.

The three inner layers of PEI are intended to have a many to many relation. Thus a policy model at the P layer may have many different manifestations at the E layer. Conversely

[3] http://www.fbi.gov/pressrel/pressrel08/darkmarket101608.htm

[4] Application-centric security is different from application security. I understand application security to be the discipline that mitigates errors, bugs and sloppy practice in the application life cycle and supporting environment by techniques such as application firewalls, vulnerability scanners and secure coding. A good example of this in the web applications arena is represented by the OWASP community (see www.owasp.org).

[5] Policy in the larger sense permeates all layers of this framework. At the P layer the policy that is of concern is intrinsic to the application and unrelated to policy decisions regarding enforcement or implementation. As such it should more precisely be called application policy.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8382
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8382

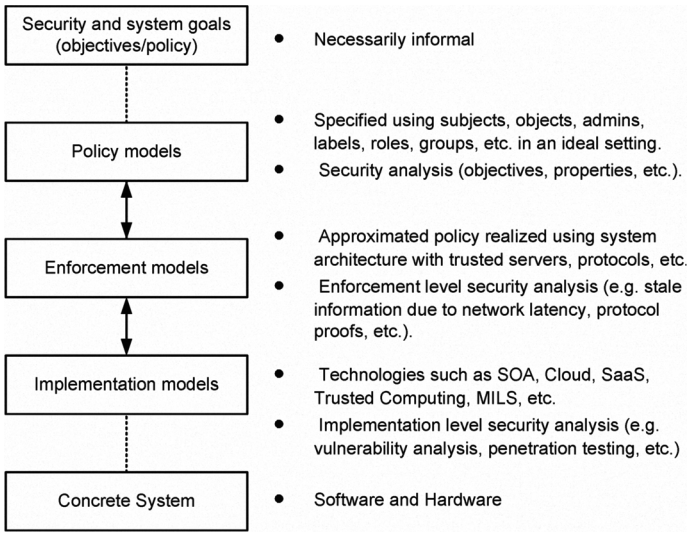

Fig. 1. The PEI Models Framework (At the E layer Architecture is often used rouhgly synonymous with Model, and at the I layer Architecture and Platform are often so used).

an enforcement model at the E layer may be able to support many different models at the O layer. For example, a suitably configurable attribute-based enforcement model at the E layer can enforce distinct P layer models such as Role-Based Access Control or Mandatory Access Control. In this respect the layers are closer to the layers of a network stack rather than the layers of a classic waterfall software engineering methodology.

Turning to the three PEI layers we discuss some salient characteristics. At the P layer the policy model is developed in an idealized context where it is assumed the relevant information required to make access decisions, such as subject-object attributes and attendant policies, is instantly and securely available and up to date. One can visualize this as idealized centralized system where everything is together in one computer. The motivation is to focus on the real policy needs of the application without being distracted by implementation details and practical realities of distributed systems. The E layer deals with the approximations and additional servers introduced by the distributed nature of real-world distributed systems. The goal is to make the approximations explicit and controllable since perfect correspondence to the idealized P layer is impossible. Finally the I layer spells out detailed implementation protocols and mechanisms. The concept of a model arises at all three layers. At the E layer the term architecture is also used, while at the I layer architecture and platform are also used. While many researchers implicitly follow layers similar to these, PEI is the first framework to explicitly articulate and evolve the methodology.

Examples of the many-to-many relationship between the PEI layers are given in [23], [35] in context of secure information sharing and in [32] in context of Role-Based Access Control. The Enforcement model discussed in [20] shows the nature of the architecture and model required at this layer and directly addresses the approximation question. This E model

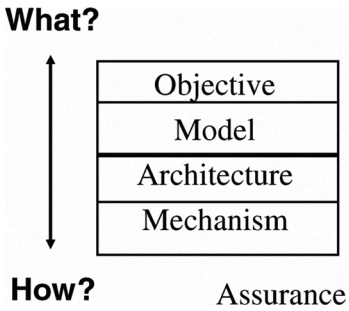

Fig. 2. The Precursor OM-AM Framework.

can enforce the various P models outlined in [22] and further developed in [21].

A. Relationship of PEI to OM-AM

It should be mentioned that PEI is an evolution of the earlier OM-AM framework [32] illustrated in figure 2. The Objective (O) layer of OM-AM corresponds directly to the topmost objectives layer of PEI. The Models (M) layer of OM-AM corresponds to the P or Policy Models layer of PEI. The Architecture (A) layer of OM-AM corresponds to the E or Enforcement Models layer of PEI. Perhaps the E layer of PEI should have been called the Enforcement Architecture and Models layer to make this correspondence more evident. This layer has two major aspects. An Enforcement Architecture in terms of authorization, authentication, validation and certificate servers and so on, and the major protocol flows between these is needed at this layer. At the same time an Enforcement Model is also needed to specify details such as the tolerance for possibly stale security information that is either locally cached or piggy-backed along with other protocol flows. In the OM-AM formulation my choice was to emphasize the Architecture aspect since the model aspect was nascent at that time. In PEI I have chosen to emphasize the models aspect in the E layer. The Mechanism (M) layer of OM-AM corresponds directly to the Implementation layer of PEI.

III. Conclusion

In this paper I have given a brief overview of my thoughts on the new emerging era of application-centric security and the PEI framework that is applicable to this purpose. PEI was formulated in 2006 and is closely related to the earlier OM-AM framework of 2000. The term application-centric has been coined in this paper, so PEI and its roots predate this term considerably. While there are only a few papers of mine that directly address PEI, it has been a guiding principle for my research since the OM-AM days. I am sure PEI will evolve in the future.

Acknowledgement

I thank the conference organizers for the opportunity to present this invited paper.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8382
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8382

REFERENCES

[1] TCG Specification Architecture Overview. *http://www.trustedcomputinggroup.org*.

[2] *Intel trusted execution technology - preliminary architecture specification*. Intel Corporation, 2006.

[3] G.-J. Ahn and R. Sandhu. The RSL99 language for role-based separation of duty constraints. pages 43–54, Fairfax, VA, October 28-29 1999. ACM.

[4] S. Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Secur.*, 3(3):186–205, 2000.

[5] D. Bell and L. LaPadula. Secure computer systems: Unified exposition and Multics interpretation. Technical Report ESD-TR-75-306, The Mitre Corporation, Bedford, MA, March 1975.

[6] D. Clark and D. Wilson. A comparison of commercial and military computer security policies. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 184–194, Oakland, CA, May 1987.

[7] D. Denning. A lattice model of secure information flow. *Communications of the ACM*, 19(5):236–243, 1976.

[8] D. E. Denning. An intrusion-detection model. *IEEE Trans. Softw. Eng.*, 13(2):222–232, 1987.

[9] D. Denning et al. The SeaView security model. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, pages 218–233, Oakland, CA, 1988.

[10] Department of Defense National Computer Security Center. *Department of Defense Trusted Computer Systems Evaluation Criteria*, December 1985. DoD 5200.28-STD.

[11] Department of Defense National Computer Security Center. *Trusted Network Interpretation of the Trusted Computer Systems Evaluation Criteria*, July 1987. NCSC-TG-005.

[12] Department of Defense National Computer Security Center. *Trusted Database Interpretation of the Trusted Computer Systems Evaluation Criteria*, April 1991. NCSC-TG-021.

[13] C. Farkas and S. Jajodia. The inference problem: a survey. *SIGKDD Explor. Newsl.*, 4(2):6–11, 2002.

[14] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli. Proposed nist standard for role-based access control. *ACM Trans. Inf. Syst. Secur.*, 4(3):224–274, 2001.

[15] W.-M. Hu. Reducing timing channels with fuzzy time. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, pages 8–20, Oakland, CA, May 1991.

[16] W. Jackson. Under attack: Common criteria has loads of critics, but is it getting a bum rap? *Government Computer News*, Aug 10, 2007.

[17] T. Jaeger. *Operating System Security*. Morgan Claypool, 2008.

[18] S. Jajodia and R. Sandhu. Polyinstantiation integrity in multilevel relations. In *Proceedings of IEEE Symposium on Research in Security and Privacy*, pages 104–115, Oakland, CA, May 1990.

[19] S. Jajodia and R. Sandhu. Toward a multilevel secure relational data model. In *Proc. ACM SIGMOD International Conference on Management of Data*, pages 50–59, Denver, CO, May 1991.

[20] R. Krishnan, J. Niu, R. Sandhu, and W. Winsborough. Stale-safe security properties for group-based secure information sharing. In *Proceedings of the 6th ACM workshop on Formal methods in security engineering*, pages 53–62. ACM New York, NY, USA, 2008.

[21] R. Krishnan, R. Sandhu, J. Niu, and W. Winsborough. Foundations for Group-Centric Secure Information Sharing Models. *Proc. 14th ACM Symposium on Access Control Models and Technologies (SACMAT), Stresa, Italy, June 3-5, 2009, to appear.*

[22] R. Krishnan, R. Sandhu, J. Niu, and W. Winsborough. A conceptual framework for group-centric secure information sharing. *Proc. of 4th ACM Symposium on Information, Computer and Comm. Security*, pages 384–387, 2009.

[23] R. Krishnan, R. Sandhu, and K. Ranganathan. PEI models towards scalable, usable and high-assurance information sharing. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 145–150, New York, NY, USA, 2007. ACM.

[24] B. W. Lampson. A note on the confinement problem. *Commun. ACM*, 16(10):613–615, 1973.

[25] S. Levy. *Crypto: How the Code Rebels Beat the Government Saving Privacy in the Digital Age*. Penguin Books, 2001.

[26] J. Park and R. Sandhu. The UCON ABC usage control model. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):128–174, 2004.

[27] A. Pretschner, M. Hilty, and D. Basin. Distributed usage control. *Communications of the ACM*, 49(9):39–44, 2006.

[28] W. Rjaibi. An introduction to multilevel secure relational database management systems. In *CASCON '04: Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research*, pages 232–241. IBM Press, 2004.

[29] R. Sandhu. Separation of duties in computerized information systems. In S. Jajodia and C. Landwehr, editors, *Database Security IV: Status and Prospects*, pages 179–189. North-Holland, 1991.

[30] R. Sandhu. A lattice interpretation of the Chinese Wall policy. In *Proceedings of NIST-NCSC National Computer Security Conference*, pages 329–339, Baltimore, MD, October 1992.

[31] R. Sandhu. Lattice-based access control models. *IEEE Computer*, 26(11):9–19, November 1993.

[32] R. Sandhu. Engineering authority and trust in cyberspace: the OM-AM and RBAC way. In *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*, pages 111–119, New York, NY, USA, 2000. ACM.

[33] R. Sandhu and F. Chen. The multilevel relational data model. *ACM Transactions on Information and System Security*, 1(1), November 1998.

[34] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, February 1996.

[35] R. Sandhu, K. Ranganathan, and X. Zhang. Secure information sharing enabled by trusted computing and PEI models. In *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 2–12, New York, NY, USA, 2006. ACM.

[36] R. Simon and M. Zurko. Separation of duty in role-based environments. pages 183–194, Rockport, Mass., June 1997.

[37] R. Uppuluri. Synthesizing fast intrusion prevention/detection systems from high-level specifications. In *the Proceedings of the 8th USENIX Security Symposium*, 1999.

Digital Object Identifier: 10.4108/ICST.COLLABORATECOM2009.8382
http://dx.doi.org/10.4108/ICST.COLLABORATECOM2009.8382