2014

Security Policies That Make Sense for Complex Systems: Comprehensible Formalism for the System Consumer

Rhonda R. Henning
Nova Southeastern University, rhenning@harris.com

Security Policies That Make Sense for Complex Systems: Comprehensible Formalism for the System Consumer

by

Ronda R. Henning

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

in

Information Systems

Graduate School of Computer and Information Sciences
Nova Southeastern University

2013

We hereby certify that this dissertation, submitted by Ronda Henning, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

_____ _____

James D. Cannady, Ph.D. Date
Chairperson of Dissertation Committee

_____ _____

Joseph Gulla, Ph.D. Date
Dissertation Committee Member

_____ _____

Rayford Vaughn, Jr. , Ph.D. Date
Dissertation Committee Member

Approved:

_____ _____

Eric S. Ackerman, Ph.D. Date
Dean, Graduate School of Computer and Information Sciences

Graduate School of Computer and Information Sciences
Nova Southeastern University

2013

An Abstract of a Dissertation Submitted to Nova Southeastern University

in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Security Policies That Make Sense for Complex Systems: Comprehensible Formalism for the System Consumer

By

Ronda R. Henning
December 2013

Information Systems today rarely are contained within a single user workstation, server, or networked environment. Data can be transparently accessed from any location, and maintained across various network infrastructures. Cloud computing paradigms commoditize the hardware and software environments and allow an enterprise to lease computing resources by the hour, minute, or number of instances required to complete a processing task. An access control policy mediates access requests between authorized users of an information system and the system's resources. Access control policies are defined at any given level of abstraction, such as the file, directory, system, or network, and can be instantiated in layers of increasing (or decreasing) abstraction. For the system end-user, the functional allocation of security policy to discrete system components, or subsystems, may be too complex for comprehension. In this dissertation, the concept of a metapolicy, or policy that governs execution of subordinate security policies, is introduced. From the user's perspective, the metapolicy provides the rules for system governance that are functionally applied across the system's components for policy enforcement. The metapolicy provides a method to communicate updated higher-level policy information to all components of a system; it minimizes the overhead associated with access control decisions by making access decisions at the highest level possible in the policy hierarchy. Formal definitions of policy often involve mathematical proof, formal logic, or set theoretic notation. Such policy definitions may be beyond the capability of a system user who simply wants to control information sharing. For thousands of years, mankind has used narrative and storytelling as a way to convey knowledge. This dissertation discusses how the concepts of storytelling can be embodied in computational narrative and used as a top-level requirements specification. The definition of metapolicy is further discussed, as is the relationship between the metapolicy and various access control mechanisms. The use of storytelling to derive the metapolicy and its applicability to formal requirements definition is discussed. The author's hypothesis on the use of narrative to explain security policy to the system user is validated through the use of a series of survey instruments. The survey instrument applies either a traditional requirements specification language or a brief narrative to describe a security policy and asks the subject to interpret the statements. The results of this research are promising and reflect a synthesis of the disciplines of neuroscience, security, and formal methods to present a potentially more comprehensible knowledge representation of security policy.

Acknowledgements

This dissertation was facilitated by many people along the journey; those who kept me on the path and offered directions along the way. I would be remiss not to acknowledge their contributions here, starting with those who encouraged my curiosity about security policies and their implementations Roger R. Schell, Marvin Schaefer, David E. Bell, and Marshall Abrams were never too busy to very patiently answer my never-ending questions. Blaine Burnham helped shape my initial dissertation subject, and Rayford Vaughn convinced me that solving world peace was a noble thought but not the answer to a survivable dissertation. Richard Ford became the voice of urgency, as he consistently and patiently inquired about my progress. Joe Gulla helped me clarify the interrelationship between traditional requirements and security requirements. James Cannady let me take the path less traveled as I worked through the cognitive neurosciences and computer assisted narrative technologies to fuse the concepts into my dissertation research.

My co-workers and peers offered more words of encouragement than I ever thought I would need: their inquiries of "are you done yet?" and patient endurance when I described why I was still working on it kept me moving in the right direction. They may not have understood it, but they understood it was my passion.

My parents, Ronald and Julia, instilled my curiosity and love of learning at an early age. They never discouraged my studies and taught me that hard work would eventually pay off. For all the papers they reviewed and trips to the library over the years: Thank You. My siblings, Julia, Gloria, and Ronald, were always there to remind me that family weighed in balance with studies.

Finally, none of it would ever have happened without the enduring love and patience of my late husband, Steven Eric Rose, who did not live to see the completion of this milestone. Steven encouraged my goals and did his best to push me along the way to fulfill my dreams. He never complained about my hours of research work, or my professional obligations. Without his love, support and devotion I would have never made it this far.

+

Table of Contents

APPENDICES

List of Tables

List of Figures

List of Figures (continued)

Chapter 1

Introduction

The Oxford English Dictionary defines security (OED, 2012) as:

- the state of being free from danger or threat: *the system is designed to provide maximum **security against** toxic spills; job security*
- the safety of a state or organization against criminal activity such as terrorism, theft, or espionage: *a matter of national security*
- procedures followed or measures taken to ensure the safety of a state or organization: *amid tight security the presidents met in the Colombian resort*
- the state of feeling safe, stable, and free from fear or anxiety: *this man could give the emotional security she needed*

and further defines cybersecurity (OED, 2012) as:

[as noun]:
the state of being protected against the criminal or unauthorized use of electronic data, or the measures taken to achieve this:
some people have argued that the threat to cybersecurity has been somewhat inflated

[as modifier]:
IT security professionals said that outsourcing would be the biggest cybersecurity threat

From these definitions, it becomes evident that a sense of security requires more than just a single person: it requires a person with something that needs security, such as data, and a second person who desires to take that sense of security away from an individual by stealing the information or otherwise exploiting a vulnerability. In Medieval Times, serfs gained a sense of security by working for lords and living within the walls of their castles. And so it is today, as we seek a sense of security for our data with protective

countermeasures such as firewalls, intrusion detection systems, and virtual private networks.

Whether we wish to admit it or not, security is a social function (Schaefer, 2009): It is about a person, organization, or entity determining which information to share. Schaefer further states:

> Security, at its most basic, requires the ability to differentiate, to recognize *one object* is different from *another*. We differentiate objects for the many reasons we assign attributes and values. For survival, for example, we place a skull and crossbones on a bottle of poison. How do we differentiate? One assigns an attribute to an observation as a label with meaning. The label is something reliably, repeatedly measurable, using our five senses, perhaps using tools or technology as extensions of our senses. Differentiation may be made by scientific observation and objectivity, or by just because, reasoning overridden by *subjective feeling* (Schaefer, 2009, p. 2) (emphasis from original text).

Schell states that labels can be used as a social mechanism for understanding the sensitivity of the user's current context; be it processing the reading or the writing of information. That is, labels provide the user an external cue as to the sensitivity of the information when it is addressed in the environmental context of who, what, when, where, why, and how (Schell, 2001).

In this context, a *security policy* addresses the protection mechanisms employed to protect an organization's assets from potential misuse. In reality, a security policy is a set of clearly articulated rules that specify constraints about data usage (Bell D. E., 2005). The access control policy defines how system users interact with the data stored within the system.

This dissertation explores the use of structured storytelling paradigms to extract the security policy for the system consumer. We begin with a background

discussion of security policy and traditional formalism used for policy expression. From this foundation, we move to a discussion of secure system design techniques based in formalism, and explore the usability constraints associated with formal methods. We present an alternative policy definition strategy: that of storytelling, and describe an experiment to determine the usability of this approach. Finally, we present the results of the experiment; discuss its implications, and potential areas for further research in this area.

Historical Background

Winsborough poses the question of whether policy can be distinguished from requirements and states that policy is intentional, in that it provides rules characterizing the author's intentions for system behavior; whereas requirements are not concerned about the underlying intentions (Winsborough, 2004). Winsborough further states that a policy is a set of rules that are used to manage and control the state and state transitions of one or more managed objects (p.20) An individual policy rule is considered an intelligent data container; containing:

- knowledge and metadata that define the semantics and behavior of the policy rule and its effect on the rest of the system.
- A group of events that can be used to trigger the evaluation of the condition clause of a policy rule.
- A group of conditions aggregated by the policy rule.
- A group of actions aggregated by the policy rule (p. 20)

Beyond these statements, Winsborough speculates on the feasibility of a single policy language, responsible for translating policies into plans of action, composing policies, and interpreting policies amongst domains. Further, such a language would have to

address conflict detection and resolution, dependency analysis, and allow translation into multiple graphical user interfaces to address the needs of various constituencies (p. 42). He concludes with a wish list for policy specification, namely that policy creation techniques should be (i) sufficiently expressive of preferences regarding cost v. performance, security, risk, and reliability; (ii) sufficiently structured and/or naturally suited to human psychology and cognition to keep specification errors to an absolute minimum; and (iii) robust to specification errors (p. 43).

Current generation computer systems are rarely monolithic architectures. Rather, there are user clients, connected to various servers through the Internet, traversing routers, switches, and firewalls to fulfill information requests. In this environment, every security policy becomes a meta-policy, and the individual policy rules can be decomposed based on their function, forming subordinate policies to address specific concerns, for example: accountability, authentication, contingency, or access control. Figure 1 illustrates a representative policy model for an enterprise application (Sherwood, Clark, & Lynas, 2008).

With such a large number of subordinate policies composing a security meta-policy, authoring a policy becomes an exercise in formal specification to ensure policy correctness. However, the higher the degree of formalism desired, the higher the degree of expertise required to generate a policy. As an example, consider the following:

Jajodia developed the Authorization Specification Language (ASL) to specify various access control policies with stratified clause form logic (Jajodia, 1997). The following are two example rules in ASL:

cando (file 1, Customer, +read) <- {members of the role customer can read file1}
cando (file 2, s, +write) <- in (s, Employee) & -^OM (s, Customer)
{subjects active in the role "Employee" and Not the role "Customer" may write to file 2}

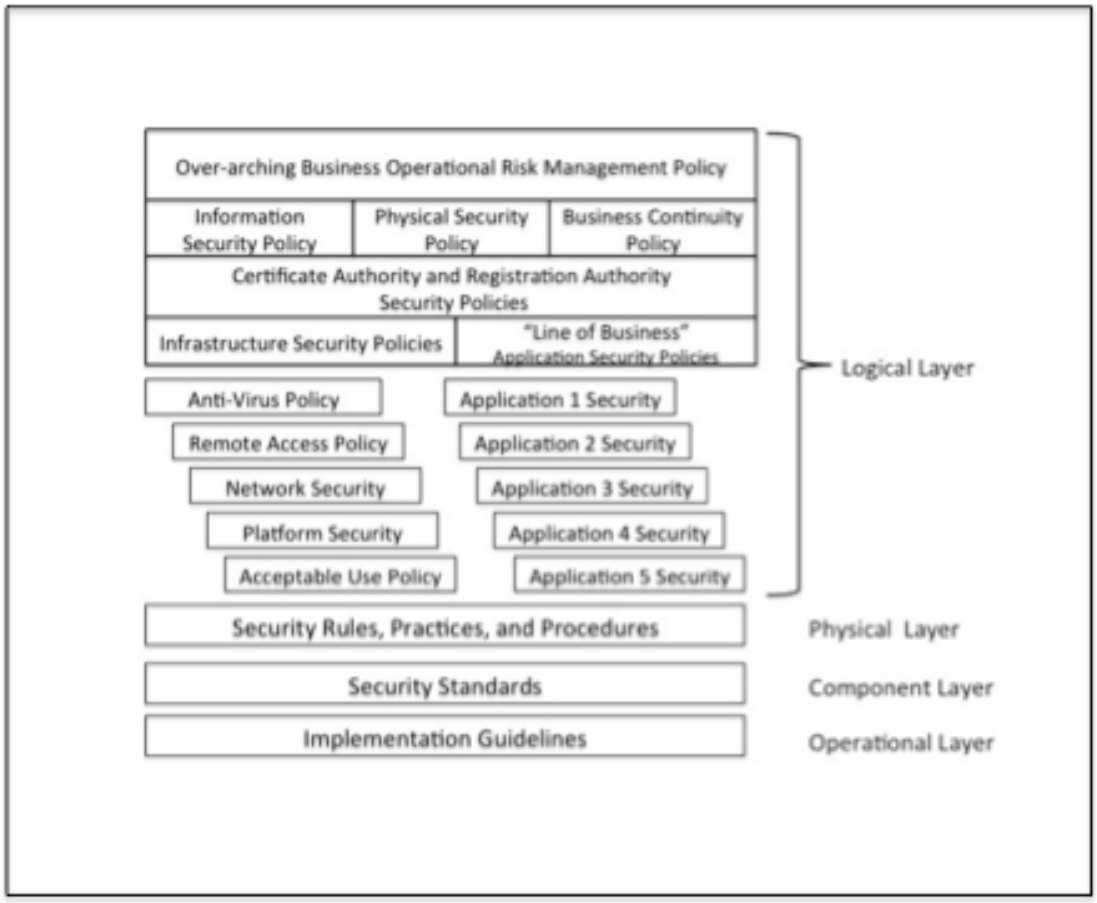
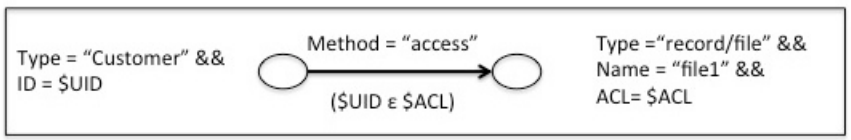


Figure 1 – The many layers of security policy (Sherwood, Clark, & Lynas, 2008).

An alternate language specification is proposed in LaSCO (Hoagland J.A., 1998), where the following graphical structure is used to indicate that a Customer needs to have an ID represented by the policy variable $UID included in the access control list of file 1 in order to have access to it:



In most access control models, access control is defined as a triple consisting of the <subject, object, privileges> associated with a given data container, for example, a file or a row in a database. In the early days of computing, much discussion surrounded how access control models should be represented in computer systems. (Schell, 1979) developed the notion of multilevel mode of operations, stating:

> In multilevel mode, the computer must internally distinguish multiple levels of information sensitivity and user authorization. Internal Controls of hardware and programs must assure that each user has access to only authorized information (p. 20).

Schell further stated:

> The security kernel design is derived directly from a precise specification (i.e. mathematical model) of its functions (like a cryptographic algorithm). This mathematical model is a precise formulation of access rules based on user attributes (clearance, need-to-know) and information attributes (classification). (p. 21).

By 2001, Schell had a slightly different perspective, stating that users want the convenience of being in the same virtual integrity domain as the least mindful and least informed among them (Schell, 2001). He further stated that to counter malicious software, systems must be designed and built to have all of the following properties:

- No exploitable flaws.
- Enforce security policies on information flow, thereby bounding the damage of malicious applications software.
- Built to be subject to third party inspection and analysis to confirm the protections are correct, complete, and do nothing more than advertised.

To summarize, the security model must be a valid representation of the behavior with regard to the information protection of the entire system. The model must include a proven security theorem, which establishes that the model's behavior always complies with the security requirements for the policy of interest rather than being a formalization

of the mechanism itself. To illustrate, Schell defined three domains for execution, as illustrated in Figure 2.

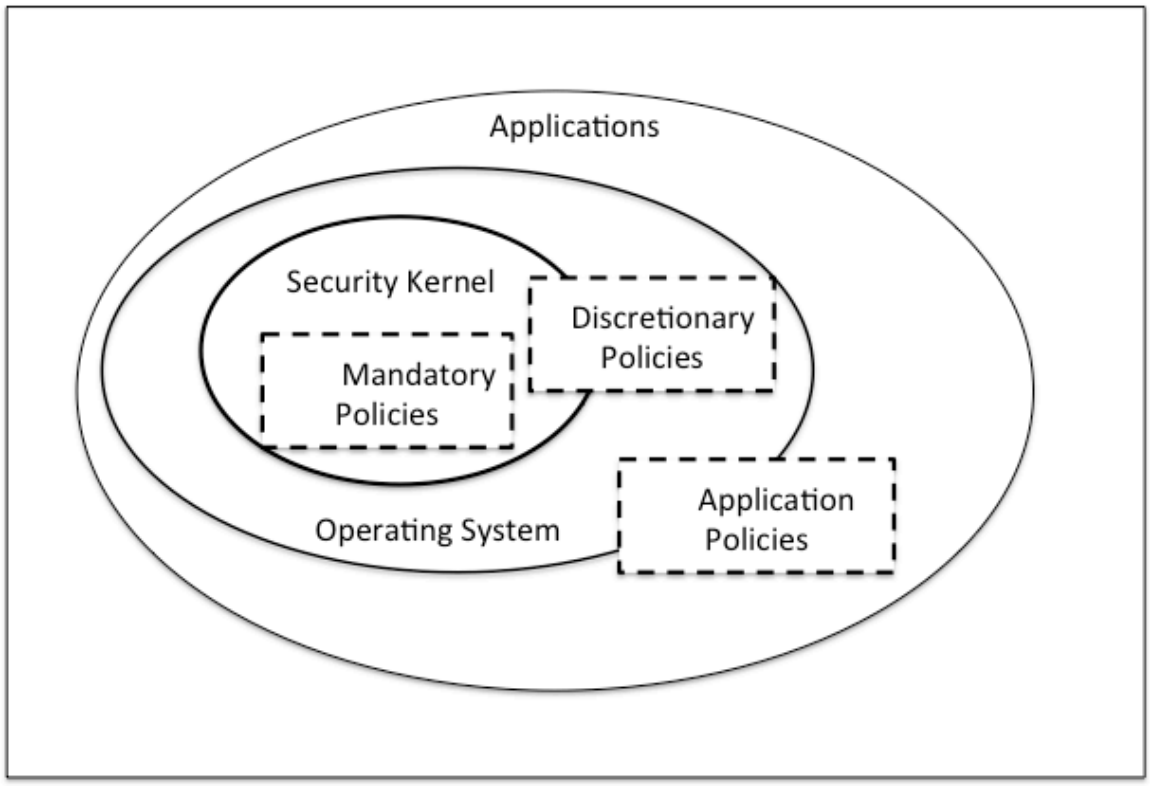


Figure 2. Domains for execution reflecting mandatory, discretionary, and application layer security policies.

Unfortunately, separate domains do not reflect the security policies of most civil government and commercial organizations. The U.S. Government has a uniform classification hierarchy for information, (Top Secret, Secret, Confidential, and Unclassified) which can be represented as a mathematical lattice structure. Commercial organizations usually do not maintain a similar uniform classification hierarchy that is recognized and enforced by other organizations, but instead implement access controls based upon a user's role in the organization. For example: a user may have a functional role (system administrator), an organizational role (manger, IT department), and an

administrative role (time card approver). The concept of Role-based access control (RBAC) was introduced by (Ferraiolo, 1995; Schell, 1979), and (Sandhu R. C., 1996) to more accurately model the workings of a commercial enterprise. In RBAC, access control is based on a four-element set (user, group, object, privileges). A user may belong to many groups, each with a different privilege set. In the worst-case model, every user has their own group, and there are as many groups to administer as there are users. In (Ferraiolo, 1995) a role is defined and centrally administered within an organization. RBAC as a modeling tool has been accepted as a useful tool that reflects both how organizations function and how information access is implemented in most commercial applications.

However, with the flexibility of RBAC, there are some limitations. In a distributed network centric enterprise, it may take several hours to confirm the update of an application's access control roles. Because several applications may use their own security services instead of the centralized security services of the operating system, it may be difficult to determine whether an access control policy has been completely administered. In fact, it may well be that RBAC in a distributed enterprise can violate the three primary engineering principals of a security reference monitor validation mechanism (Schell, 1979, p. 29):

1. Completeness – that the policy is invoked on every access to data
2. Isolation – the security mechanism is protected from unauthorized modification
3. Verifiability – the policy must be small and simple for complete test and verification.

There are times when an event may occur that requires comprehensive pre-emption or revocation of a security policy. For example, after September 11, 2001, data security policies for U.S. Government web sites were changed (U.S. Government, 2003). In these instances, waiting for confirmation that the access control policy has been updated throughout the enterprise may not be possible. (Hosmer, 1991) presented the notion of a *metapolicy* to address instances of arbitration among diverse domains implementing disparate security policies. In her paper, the use of a metapolicy to address immediate access control policy changes for an enterprise is presented. The paper discussed how a metapolicy approach differs from the current work on context and constraint based access control policies. It then discussed the problems associated with metapolicy creation and administration, including how multiple policies may coexist within a domain. The use of narrative storytelling, and computer-assisted storytelling is presented as an alternative method of meta-policy formation. This technique is examined in the context of Risk-Adaptive Access Control (RAdAC), the current state of thought for globally distributed security policies.

The remainder of this dissertation is structured as follows. The problem of timely policy administration is presented. Next, the barriers and issues associated with the coexistence of multiple policies in a single domain are addressed. A survey of the literature addressing the technologies of system design, policy, requirements engineering, and computer assisted storytelling is presented. A metapolicy creation methodology is proposed as an experiment and the experimental results are presented. Finally, future work on the use of meta-policies for access control is addressed.

PROBLEM STATEMENT

In the current generation of information systems, security policy enforcement mechanisms are dispersed throughout the system architecture as countermeasures to specific threats (Henning, 2002). For example, virus scanning, firewalls, virtual private networking (VPN) clients, and intrusion prevention sensors were all created to respond to specific threats to information systems. Figure 3 illustrates this functional architecture. In essence, every time a vulnerability is identified, publicized, and exploited, another stopgap security countermeasure is created in response to the threat.

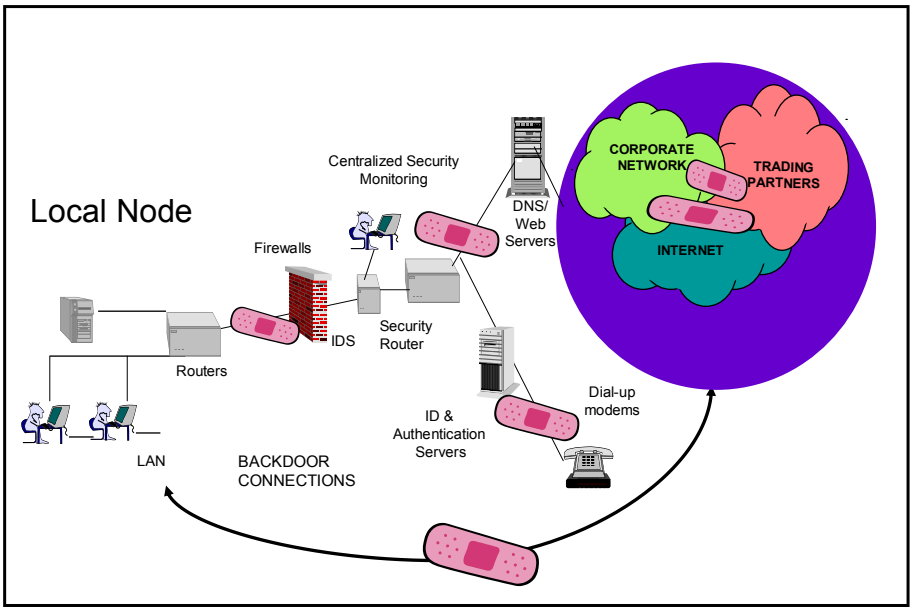


Figure 3. Functional security architecture, with countermeasures deployed against specific threats.

The problem with this approach to security is that it is highly reactive, in that system protections are deployed in response to specific threat information. Ideally,

general-purpose security mechanisms such as access control lists exist so that adaptive security policies can be defined and deployed in response to a specific threat. For example, an intrusion prevention sensor can be provided an attack signature that allows it to identify a new type of vulnerability and neutralize it.

In addition to being reactive to threats, this type of security policy is not particularly useful in an enterprise environment. Within the context of an enterprise, security policies are generally high-level, technology neutral, concern risks, set directions and procedures, and define penalties and countermeasures if the policy is transgressed (Rees, 2003). Unfortunately, those grand enterprise access control policies do not translate into implementable mechanisms. Figure 4 illustrates the breakdown between implementation capabilities and user access control policy specification.

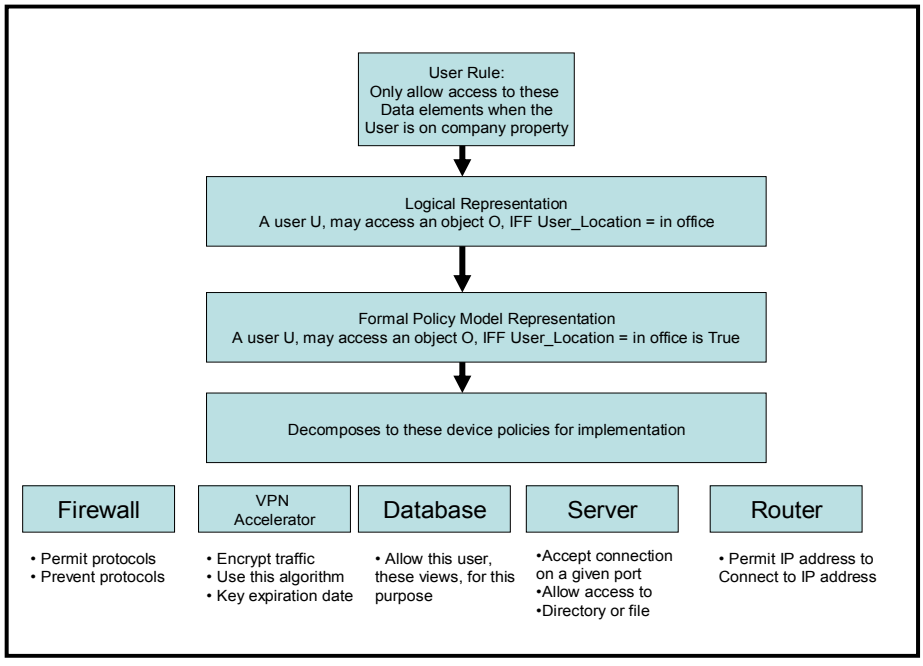


Figure 4 – Decomposition of policy model representations demonstrates the disconnect between implementation and formalism in enterprise models.

The issue, then, is how to specify a security policy at a high level that has significance to the policy users, and can be traced to lower level implementation guidance and functional system decomposition activities. Figure 5 illustrates the concept of a security policy specification that accommodates traditional non-ambiguous security policy information (user X can only see SECRET Data), contextual information (user X can only see SECRET data if in the office) and usage constraints (User X cannot send information outside a given domain), as well as more abstract notions such as "share no information outside the organization." In the ideal world, a security policy can be defined as a series of binary grant/deny operations. In reality, grant/deny operations are contingent upon environmental, organizational, and operational constraints, which may be defined with varying degrees of specificity. In the extreme case, security policy could be defined as "if a user says share the data, then share the data." However, it becomes much more difficult to embed this type of constraint within a computer system. An analogous situation would be to "know art when you see it."

In the ideal case, a security policy can be decomposed to address various *contexts, or environments,* for its enforcement. Security policies define acceptable and unacceptable behavior for software systems (Schneider, 2000). For example, a user in a remote office may not have the same access rights to information as a user at corporate headquarters. Specification of contextual security information facilitates creation of a useful access control policy.

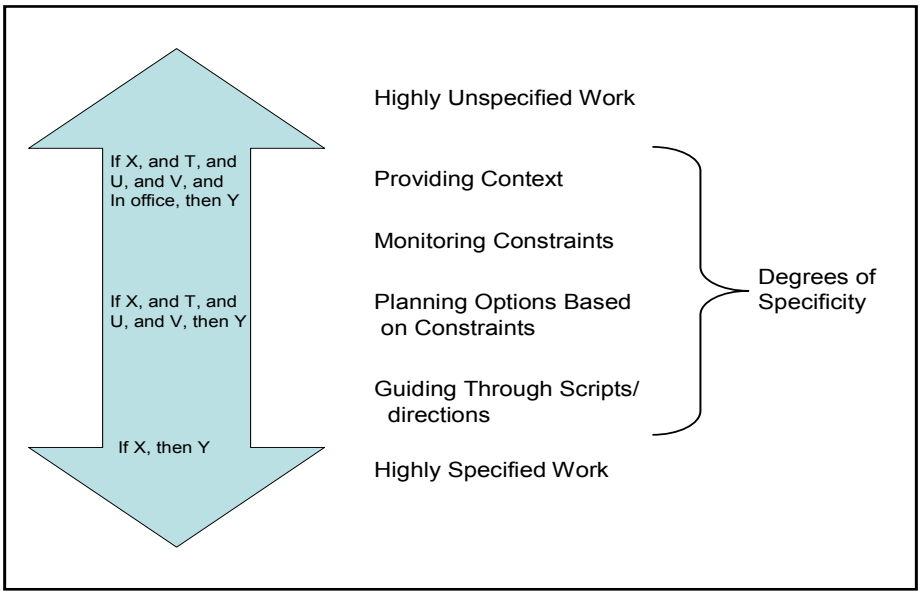


Figure 5 - Degrees of specificity that can be applied to security policy.

(Burnside & Keromytis, 2007) discuss the concept of enterprise security management with global security policies. Every policy decision is made with near-global knowledge, and then re-evaluated as the global knowledge changes. Four major types of components are applied:

1. Sensors. Small programs scattered around the network that generate events corresponding to observed network and application behavior.
2. Events. Any action performed by an application that may be relevant to some policy decision. Events may be positive or negative.
3. Policy. A list of objectives, rules for behavior, requirements and responses, whose goal is to ensure the security of the network.
4. Actuators. A program, which modifies application behavior after being triggered by a policy; the policy enforcement point.

A network, consisting of applications and network links, is observed by the sensors, each generating events in response to requests. Events are evaluated by policy, which makes decisions and notifies the actuators to modify application behavior in response.

DISSERTATION GOAL

The goal of this dissertation `is to determine the feasibility of defining a top-level security policy supporting a significant degree of formalism that can accommodate verification of completeness and traceability to lower-level functional system specifications. The objective is to define the set of logical security principles that govern system behavior, are enforceable throughout the system design process, and are comprehensible to the system stakeholders. The scope of this research is the solicitation of these logical security principles in a rigorous and standardized methodology such that formalism can be subsequently applied to the policy model. The system environment may not be well defined during early implementation phases or in usage scenarios, which, in turn, inhibits the development of effective foundation security measures. If the foundation security mechanisms are not in place early, it may not be practical to apply them during later phases without extensive system rework. An end-user should never have to be concerned with the format of an access control list or firewall rule: a general statement of access management should be traceable to lower-level implementation specific constraints. These general statements, or metapolicy, provide the overarching governance framework for an information system.

RESEARCH HYPOTHESIS

Kendall Haven, in Story Proof (Haven, 2007) states:

> Mankind has learned to read and write only in the past few hundred years. Logical, expository, and argumentative forms first emerged perhaps 5,000 years ago. But humans have been telling stories for 100,000 years or more. Evolutionary biologists tell us that 100,000 years of story

> dominance in human interaction has rewired the human brain to be predisposed before birth to think in, make sense in, and create meaning from stories (p.24).

Boyd (Boyd, 2009) states "humans are hyper-intelligent and hyper-social animals. By integrating intelligence, cooperation, pattern-seeking, alliance making, and beliefs and knowledge from other people, stories make us stronger and more effective."

Haven further states:

> By the beginning of kindergarten, the concepts of story "trouble" character, temporal sequencing, cause-and-effect sequencing, and goal are well fixed and known. Given a character and a goal, children will easily identify the type of trouble that is most likely to occur and will correctly identify that trouble will emerge to block a character from reaching the stated goal. Additionally, they *know* to search for hints of upcoming trouble. They know what to expect from a story and will adjust their perceptions and their interpretations of narrative inputs to find (or create) it (p. 25).

To extend further and summarize cognitive science: humans require that events make sense, and create or mentally invent what is needed to make sense through cause-and-effect sequencing, temporal sequencing, centering around a common theme, and character analysis. Bransford and Brown stated, "the mind imposes structure on the information available from experience and interprets (creates meaning for) experience through this story structure" (Bransford, 2000).

The research question posed is simple: if the human brain is most receptive to the narrative, story-like structure, why not apply it to the concept of security policy? Current device implementation languages such as Cisco's Internet Operating System (IOS), Linux, or Windows OS demand specific programming language syntax. This syntactic level of complexity is multiplied in a heterogeneous infrastructure of network

components, servers, clients, and applications. Expecting a system developer to understand and implement policy consistently within a complex architecture may be beyond a programmer's level of understanding and result in error-ridden implementations that are exploitable by malicious users.

Our hypothesis is that the security policy elements required to implement complex security meta-policies can be best expressed as story or narrative elements. Through the use of narrative story, security policy can be expressed in the format most familiar to mankind, and most commonly used to represent socially acceptable normative behavior. In turn, these narrative elements can be decomposed into sub-policies and system requirements such that the policy enforcement mechanisms can be explicitly traced back to the specific story elements in the top-level policy statement. What we seek to provide is the linkage from the computational language used to create machine instructions to the comprehensible language used to express policy rules in everyday life.

For example, a parent tells a child to play in the yard, not the street. A parent does not express that as a logical axiom, but the intent is to enforce a boundary on the area acceptable for play. The child comes to understand that boundary through repeated statement of the rule, and possibly punishment for disobedience. The parent intends to provide a safe play area for the child, namely, the yard. The safety of the play area is implicit in the rule, and is not logically expressed. In a computational language, the domain of the yard would be defined, and an executable statement would be tested to determine if it was within the domain or beyond the domain's boundaries.

RELEVANCE AND SIGNIFICANCE

This section discusses the various research conducted on access control models that is relevant to the concept of metapolicy formulation and implementation.

In (Hosmer, 1991) the concept of a metapolicy was introduced. A metapolicy is a policy about other policies, the rules and assumptions about the policies, and explicitly states the coordination of interaction among policies rather than implicitly leaving such coordination to the administrators. (p. 2). Hosmer's interpretation was that a metapolicy would address how diverse policies could interact across domain boundaries, how data could be updated across domains, and how precedence could be determined and ambiguity removed.

Provisions were made for concurrent support of multiple policies to meet multiple security goals or the needs of different organizations with their own policy intentions; the provision was made for multiple policies. The constraints on support for multiple metapolicies were that each metapolicy had its own:

- Source or owner,
- Enforcement authorities, which could be different from the source, and
- Evolutionary timeframe (Hosmer, 1991, pp. 4-5).

Initially, metapolicies were envisioned as being flexible, potentially layered, tamperproof, and providing a controlling representation of the organization, system, or security policy they represented. In (Hosmer, 1993), the concept of a *multipolicy paradigm* was presented. A key use of multipolicies was for changing circumstances, for example when a country moves from peace to war. (p. 1). The emphasis was on explicit statements of interaction that could continuously enforce the multipolicy intersection and

be formally specifiable and subject to verification of tamper-resistance, the very characteristics Schell presented as desirable for a security kernel architecture.

Bell, and LaPadula, (1976) modeled a multilevel security policy that was implemented in the Multics operating system. This model essentially partitioned the operating system into N-levels of processing, where processing between levels was governed by the Multics security policy. In (Bell D. , 1994) modeling an instance of a "Multipolicy Machine" is presented and 4 levels of abstraction are associated with any given security policy (p. 2):

1. An organization abstraction, written as a narrative, for people to read;
2. A conceptual abstraction, discussing an organizational policy at the concept level;
3. An abstract level, describing the design and tracing the conceptual requirements; and
4. An implementation level, describing the design as developed.

Bell further uses requests, decisions, and state-transition decisions to describe the computational machine model of a multi-policy system component.

(Baskerville, 2002) and (Hafmann & Kuhnhauser, 1999) addressed the concept of an information security meta-policy for an organization, and the characteristics of such a meta-policy. Security is considered a facilitating capability, not a hindrance, and there is recognition that access control policies change over time. Meta-policies, in this discussion, must possess the characteristic of political simplicity, and be criterion-oriented: that is, they must be comprehensible and produce a measurable result. (p. 341). In essence, these meta-policies require explicit statements enumerating the subject's accessing data objects, and the rules for access that will be enforced.

(Hafmann & Kuhnhauser, 1999) demonstrated that the multi-policy concept could be implemented in a Distributed Computing Environment (DCE) in software, with a collection of small software security components that were used to enforce policy separation and, persistency, mediation, and policy domains.

Finally, (MacGraw, 2009) defines the concept of Risk Adaptable Access Control (RADAC). In RADAC, the context of the information usage and the potential risk of unauthorized disclosure are incorporated into the access control decision. A conceptual architecture for RADAC is illustrated if Figure 6.

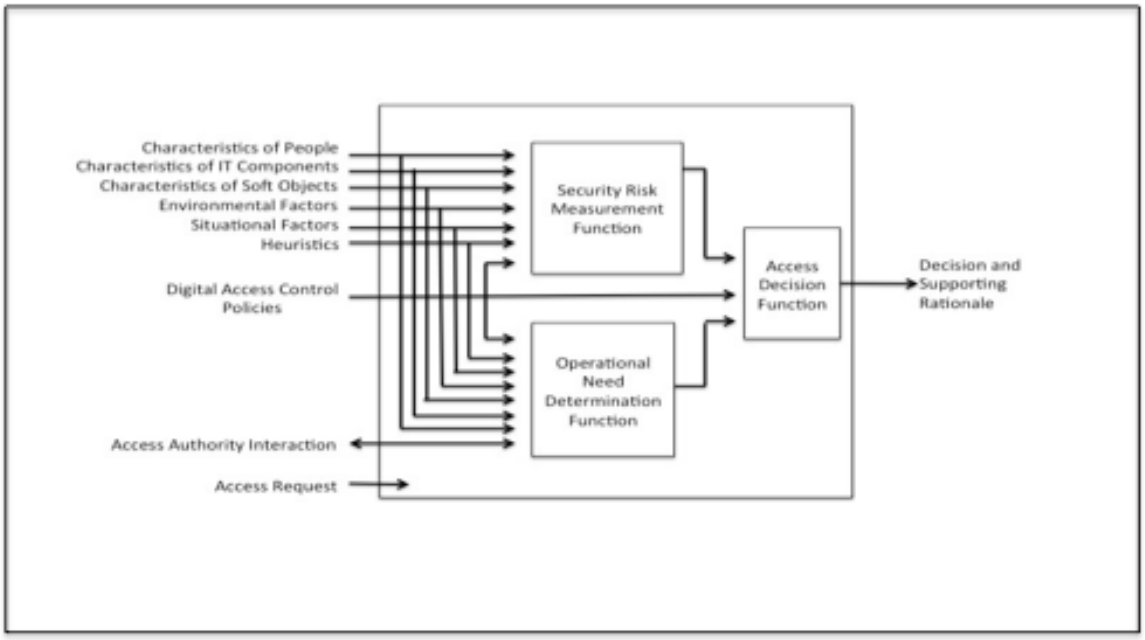


Figure 6. Architecture for Risk Adaptable Access Control (Government, Global Information Grid Information Assurance Reference Capabilities Document, 2004).

Note the inputs to the Security Risk Measurement and Operational Need Determination Functions are included in the access decision. However, little research has been conducted to date on how this information will be gathered or placed in an actionable

format. The use of narrative to solicit this information may present the most viable near-term solution for data collection and integration into the access decision function.

BARRIERS AND ISSUES

This section discusses the barriers and issues associated with security policy definition and integration into the requirements and system design process.

Definition of Context

The dictionary definition of the word "context" is the "circumstances or events that form the environment within which something exists or takes place (Oxford English Dictionary). Describing the general context of an application would be an infinite problem, as there are always new observations or attributes to incorporate into the context. In (Covington, Srinivasan, Dey, Ahamad, & Abowd, 2001) the environmental roles are defined as the security relevant aspects of the environment. The question then becomes, which aspects of the environment are security relevant. These are contingent on the various interpretations of the environment that each application uses for its access control decisions. Covington further emphasizes that environmental roles are used to maintain uniformity across a diverse environment. (p. 12). Further, those devices that monitor the environmental conditions, the sensors, must be authenticated and the integrity of their data guaranteed, or the environmental policy components could be compromised.

(Strembeck, 2004) states: "every goal and obstacle can be used to define a context condition and can map to a concrete access control service." (p. 400). It becomes

necessary, then, to have an environmental model in mind prior to exploring a context-based security policy.

Conflicting Policies

(Wang & Livny, 2004) discuss the issue of policy reconciliation in heterogeneous environments. The notion of a reconciliation algorithm is introduced to find a security policy that consistently adheres to the security policies of all participating domains (p. 1). Wang's model applies acyclic graph theory to model the security mechanisms employed by various environments to provide a framework for policy an analysis. Further, the use of acyclic graph theory exposes commonalities in policy and countermeasures to provide an efficient reconciliation method (linear in size v. N-P-complete).

The computing landscape has matured to the point where basic security mechanisms exist in most system architectures today. That is, there is some authentication, access control, and domain separation supported by the majority of operating systems; what differs is the degree of robustness and strength supported. With the maturation of the security mechanisms, it becomes more feasible to define a structured characterization of a system security environment using policy reconciliation.

The Existence of Supporting Modeling Tools and Concepts

(Jaeger, 2001) discusses the concept of safety in access control models. A *safe access control model* is one in which a given access control will not inadvertently leak access rights to unauthorized persons. Safe models require restrictive security policies,

namely policies that apply constant values as constraints, because variable constraint-based policies are difficult to administer. (p. 158). Jaeger also applies graph theory to design comprehensible security policies.

(Bertino, Ferrari, & Perlasca, 2001) presents a framework for logical reasoning about access control models. In this framework, access control models are modeled in the C-datalog language to develop a common basis for comparison.

The Existence of More Robust Security Models

In the early history of security technologies, security models reflected either mandatory access controls (MAC) or discretionary access controls (DAC) as a security mechanism. More flexible models such as RBAC were nonexistent in the formal sense. The last 10 years have brought the concepts of Usage-controlled models (UCON) (Sandhu R. , 2004) Type Enforcement, and other security models that provide a more granular model of access control interactions.

For example, in Sandhu's UCON model, the traditional lattice-based access control models are used in conjunction with a policy-based authorization management infrastructure. In the past, access decisions were binary; validated on an as-needed basis, with the access maintained for the life of the session. The UCON model allows access rights to change during the life of a session, treating access as a consumable, specifiable event that can exist for a single object access or all attempted object access instances within a session (p. 1).

The Acceptance of Artificial Intelligence Tools to Facilitate Metapolicy Modeling

The existence of artificial intelligence tools that offered data reduction and the ability to replicate results consistently has also been a relatively recent development. Artificial intelligence techniques were not considered reliable – the results depended upon the interpretation of the analysis and of the input data sets. As a result, the technology was dismissed by all but the formal specification community.

In the mid-1990's the Defense Advanced Research Project Agency (DARPA) instituted a new set of initiatives to improve the information assurance tool environment. This initiative led to several developments in data visualization and intrusion detection technologies. It also brought an increased acceptance of artificial intelligence tools and techniques to the information assurance community.

For example, the use of directed acyclic graph theory to model RBAC and RBAC constraints lends itself nicely to the use of Bayesian belief networks (Lueger). In this case, a security policy structure can be represented as a graph, and the nodes of the graph assigned probabilities commensurate with the potential for policy violation.

Similarly, (Lin, 2000) uses information tables to represent policy conflict analysis among multiple security policies. The number of policies, and the number of issues that can be potentially associated with each policy, lend themselves to the use of information tables. The use of decision tree analysis techniques would potentially accommodate the analysis of finer granularity security policies.

The Realization that Policy Models are Contextual in Nature but must have a Formal Foundation.

It is not sufficient to state a security policy without having a degree of formalism attached to it. System consumers usually have no desire to learn or understand the formal methods required to generate a sound security system. As succinctly stated in (Bell, 2005):

> "The Bell-La Padula Model demonstrated the importance of a clear definition of the "security" being addressed. Without a clear definition, one faces unending complaints about "essential" aspects of security being omitted. How can you call a system secure if it doesn't prohibit (or require) N?"

Formal policy foundation mechanisms have not been readily available. Design languages that can address most aspects of a system's context have only begun emerging in the research community. (Alexander, 2006) describes the use of Rosetta as a system level design language, as opposed to a software component level design language. Rosetta addresses the systems engineering specialties such as reliability, maintainability, and latency. In such a model, there is no reason why security cannot be addressed as the specialty engineering discipline it is.

ASSUMPTIONS, LIMITATIONS, AND DELIMITATIONS

This section of the dissertation discusses the context of the study.

Assumptions

(Edwards, 2005) states that the tradeoff between structure and unstructured representations exists for many types of information, but is especially problematic for contextual data for the following reasons:

- Context represents information about people that is very often ambiguous by nature, subtle in its interpretation, and can be applied to many uses.

- There is a great range of information about humans that is potentially useful (ranging from general information about a user's location or actions, to domain-dependent information such as a user's context in a specific application).
- Different sorts of context are important to different applications.

As such, we are assuming we can structure contextual data sufficiently for use within the field of the study of security policy.

Limitations

The primary limitation of this research is the selection of an appropriate enterprise level application with a sufficiently robust security context. An example with a simplistic policy will be successful, but may yield results only marginally better than a traditional digital access control policy. On the other hand, an example with a highly complex policy may prove too difficult to model, or result in a narrative that is so complex that the casual reader would prefer formal methods as a more comprehensible approach.

Delimitations

To address all the factors associated with a meta-policy would be beyond the scope of this investigation. Rather, the study is limited to the context of access control policies, as defined in NIST 800-53 (Government, Special Publication 800-53 rev. 4, , 2011). Table 1 enumerates the requirements family of access control requirements as enumerated in NIST 800-53. The table presents the families of control requirements, and enumerates them in alphabetical order. The controls are marked as applicable to high, medium, and low robustness systems, based upon the risk appetite deemed acceptable to the system's designated authorization authority. A system with a low robustness level

would have fewer access control requirements with lower levels of assurance associated with them. A system with a high robustness level would have more stringent access control requirements with higher levels of assurance associated with the correct operation of the access controls. NIST 800-53 defines three control baselines for automated information systems, designated as high, medium, and low robustness.

Table 1. Requirements family of access control from NIST 800-53.

Control Number	Control Name *Control Enhancement Name*	Control Baselines		
		Low	Mod	High
AC-1	**Access Control Policy and Procedures**	**X**	**X**	**X**
AC-2	Account Management	X	X	X
AC-2 (1)	Account Management| Automated System Account Management		X	X
AC-2 (2)	Account Management| Removal of Temporary/Emergency Accounts		X	X
AC-2 (3)	Account Management| Disable Inactive Accounts		X	X
AC-2 (4)	Account Management| Automated Audit Actions		X	X
AC-2 (5)	Account Management| Inactivity Logout/Typical Usage Monitoring			X
AC-2 (6)	Account Management| Dynamic Privilege Management			
AC-2 (7)	Account Management| Role-Based Schemes			
AC-2 (8)	Account Management| Dynamic Account Creation			
AC-2 (9)	Account Management| Restrictions on Use of Shared Groups/Accounts			
AC-2 (10)	Account Management| Shared/Group Account Requests/Approvals			
AC-2 (11)	Account Management| Shared/Group Account Credential Renewals			
AC-2 (12)	Account Management| Usage Conditions			X
AC-2 (13)	Account Management| Account Reviews			X
AC-2 (14)	Account Management| Account Monitoring/Atypical Usage			
AC-2 (15)	Account Management| Disable Accounts of High-Risk Individuals			
AC-3	**Access Enforcement**	**X**	**X**	**X**
AC-3 (1)	Access Enforcement| Restricted Access to Privileged Functions	Incorporated into AC-6		
AC-3 (2)	Access Enforcement| Dual Authorization			
AC-3 (3)	Access Enforcement| Nondiscretionary Access Control			
AC-3 (4)	Access Enforcement| Discretionary Access Control			
AC-3 (5)	Access Enforcement| Security-Relevant Information			
AC-3 (6)	Access Enforcement| Protection of User and System Information	Incorporated into MP-4 and SC-28		
AC-3 (7)	Access Enforcement| Mandatory Access Control			
AC-3 (8)	Access Enforcement| Role-Based Access Control			
AC-3 (9)	Access Enforcement| Revocation of Access Authorizations			
AC-3 (10)	Access Enforcement| Network Access Security-Related Functions			
AC-4	**Information Flow Enforcement**		**X**	**X**
AC-4 (1)	Information Flow Enforcement| Object Security Attributes			
AC-4 (2)	Information Flow Enforcement| Processing Domains			
AC-4 (3)	Information Flow Enforcement| Condition/Operational Changes			
AC-4 (4)	Information Flow Enforcement| Content Check Encrypted Data			
AC-4 (5)	Information Flow Enforcement| Embedded Data Types			
AC-4 (6)	Information Flow Enforcement| Metadata			
AC-4 (7)	Information Flow Enforcement| One-Way Flow Mechanisms			
AC-4 (8)	Information Flow Enforcement| Security Policy Filters			

Table 1. Requirements family of access control from NIST 800-53 (cont.).

Control Number	Control Name *Control Enhancement Name*	Control Baselines		
		Low	Mod	High
AC-4 (9)	Information Flow Enforcement| Human Reviews			
AC-4 (10)	Information Flow Enforcement| Enable/Disable Security Policy Filters			
AC-4 (11)	Information Flow Enforcement| Configuration of Security Policy Filters			
AC-4 (12)	Information Flow Enforcement| Data Types Identifiers			
AC-4 (13)	Information Flow Enforcement| Decomposition into Policy-Relevant Subcomponents			
AC-4 (14)	Information Flow Enforcement| Policy Filter Constraints on Data Structures and Content			
AC-4 (15)	Information Flow Enforcement| Detection of Unsanctioned Information			
AC-4 (16)	Information Flow Enforcement| Information Transfers on Interconnected Systems			
AC-4 (17)	Information Flow Enforcement| Domain Authentication			
AC-4 (18)	Information Flow Enforcement| Security Attribute Binding			
AC-4 (19)	Information Flow Enforcement| Protection of Metadata			
AC-4 (20)	Information Flow Enforcement| Classified Information			
AC-4 (21)	Information Flow Enforcement| Physical/Logical Separation of Information Flows			
AC-5	**Separation of Duties**		**X**	**X**
AC-6	**Least Privilege**		**X**	**X**
AC-6 (1)	Least Privilege| Authorize Access to Security Functions		X	X
AC-6 (2)	Least Privilege| Non-Privileged Access for Nonsecurity Functions		X	X
AC-6 (3)	Least Privilege| Network Access to Privileged Commands		**X**	**X**
AC-6 (4)	Least Privilege| Separate Processing Domains			
AC-6 (5)	Least Privilege| Privileged Accounts		X	X
AC-6 (6)	Least Privilege| Privileged Access by Non-Organizational Users			
AC-6 (7)	Least Privilege| Review of User Privileges			
AC-6 (8)	Least Privilege| Privilege Levels for Code Execution			
AC-7	**Unsuccessful Login Attempts**	**X**	**X**	**X**
AC-7 (1)	Unsuccessful Login Attempts| Automatic Account Lock	Incorporated into AC-7		
AC-7 (2)	Unsuccessful Login Attempts| Purge Mobile Device			
AC-8	**System Use Notification**	**X**	**X**	**X**
AC-9	**Previous Logon (Access) Notification**			
AC-9 (1)	Previous Logon Notification| Unsuccessful Logons			
AC-9 (2)	Previous Logon Notification| Successful/Unsuccessful Logons			
AC-9 (3)	Previous Logon Notification| Notification of Account Changes			
AC-9 (4)	Previous Logon Notification| Additional Logon Information			
AC-10	**Concurrent Session Control**			**X**
AC-11	**Session Lock**		**X**	**X**
AC-11 (1)	Session Lock| Pattern Hiding Displays	Incorporated into AC-11		

Table 1. Requirements family of access control from NIST 800-53 (cont.).

Control Number	Control Name *Control Enhancement Name*	Control Baselines		
		Low	**Mod**	**High**
AC-12	**Session Termination**	Incorporated into SC-10		
AC-13	**Supervision and Review – Access Control**	Incorporated into AC-2 and AU-6		
AC-14	**Permitted Actions without Identification or Authentication**	X	X	X
AC-14 (1)	Permitted Actions without Identification or Authentication| Necessary Uses	Incorporated into AC-14		
AC-15	**Automated Marking**	Incorporated into MP-3		
AC-16	**Security Attributes**			
AC-16 (1)	Security Attributes| Dynamic Attribute Association			
AC-16 (2)	Security Attributes| Attribute Value Changes by Authorized Individuals			
AC-16 (3)	Security Attributes| Maintenance of Attribute Associations by Information Systems			
AC-16 (4)	Security Attributes| Association of Attributes by Authorized Individuals			
AC-16 (5)	Security Attributes| Attribute Displays for Output Devices			
AC-16 (6)	Security Attributes| Maintenance of Attribute Association by Organization			
AC-16 (7)	Security Attributes| Consistent Attribute Interpretation			
AC-16 (8)	Security Attributes| Association Techniques/Technologies			
AC-16 (9)	Security Attributes| Attribute Reassignment			
AC-16 (10)	Security Attributes| Attribute Configuration by Authorized Individuals			
AC-16 (11)	Security Attributes| Permitted Attributes for Specified Information Systems			
AC-16 (12)	Security Attributes| Permitted Values and Ranges for Attributes			
AC-17	**Remote Access**	X	X	X
AC-17 (1)	Remote Access| Automated Monitoring/Control		X	X
AC-17 (2)	Remote Access| Protection of Confidentiality/Integrity Using Encryption		X	X
AC-17 (3)	Remote Access| Managed Access Control Rights		X	X
AC-17 (4)	Remote Access| Privileged Commands/Access		X	X
AC-17 (5)	Remote Access| Monitoring for Unauthorized Connections	Incorporated into AC-17		
AC-17 (6)	Remote Access| Protection of Information			
AC-17 (7)	Remote Access| Additional Protection for Security Function Access	Incorporated into AC-3		
AC-17 (8)	Remote Access| Disable Nonsecure Network Protocols	Incorporated into CM-7		
AC-17 (9)	Remote Access| Disconnect/Disable Access			
AC-18	**Wireless Access**	**X**	**X**	**X**
AC-18 (1)	Wireless Access| Authentication and Encryption		X	X
AC-18 (2)	Wireless Access| Monitoring Unauthorized Connections	Incorporated into AC-18		
AC-18 (3)	Wireless Access| Disable Wireless Networking			

Table 1. Requirements family of access control from NIST 800-53(cont.).

Control Number	Control Name *Control Enhancement Name*	Control Baselines		
		Low	Mod	High
AC-18 (4)	Wireless Access| Restrict Configurations by Users			X
AC-18 (5)	Wireless Access| Confine Wireless Communications			X
AC-19	**Access Control for Mobile Devices**	X	X	X
AC-19 (1)	Access Control for Mobile Devices| Use of Writable/Removable Media	Incorporated into MP-7		
AC-19 (2)	Access Control for Mobile Devices| Use of Personally Owned Removable Media	Incorporated into MP-7		
AC-19 (3)	Access Control for Mobile Devices| Use of Removable Media with No Identifiable Owner	Incorporated into MP-7		
AC-19 (4)	Access Control for Mobile Devices| Restrictions for Classified Information			
AC-19 (5)	Access Control for Mobile Devices| Personally Owned Devices			
AC-19 (6)	Access Control for Mobile Devices| Full Disk Encryption		X	X
AC-19 (7)	Access Control for Mobile Devices| Central Management of Mobile Devices			
AC-19 (8)	Access Control for Mobile Devices| Remote Purging of Information			
AC-19 (9)	Access Control for Mobile Devices| Tamper Detection			
AC-20	**Use of External Information Systems**	X	X	X
AC-20 (1)	Use of External Information Systems| Limits on Authorized Use		X	X
AC-20 (2)	Use of External Information Systems| Portable Storage Media		X	X
AC-20 (3)	Use of External Information Systems| Personally Owned Information Systems/Devices			
AC-20 (4)	Use of External Information Systems| Network Accessible Storage Devices			
AC-21	**Collaboration and Information Sharing**		X	X
AC-21 (1)	Collaboration and Information Sharing| Automated Decision Support			
AC-21 (2)	Collaboration and Information Sharing| Information Search and Retrieval			
AC-22	**Publicly Accessible Content**	X	X	X
AC-23	**Data Mining Protection**			
AC-24	**Access Control Decisions**			
AC-24 (1)	Access Control Decisions| Transmit Access Authorization Information			
AC-24 (2)	Access Control Decisions| No User or Process Identity			
AC-25	**Reference Monitor Function**			

DEFINITION OF TERMS

The following terms are defined in this section with the origin of the definition in parenthesis after the term.

Adequate Security (OMB Circular A-130, Appendix III)	Security commensurate with the risk and magnitude of harm resulting from the loss, misuse, or unauthorized access to or modification of information.
Assurance (CNSSI 4009)	Measure of confidence that the security features, practices, procedures and architecture of an information system accurately mediate and enforce the security policy.
Attribute-Based Access Control (ABAC) (NIST 800-53)	Access control based on attributes associated with and about subjects, objects, targets, initiators, resources, or the environment. An access control rule set defines the combination of attributes under which an access may take place.
Audit Log (CNSSI 4009)	A chronological record of information system activities, including records of system accesses and operations performed in a given period.
Authentication (FIPS 200)	Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.
Availability (44 U.S. C., Sec. 3542)	Ensuring timely and reliable access to and use of information.
Classified Information (NIST 800-53)	Information that has been determined:(i) pursuant to Executive Order 12958 as amended by Executive Order 13292, or any predecessor Order, to be classified national security information; or (ii) pursuant to the Atomic Energy Act of 1954, as amended, to be Restricted Data (RD).

Baseline Configuration (NIST 800-53)	A documented set of specifications for an information system, or a configuration item within a system, that has been formally reviewed and agreed on at a given point in time, and which can be changed only through change control procedures. The baseline configuration is used as a basis for future builds, releases, and/or changes.
Confidentiality (44 U.S.C., Sec. 3542)	Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.
Controlled Unclassified Data (E.O. 13556)	A categorical designation that refers to unclassified information that does not meet the standards for National Security Classification under Executive Order 12958, as amended, but is (i) pertinent to the national interests of the United States or to the important interests of entities outside the federal government, and (ii) under law or policy requires protection from unauthorized disclosure, special handling safeguards, or prescribed limits on exchange or dissemination. Henceforth, the designation CUI replaces Sensitive But Unclassified (SBU)
Countermeasures (CNSSI 4009)	Actions, devices, procedures, techniques, or other measures that reduce the vulnerability of an information system. Synonymous with security controls and safeguards.
Cyber Attack (CNSSI 4009)	An attack, via cyberspace, targeting an enterprise's use of cyberspace for the purpose of disrupting, disabling, destroying, or maliciously controlling a computing environment/infrastructure; or destroying the integrity of the data or stealing controlled information.
Cyber Security (CNSSI 4009)	The ability to protect or defend the use of cyberspace from cyber attacks.

Cyberspace (CNSSI 4009)	A global domain within the information environment consisting of the interdependent network of information systems infrastructures including the Internet, telecommunications networks, computer systems, and embedded processors and controllers.
Defense-in-Depth (NIST 800-53)	Information security strategy integrating people, technology, and operations capabilities to establish variable barriers across multiple layers and missions of the organization.
Discretionary Access Control (NIST 800-53)	A type of access control that restricts access to objects based on the identity of the subjects or groups to which subjects belong. The access controls are discretionary because subjects with certain privileges are capable of passing those privileges on to any other subjects, either directly or indirectly. Nondiscretionary access controls restrict this capability.
Domain (CNSSI 4009)	An environment or context that includes a set of system resources and a set of system entities that have the right to access the resources as defined by a common security policy, security model, or security architecture. See *Security Domain*.
Enterprise (CNSSI 4009)	An organization with a defined mission/goal and a defined boundary, using information systems to execute that mission, and with responsibility for managing its own risks and performance. An enterprise may consist of all or some of the following business aspects: acquisition, program management, financial management (e.g., budgets), human resources, security and information systems, information, and mission management.

Enterprise Architecture (CNSSI 4009)	The description of an enterprise's entire set of information systems: how they are integrated, how they interface to the external environment at the enterprise's boundary, how they are operated to support the enterprise mission, and how they contribute to the enterprise's overall security posture.
Identity-Based Access Control (NIST 800-53	Access control based on the identity of the user (typically relayed as a characteristic of the process acting on behalf of that user) where access authorizations to specific objects are assigned based on the user's identity.
Information (CNSSI 4009)	Any communication or representation of knowledge such as facts, data, or opinions in any medium or form, including textual, numerical, graphic, cartographic, narrative, or audiovisual.
(FIPS 199)	An instance of an information type.
Information Owner (CNSSI 4009)	Official with statutory or operational authority for specified information and responsibility for establishing the controls for its generation, collection, processing, dissemination, and disposal.
Information Resources (44 U.S.C., Sec. 3502)	Information and related resources, such as personnel, equipment, funds, and information technology.
Information Security (44 U.S.C., Sec 3542)	The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.
Information Security Architecture (NIST 800-53)	An embedded, integral part of the enterprise architecture that describes the structure and behavior for an enterprise's security processes, information security systems,

	personnel and organizational sub-units, showing their alignment with the enterprise's mission and strategic plans.
Information Security Policy (CNSSI 4009)	Aggregate of directives, regulations, rules, and practices that prescribes how an organization manages, protects, and distributes information.
Information System (44 U.S.C., Sec. 3502)	A discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information. (Note: Information systems also include specialized systems such as industrial/process control systems, telephone switching and private branch exchange (PBX) systems, and environmental control systems.)
Information Security Risk (NIST 800-53)	The risk to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the Nation due to the potential for unauthorized access, use, disclosure, disruption, modification, or destruction of information and/or information systems.
Information System-Related Security Risks (NIST 800-53)	Risks that arise through the loss of confidentiality, integrity, or availability of information or information systems and consider impacts to the organization (including assets, mission, functions, image or reputation), individuals, other organizations and the Nation. See *Risk*.
Information Type (FIPS 199)	A specific category of information (e.g., privacy, medical, proprietary, financial, investigative, contractor sensitive, security management) defined by an organization or in some instances by a specific law, Executive Order, directive, policy, or regulation.

Integrity (44 U.S.C., Sec 3542)	Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity.
Label (NIST 800-53)	See *Security Label*.
Local Access (NIST 800-53)	Access to an organizational information system by a user (or process acting on behalf of a user) communicating through a direct connection without the use of a network.
Malicious Code (NIST 800-53)	Software or firmware intended to perform an unauthorized process that will have adverse impact on the confidentiality, integrity, or availability of an information system. A virus, worm, Trojan horse, or other code-based entity that infects a host. Spyware and some forms of adware are also examples of malicious code.
Marking (NIST 800-53)	*See Security Marking*.
Media (FIPS 200)	Physical devices or writing surfaces including, but not limited to, magnetic tapes, optical disks, magnetic disks, Large-Scale Integration (LSI) memory chips, and printouts (but not including display media) onto which information is recorded, stored, or printed within an information system.
Metadata (NIST 800-53)	Information describing the characteristics of data including, for example, structural metadata describing data structures (e.g., data format, syntax, and semantics) and descriptive metadata describing data contents (e.g., information security labels).
Network (CNSSI 4009)	Information system(s) implemented with a collection of interconnected components. Such components may include routers, hubs, cabling, telecommunications controllers, key distribution centers, and technical control devices.

Network Access (NIST 800-53)	Access to an information system by a user (or a process acting on behalf of a user) communicating through a network (e.g., local area network, wide area network, Internet).
Nondiscretionary Access Control (NIST 800-53)	A type of access control that restricts access to objects based on the identity of subjects or groups to which the subjects belong. The access controls are nondiscretionary because subjects with certain privileges are restricted from passing those privileges on to any other subjects, either directly or indirectly – that is, the information system strictly enforces the access control policy based on the rule set established by the policy.
Object (NIST 800-53)	Passive information system-related entity (e.g. devices, files, records, tables, processes, programs, domains) containing or receiving information. Access to an object (by a subject) implies access to the information it contains. See *Subject*.
Organization (FIPS 200)	An entity of any size, complexity, or positioning within an organizational structure (e.g., a federal agency or, as appropriate, any of its operational elements.
Organizational User (NIST 800-53)	An organizational employee or an individual the organization deems to have equivalent status of an employee (e.g., contractor, guest researcher, individual detailed from another organization, individual from an allied nation).
Privileged Account (NIST 800-53)	An information system account with the authorizations of a privileged user.
Privileged User (CNSSI 4009)	A user that is authorized (and therefore, trusted) to perform security-relevant functions that ordinary users are not authorized to perform.

Remote Access (NIST 800-53)	Access to an organizational information system by a user (or a process acting on the behalf of a user) communicating through an external network (e.g., the Internet).
Risk (FIPS 200)	A measure of the extent to which an entity is threated by a potential circumstance or event, and typically a function of (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence. Information system-related security risks are those risks that arise from the loss of confidentiality, integrity, or availability of information or information systems and reflect the potential adverse impacts to organizational operations (including mission, functions, image, or reputation), organizational assets, individuals, other organizations, and the Nation.
Risk Assessment (NIST 800-53)	The process of identifying risks to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the Nation, resulting from the operation of an information system. Part of risk management incorporates threat and vulnerability analyses, and considers mitigations provided by security controls planned or in place. Synonymous with risk analysis.
Risk Management (CNSSI 4009)	The program and supporting processes to manage information security risk to organizational operations (including mission, functions, image, reputation), organizational assets, individuals, other organizations, and the Nation, and includes: (i) establishing the context for risk related activities; (ii) assessing risk; (iii) responding to risk once determined; and (iv) monitoring risk over time.

Role-Based Access Control (NIST 800-53)	Access control based on user roles (i.e., a collection of access authorizations a user receives based on an explicit or implicit assumption of a given role). Role permissions may be inherited through a role hierarchy and typically reflect the permissions needed to perform defined functions within an organization. A given role may apply to a single individual or to several individuals.
Security (CNSSI 4009)	A condition that results from the establishment and maintenance of protective measures that enable an enterprise to perform its mission or critical functions despite risks posed by threats to its use of information systems. Protective measures may involve a combination of deterrence, avoidance, prevention, detection, recovery, and correct that should form part of the enterprise's risk management approach.
Security Assurance (NIST 800-53)	See *Assurance*.
Security Attribute (NIST 800-53)	An abstraction representing the basic properties or characteristics of an entity with respect to safeguarding information; typically associated with internal data structures (e.g., records, buffers, files), within the information system and used to enable the implementation of access control and flow control policies, reflect special dissemination, handling, or distribution instructions, or support other aspects of the information security policy.
Security Domain (CNSSI 4009)	A domain that implements a security policy and is administered by a single authority.
Security Functions (NIST 800-53)	The hardware, software, and/or firmware of the information system responsible for enforcing the system security policy and supporting the isolation of code and data on which the protection is based.

Security Label (NIST 800-53)	The means used to associate a set of security attributes with a specific information object as part of the data structure for that object.
Security Marking (NIST 800-53)	Human-readable information affixed to information system components, removable media, or output indicating the distribution limitations, handling caveats, and applicable security markings.
Security Policy (CNSSI 4009)	A set of criteria for the provision of security services.
Security Requirements (FIPS 200)	Requirements levied on an information system that are derived from applicable laws, Executive Orders, directives, policies, standards, instructions, regulations, procedures, or organizational mission/business case needs to ensure the confidentiality, integrity, and availability of the information being processed, stored, or transmitted.
Security-Relevant Information	Any information within the information system that can potentially impact the operation of the security functions or the provision of security services in a manner that could result in failure to enforce the system security policy or maintain isolation of code and data.
Subject (NIST 800-53)	Generally an individual, process, or device causing information to flow among objects or change the system state. See *Object*.
Subsystem (NIST 800-53)	A major subdivision or component of an information system consisting of information, information technology, and personnel that performs one or more specific functions.
System (NIST 800-53)	See *Information System*.

Threat (CNSSI 4009)	Any circumstance or event with the potential to adversely impact organizational operations (including mission, functions, image, or reputation, organizational assets, individuals, other organizations, or the Nation through an information system via unauthorized access, destruction, disclosure, modification of information, and/or denial of service.
Trustworthiness (of an Information System) (NIST 800-53)	The degree to which an information system (including the information technology components that are used to build the system) can be expected to preserve the confidentiality, integrity, and availability of the information being processed, stored, or transmitted by the system across the full range of threats. A trustworthy information system is a system that is believed to be capable of operating within defined levels of risk despite the environmental disruptions, human errors, structural failures, and purposeful attacks that are expected to occur in its environment of operation.
User (CNSSI 4009)	Individual, or (system) process acting on behalf of an individual, authorized to access an information system.
Vulnerability (CNSSI 4009)	Weakness in an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.

SUMMARY

In conclusion, we have presented the case for a more comprehensible system security policy definition; one that could be used to elicit information required for various security policy models and support a degree of formalism that would be amenable to formal methods and logical proof if desired. More significantly, this policy elicitation through the use of structured storytelling would be a less intimidating, more descriptive technique that would lend itself to contextual security information such as that required for risk adaptive access control models.

Chapter 2

Review of the Literature

This section describes the relevant research literature associated with this proposal. For completeness and clarity, this chapter addresses four distinct fields of research:

- Security policies and modeling,
- The use of artificial intelligence techniques,
- The use of storytelling as a design tool, and
- Computer assisted storytelling techniques.

The combination of these fields of study forms the basis for the research.

Security Policies and Modeling

In its most basic form, access control prevents unauthorized use of resources (Ferraiolo, D., Barkley, J.F., & Kuhn, D.R., 1999). It involves an *access controller* that grants or denies the request of a *subject* to perform an *operation* on an *object* according to the *access control policy*. The *subject* identifies an entity and its accompanying attributes. The *operation* makes information flow to or from the *object*: It is either read or written to system resources. The *operation* includes *access* and the accompanying activities of collection, storage, processing and distribution of information. The *access policy* specifies the usage rights of the *subject* to perform the *operation* on the *object*. The *access controller* executes *subject authentication* and *access authorization*. The *access controller* performs *subject authentication* on the basis of the *Token (T)* and a Subject Identity (S_{ID}). The *access controller* performs *access authorization* by determining the Permission of the Subject to execute the Operation on the Object. Figure 7 illustrates this information flow.

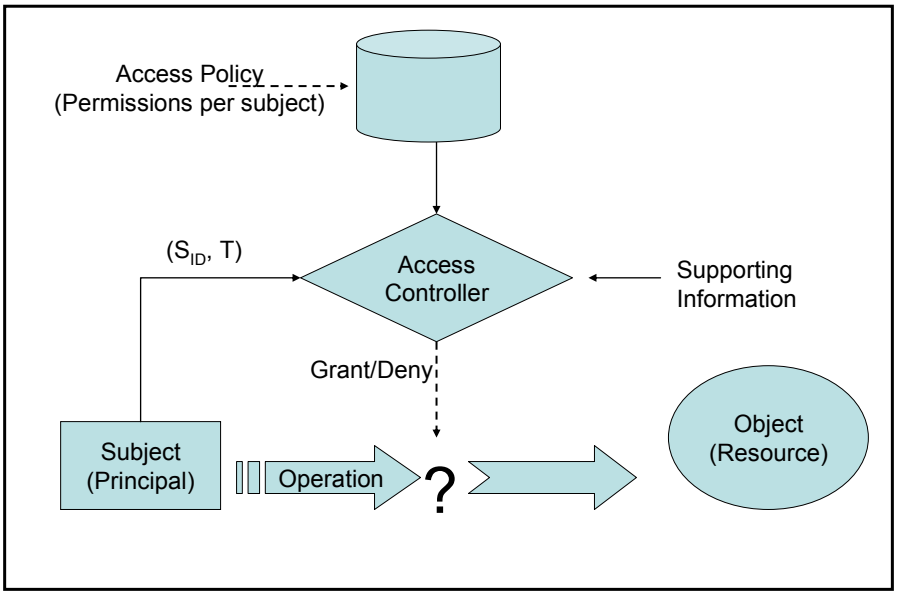


Figure 7 – The access control process (Hulsebosch, et al, 2005).

Access control can be further delineated into real time access controls and non-real time access controls. Figure 8 represents a workflow for non-real-time access control. Figure 9 represents a workflow for real-time access control decisions.

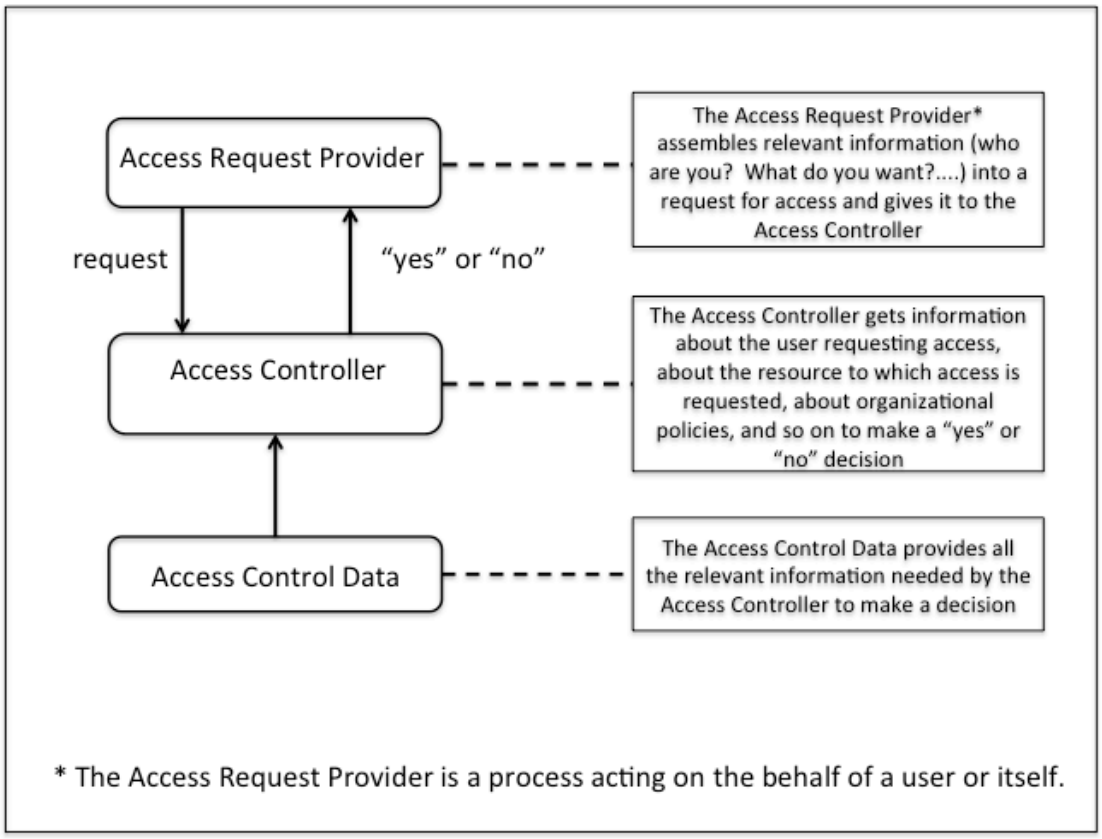


Figure 8. The general process flow of non-real time access control (NIST, 2010, p.3).

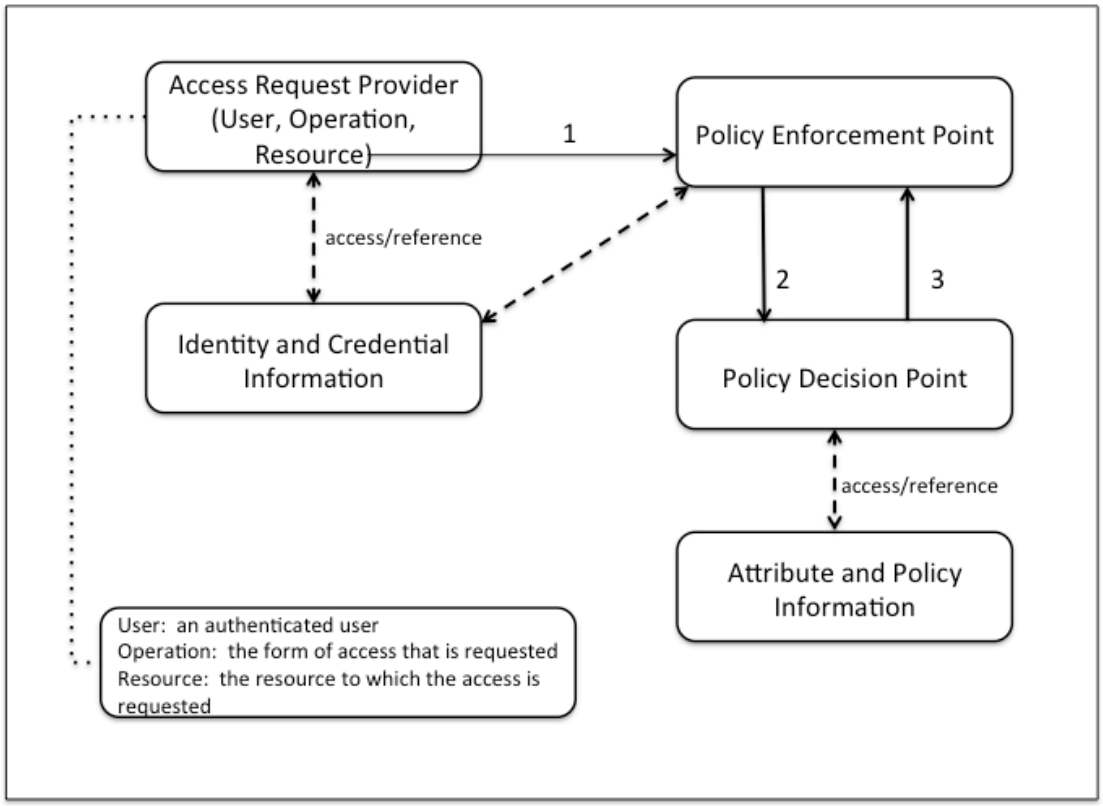


Figure 9. The process flow of real time access control decisions (NIST, 2010, p.3).

The process flow of dynamic access control policies can be generalized as shown in Figure 10. In this model, the activities of policy creation, conflict resolution, promulgation, and enforcement are decoupled steps in the policy distribution process. That is, a policy can be created, but may conflict with other operational needs and require resolution of such conflicting operations prior to distribution and enforcement at all nodes in a system.

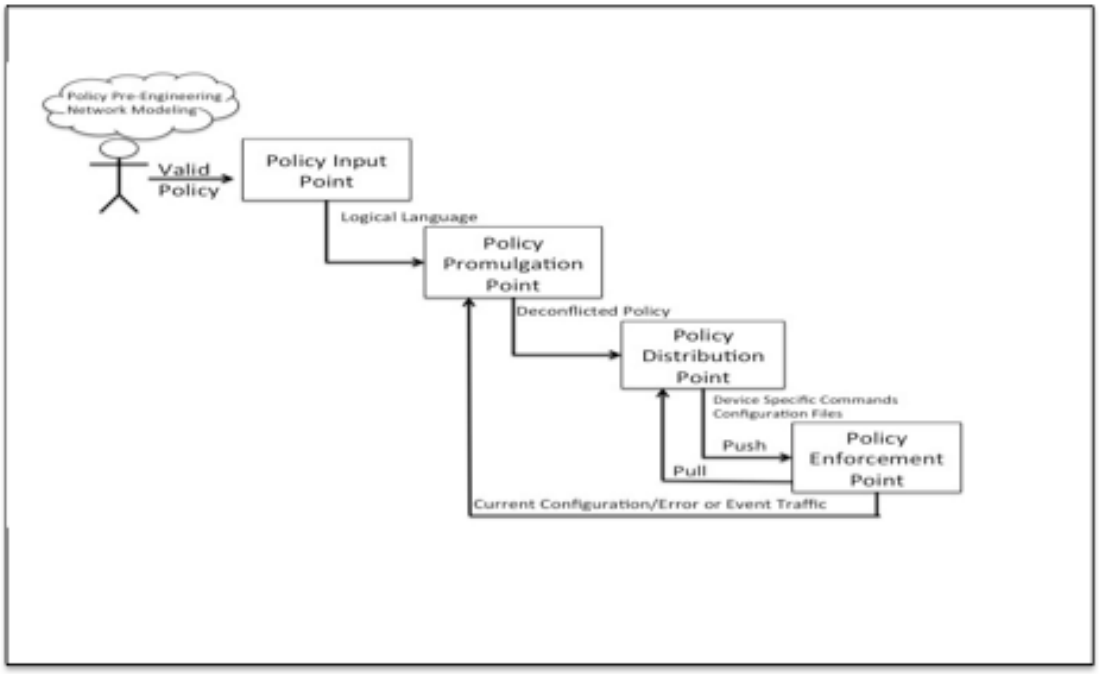


Figure 10. Processing flow to promulgate a dynamic access control policy (U.S. Government, 2004, p.3-25)

(Burnside & Keromytis, 2007) state: traditional access control mechanisms used in enterprise networks operate independently on each service. When a user issues a request to a network service, the service's access-control mechanism independently uses its security policy to make a decision on how to handle the request, and then goes inactive. There may be information relevant to the decision elsewhere in the network, but the decision is made without consulting other network entities, so the component may arrive at a *locally correct, but globally wrong* decision.

Role-based Access Control

As has been stated above, RBAC was designed to better accommodate the actual usage scenarios of the civil government and commercial organizations. Sandhu, Kuhn, and Ferraiolo (Sandhu, Kuhn, & Ferraiolo, 2000) formalized the definition of RBAC, and subsequently RBAC has been defined as an ANSI Standard (NCITS, 2004). RBAC as a

model has been further formalized in (Gligor, 1995) (Park, Neven, & Diosomito, 2004) (Han, 2000) and many others. RBAC has become the accepted implementation of access control in most commercial operating systems. This has several benefits for system administrators, because in most organizations there are well-defined roles that can be institutionalized across the enterprise. (Park, Neven, & Diosomito, 2004) defines three tiers of RBAC constraints within an enterprise: the organizational hierarchy, the enterprise hierarchy, and the system hierarchy, and suggests these hierarchies can be interrelated and reused amongst organizations. Figure 11 illustrates an RBAC implementation.

There has been recent work to extend RBAC to address attribute based access control (ABAC) within the RBAC model (Kuhn, 2010). Attribute based access control is based upon the user possessing a given attribute in his credentials to meet a rule for access to be granted (Karp, 2009). ABAC is easy to establish, but difficult to change, and RBAC requires considerable attention to support sound role creation. (Kuhn, 2010) proposes to apply a role structure to attributes that are relatively static, simplifying ABAC and supporting a more efficient attribute change process.

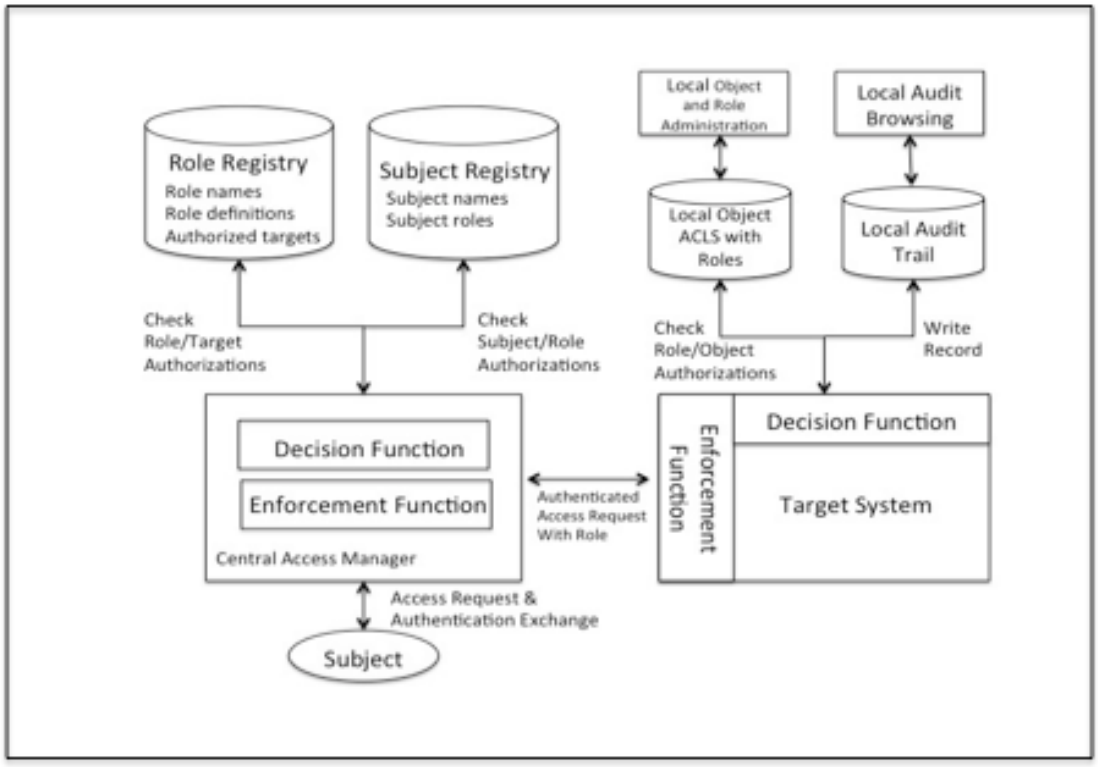


Figure 11 – An architecture for RBAC Implementation (Sherwood, Clark, & Lynas, 2008, p. 241)

Trust-based access control (TrustBAC) was proposed by (Chakraborty S. & Ray, I.., 2006) as a solution to provide increased flexibility for RBAC-type access models. TrustBAC is useful in decentralized models where the user population is dynamic and the identity of all users is not known in advance. TrustBAC uses the authentication credentials of the user to create a binary trust relationship: a user is either trusted or he is not. Trust levels are introduced to address a user's contextual credentials such as the user's behavior history, and reputation. These trust levels are then mapped to user roles and their associated permissions.

Context-based Access Control

With the advent of wireless networks and intelligent devices, ubiquitous computing brought a new collection of access control policy issues. (Hengartner, 2004) discusses the security issues associated with people location information, such as how much information should be shared about a person's location. Device privacy scenarios were introduced, where a person's laptop or PDA may divulge their location without the owner's permission. In this case, access controls were established to the device location services and the person location services, and the information was protected in transit via conventional encryption technology. Ardanga, et.al. (2006) discussed integration of location-based conditions with access controls to accommodate mobile user interaction. In this model, access is specified in terms of position, movement, and interaction of the user. For example, if a user is within a given area, and moving slowly enough, access can be granted.

Spatial Security Policies

Location based services and mobile applications bring unique access control requirements to security policy models. For example, consider the scenario of "If it's my wife, I've just left" when the person is still in the office. There have been numerous location alibis' that have been denounced as the result of electronic toll collection transactions (Klunder). (Bertino, Catania, Damiani, & Perlasca, 2005), define a geospatial framework for RBAC policies that accommodates a spatially-ware access control attribute (p.30). Using the Open Geospatial Consortium (OGC)

spatial model (OpenGIS Consortium, 1999) a boundary perimeter can be associated with a user role, creating a binding between a user's location and the access control policy. For example, if the user is not within the building, he may not be permitted to view proprietary information.

In (Lei, Daby, Davis, Banavar & Ebling), the concept of context awareness was introduced, where an application adapts to the environment to fulfill the needs of the user. Lei discusses providing context awareness as a service to a user's application environment, so the most appropriate presentation methods can be applied for data. For example, a low bandwidth connection may not be suited to a graphics-rich web environment or situational display. The concept of a controller is introduced to specify the user's context, an owner to specify how data is disseminated, and a client or application to collect authorized information. While not an access control policy as such, the specification of a context service provides an illustrative example of how ubiquitous devices apply contextual access management.

Within an RBAC environment, (Strembeck, 2004) proposes that contextual constraints can be considered a special purpose RBAC mechanism. In this model, contextual constraints are subject to dynamic checking against predefined conditions or specified values. Strembeck makes the distinction between static constraints that are specified at constraint establishment against constant values, as opposed to dynamic constraints that are evaluated against specific run-time parameters or variables. (pp. 395-396). The notion of conditional permission is presented and defined as the case of access being granted if and only if each contextual constraint associated with that access evaluates as a true statement. Using context constraints

allows the traditionally static RBAC policy to incorporate dynamic data in its role evaluation processing.

(Covington, Srinivasan, Dey, Ahamad, & Abowd, 2001) also discuss context-based security, introducing the notion of environmental roles. In this presentation, the context is that of a context aware environment, in which the behavior of the applications is tailored based on the user's environmental context. For example, an intelligent entertainment system may increase the volume and change the station on the radio depending upon whether the user is at work or at home. In this model, the environmental state must be semantically represented, and the rules associated with that state captured for environmental context to be used for access management.

Risk Adaptive Access Controls (RAdAC)

Pervasive connectivity does not accommodate static security policy modeling well. To address the dynamic nature of policy management and policy-based access controls, Risk Adaptive Access Control (RAdAC) models have been proposed (McGraw, 2004). These models address security based on the premise that information should be shared by default, as opposed to static need-to-know based models that by default protect information and make it unavailable for use. RAdAC uses the dimensions of security risk and operational need in addition to the classification of the information and the clearance of the user (Choudhary, 2005 ,p. 294). For example, a unit on the front lines has a very strong operational need to know opposing military troop movements, but might be denied that information if not deployed. RAdAC is an early attempt to adapt security policy to digital information

timelines as opposed to paper document models (p.294). RAdAC models make access control decisions based upon the following components (p.295):

- Characteristics of people
- Characteristics of IT Components
- Characteristics of content objects
- Environmental factors
- Situational factors
- Heuristics
- Digital Access Control Policies
- Access Authorization Operator Interaction
- User requests for access to a resource
- Decision history and supporting rationale.

Essentially, various policy elements are maintained in policy information bases (PIBs) (p. 295) and retrieved as required to address access requests. If a request is granted, the information is presented to the user. If a request is denied, subsequent PIBs are consulted, and the disposition of the request is based upon security heuristics or rules engines as well as static attributes associated with the user and the data.

Context Sensitive Access Control

Context sensitive access control policies focus on the situational environment to determine if the user should be granted access to services (Hulsebosch, Salden, Bargh, & Ebben, 2005 p. 111). In this model, the effective security controls of the physical world are used to define access controls. These may be based upon location, velocity, age, device and/or network capabilities, temperature, time of day, and possibly the user's intentions (p. 112). If the patterns of behavior that can be derived from contextual information can be captured and grouped, an effective variant of

role-based access control can be created. In this model, the access controller has to verify the contextual attributes provided by the subject to authenticate the subject's request. The access controller also has to bind the permissions in the access policy to the subject's contextual attributes to perform the access authorization function. While context sensitive access control does require an infrastructure to collect, manage, and interpret contextual information, most of these functions could be performed as background tasks (p. 117). The ability to apply contextual data to access control decisions would provide a more realistic model of access decision making in computing environments.

Usage Based Access control

An advantage to ubiquitous computing environments is transparent access to information without regard for the underlying computing infrastructure (Wang, Zang, & Cao, 2006). Within these environments, the mobility of the users presents challenges in the determination of the user's contextual information and authentication. Usage-based access control models augment traditional access control models with two additional elements:

1. Obligations – requirements that have to be followed by the subject to allow access to resources, and
2. Conditions – subject and object independent requirements that have to be passed to the access controller.

Figure 12 illustrates the components of a usage based access control model.

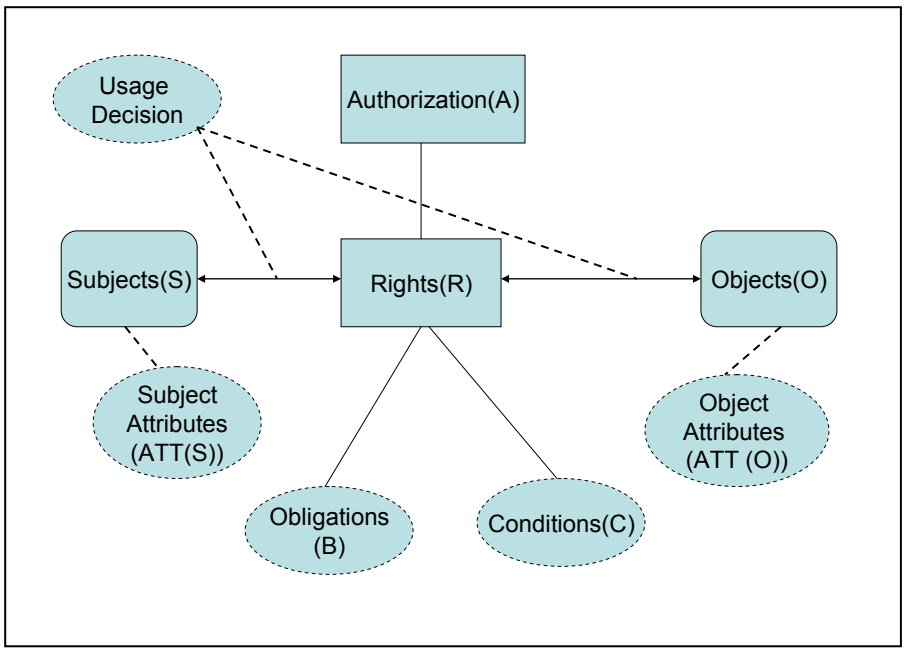


Figure 12 – Usage based access control attributes.

Collaborative Access Control Models

In the Web 3.0 world, collaborative systems become more prevalent. Groups of users communicate and cooperate to address a common task (Tolone, Ahn, & Pai, 2005). Information in collaborative applications may be of varied sensitivity. The collaborative environment allows users to create, manipulate, and provide access to information and resources. Collaboration depends upon making information available to those with a need to know, whereas access control models restrict access to information based upon the user's defined confidentiality, integrity, or availability constraints. In collaborative access control models:

- Access control must be applied and enforced at a distributed platform level.
- Should be expressive enough to specify access rights efficiently based upon varied information.

- Must scale to address the number of shared operations expected in collaborative multi-user environments.
- Support access decisions for resources and information at varying levels of granularity.
- Support transparent access for authorized users and strong exclusion of unauthorized users; yet support unrestrained collaboration.
- Allow high-level specification of access rights, to facilitate complexity management.
- Support dynamic modification at runtime to reflect the environment or collaboration dynamics.

Along with traditional access control models, the concept of Team based access control (Thomas, 1997) is examined to provide integration of user context and object context into the access control space. Figure 13 illustrates TBAC's information flow.

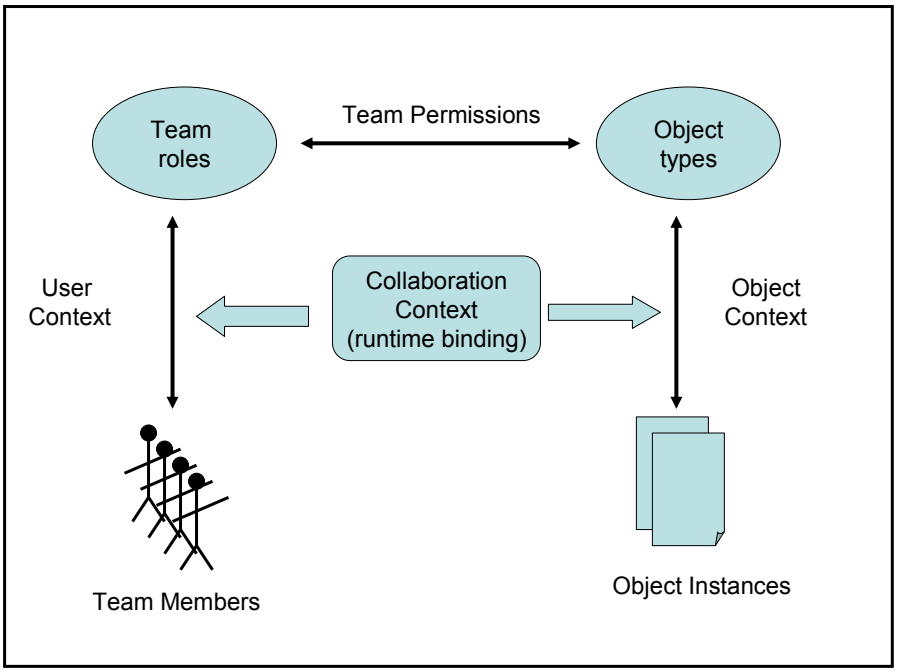


Figure 13 – The Team Based access control model (Tolone, et al, p. 36).

The authors then proceed to contrast and compare existing access control models, searching for the best fit of characteristics that address collaborative architectures. A

context based access control model is suggested that accommodates dynamic user participation and fine grained object models.

Semantic Access Controls

(Pan, Mitra, & Liu, 2006) propose the use of semantically enhanced role-based access control to facilitate database interoperability. Their model incorporates ontology mapping to accommodate semantic heterogeneity in conjunction with confidentiality constraints associated with data sharing among organizations. They contend that preserving access control across semantically heterogeneous information systems is more accurately termed semantic access control. When a query is issued that may cross database boundaries, the roles, tables, and columns are validated against the corresponding roles, tables, and columns of the other databases within an information system. Access controls are translated in real time in response to a subject's query. Figure 14 illustrates the semantic access control model.

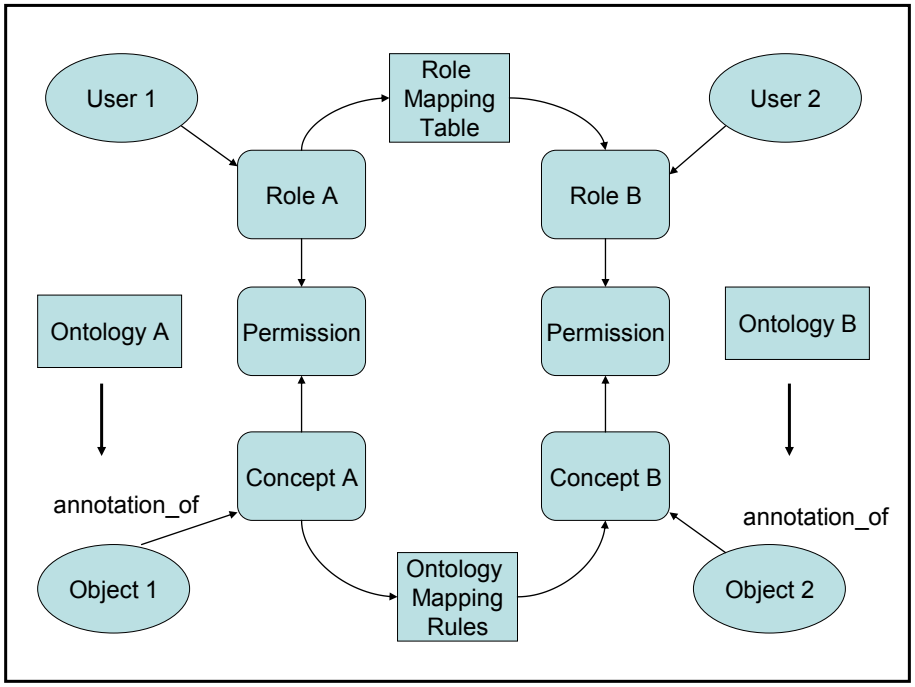


Figure 14 – Information flow in the semantic access control model (Pan, Mitra, & Liu, p. 240).

THE USE OF VARIOUS ARTIFICIAL INTELLIGENCE TECHNIQUES

As is evident from the discussion above, there is no shortage of formal policy models. What is lacking are techniques for integrating the formalism of policy modeling with policy elicitation mechanisms to understand the rules a customer wishes to enforce upon information sharing. In this section, we explore the relevant contributions from the fields of artificial intelligence to this effort.

(Atallah, McDonough, Raskin, & Nirenburg, 2000) presented four areas where natural language processing techniques could benefit the information security community. One of these four areas was the area of ontology and its application to machine translation techniques. This research team had developed a semi-automatically acquired ontology, with a semi-automatically acquired lexicon for over 40,000 words. Further, an analyzer was created to translate text from a natural language into a text-meaning representation and a generator to translate the text

meaning representation into a given natural language. This use of ontological semantics was being applied to the problem of document sanitization or redaction, where some text needs to be removed before a document is approved for public release. Essentially, the workings of a human reviewer were modeled and codified into a rule base that applied Natural Language Processing (NLP) techniques.

Of interest: the authors state that the primary function of natural language is communication among humans (p. 64), the most common mode that of fact-conveying. In this mode, the speaker and the listener are committed to the literal truth of what is said, but each interprets it differently based upon their understanding. In Natural Language Processing, this brings to bear the problems of underspecification, namely vagueness and ambiguity. Simply put, no one spoken sentence can capture all the semantic meaning of a given context. The human mind applies prior knowledge and inferences to fill in the missing data.

Subsequently, (Raskin, Hempelmann, & Triezenberg, 2004) presented an experiment in ontological semantics as a a technique to determine deception. The approach proposed represents words as meanings, including the sentences and their text, and performs logical manipulation; resulting in a system that can identify specific facts or events that contribute to the deception and of understanding what the truth behind the deception is. (Raskin, Hempelmann, Nierenberg, & Triezenberg, 2001) further define a comprehensive ontology for the domain of information assurance, providing a sound foundation for further research in the domain.

(Owen, Wakeman, Keller, Weeds, & Weir, 2005) discuss pervasive computing environments and conclude that non-technical users will want to be able to configure their own devices. In their research, a series of user studies were conducted to determine how people define the term “configuration.” Once the meaning of system configuration was defined, it was possible to determine a

formalism and create a natural language processing system to address the question, with a policy management engine to enforce system configuration policies. The authors conclude that description logic is a powerful formalism for policy representation, and that the state of natural language processing is getting close to direct policy conversion to machine language. Therefore, for at least one well-defined domain, natural language machine processing is quite feasible.

(Shi & Chadwick, 2011) state that when asked, users already intuitively know the access control policy associated with a given set of resources. If asked, the person will say yes or no according to a set of rules associated with their domain. However, translation of those rules into an executable policy requires specialized knowledge the user may not have. To counter this knowledge gap, the authors created a controlled natural language interface to allow the user to enter the policy, which is then translated to OWL's relational ontology or XACML. Roles are created first, then permissions are assigned to each role to implement the controls. These capabilities are then parsed by a natural language parser to create rules that can be exported to the desired format. User trials were conducted with instructions to create a short policy from a given usage scenario, and evaluated against XACML statements for correctness. The natural language interface scored approximately sixty percent satisfaction in trials. The authors contend that because natural language can specify the same fact in several different manners, the parser does not interpret them all congruently. The existing prototype does not address complex conditional policy statements that would be needed to enforce constraint based security policies.

Finally, we examine PolicyMorph, a constraint system that supports interactive policy development and maintenance as Access Decision Functions (ADF) (LeMay, Fatemieh, & Gunter, 2007). PolicyMorph is designed to interactively assist administrators with attribute based access control constraints. The tool reports

constraint violations and suggests resolutions to them in priority order. Further, PolicyMorph does not force the administrator to encode all constraints in formal language prior to analysis. Both an ABAC policy language and a logical constraint language are embodied in the authoring environment, and both are based on first-order logic and encoded in Prolog. The authors present a case study embodying separation of duty constraints. The resulting output does prioritize conflicts, but does not appear user friendly. The authors indicate a graphical user interface is their next priority to allow less sophisticated users to apply the environment.

(Reeder, Karat, Karat, & and Brodie, 2007) address usability challenges in security and privacy policy authoring interfaces. They state that as pervasive computing grows, users who will have to specify security policy will become less sophisticated in their expertise. Therefore, usable policy-authoring interfaces are becoming more necessary in the marketplace. Using IBM's SPARCLE workbench as an environment, they conducted a usablity study on policy authoring. There were five general challenges that must be addressed if a policy authoring infrastructure will be useful:

1. support for object grouping,
2. enforcement of consistent terminology,
3. making default rules clear,
4. communicating and enforcing rule structures, and
5. preventing rule conflicts.

A usability study was conducted with summer office interns, who were considered novice application users. Data collected during the study included the text of rules written, video of the subject and the computer screen, any think-aloud audio, and a demographic survey. Output was examined to address rule syntax, and then user activities were analyzed to determine if the subject was able to self-correct problem rules in the process. The results of their study corroborated the usability issues presented as the primary issues in creating a usable policy authoring interface.

Essentially, users do not wish to input large lists of data with minimal feedback as to the correctness of syntax.

(Johnson, Karat, Karat, & and Gruenberg, 2010) discuss using policy templates to facililty security policy authoring to proivde consistent policy interfaces across diverse policies. The templates are designed to provide a structured format to capture data. Iterative policy refinement is applied to address policy authoring, template authoring, and policy element definition as distinct steps in the policy creation process. The use of a structured template, in conjunction with the three roles associated with policy authoring, proved quite useful and policies could be created by less sophisticated users. The authors are exploring how to extract network resources to present the user with policy options appropriate for the domain of use.

(Johnson, Karat, Karat, & and Gruenberg, Optimizing a Policy Authoring Framework for Security and Privacy Policies, 2010) conducted further research in policy authoring with templates and determine that there were three additional criteria that needed to be added to policy authoring environments:

1. Support for appropriate limitations of expressivity (allow writing of deny policies or allow policies, but not both).
2. Communicate risks and threats associated with a given policy.
3. Provide access to the metadata for reference purposes.

The author's stress the need for extensibility in policy authoring tools to facilitate development of sound policy that matches the user's intended purpose.

As early as 1987 (Ayuso, Varda, & Weischedel, 1987) stated: "the success of all National Language Interface technology is predicated upon the availability of substantial knowledge bases containing information about the syntax and semantics of words, phrases, and idioms, as well as knowledge of the domain and of discourse context.

Semantic knowledge includes at least two kinds of information: selectional restrictions or case frame constraints which can serve as a filter on what makes sense semantically, and rules for translating the word senses present in an input into an underlying semantic representation. Beyond these elements:

- Basic facts about the domain must be acquired, at minimum taxonomic information about the semantic categories in the domain and binary relationships between semantic categories.
- Knowledge that relates the predicates in the domain to their representation and access in the underlying systems.
- Sets of domain plans to allow understanding of narrative and to follow the structure of discourse. Otherwise known as being able to interpret narrative in the context of stated future direction.

In IRAQ, there are 3 levels of representation for the concepts, actions, and capabilities of the domain. The domain model is separate from the model of the entities in the underlying system.

(Edwards, 2005) stated that a variety of tradeoffs that have to be made between structured and unstructured representations for many types of information. These trades become especially problematic in the case of contextual information for the following reasons:

- Context represents information about people that is very often ambiguous by nature, subtle in its interpretation, and can be applied to many uses.
- There is a great range of information about humans that is potentially useful (ranging from general information about users' locations or actions, to domain-dependent information such as a user's context in a specific application).
- Different sorts of context are important to different applications.

While highly structured data representations are amenable to use by applications (they can be easily machine parsed, processed and stored) they are problematic in situations where the needs of the applications are evolving; where the range of information that must be represented is very great; and when agreement among

multiple applications is required, in other words, the very situations posed by context-aware computing.

Storytelling As A Design Tool

In Human Computer Interaction, a mental model is a set of assumptions or beliefs about how a system works. People interact with systems according to their beliefs and assumptions about the system. Constantine and Lockwood (1999) define 4 criteria for product usability: learnability, retainability, efficiency of use, and user satisfaction. Learnability and retainability reflect on the role of mental models in usability. To the extent that a correct mental model could be learned and retained by the user, the user will become more effective.

Asgharpour, Liu, & Camp (2007) discuss five widely used conceptual mental models of security risks implicit in language or explicit in metaphors:

1. Physical safety – implicit in descriptions of locks and keys. This concept implies individual and localized control.
2. Medical Infections – security incidents interpreted as medical infections are grounded in the patterns of diffusion of malicious code as infectious diseases, and the importance of heterogeneity in the larger network, conceptualized as an ecosystem of security.
3. Criminal behavior – security violations can be crimes or may seem to be criminal. The concept of computer risks as risk of being a victim of crime implies that users or machines are targeted.
4. Warfare – implies the existence of a determined, implacable enemy, with the potential to leverage horror by leveraging the horrors or war.
5. Economic Failure – Security and network or software vulnerabilities can be seen as market failures. Vulnerabilities are perceived as external events; security failures cause downtime and expenses.

(Wash & Rader, 2011) discussed mental models, or how a user thinks about a problem, the model in the person's mind of how things work. The model allows the person to make decisions about the effects of various actions. For example, if hackers are perceived to be curious teenagers, the threat of criminal activity is perceived to be low. The critical point about mental models is that even if they are incorrect, they can

still lead to good security behaviors and more secure computer systems. Further, mental models of security threats are based on reasoning about information provided by stories recounted by friends and colleagues. This has been described as a folk model or lay theory created out of shared community experiences. Wash further states that to improve home security get home computer users to train each other and create good mental models.

Users rely on others for security because they feel like they don't have the skills to maintain proper security themselves, so they often try to avoid security decisions (p. 3). They find ways to delegate the responsibility to some external entity defined as technological (a firewall), social (another person or IT staff) or institutional (a bank). Three common approaches to address this issue are:

1. Technical solutions to take the decision out of the end users' hands. The stupid human approach requires a one size fits all security solution, but people use computers in vary diverse ways.
2. Educational approaches try to teach the details of computer security. As long as it isn't too complicated it stands a chance, at least in the short term.
3. How to support and encourage good behavior. How do people form their perceptions?

Wash & Rader believe that people form mental models of threats to security based on information they receive in the form of stories from other people like themselves, from their media, and from their experience. A "mental model is a cognitive representation in a person's mind of how things work: how a person reasons and makes inferences about a situation, allow people to make predictions about what might happen, and provide heuristics and guidelines to base behavioral choices (p. 1)." The mental model provides a chain of theories that help us reason about what to do next.

Social information sharing is an important way that we learn about the world around us and our behavior in the world. Narratives, stories told by other people, are an important component of our ability to learn about the world around us and behave appropriately. Stories tell people about each other, acting as observational learning and helping to avoid other's mistakes. Stories about others reveal useful information about how culture and society operate. Stories provide a way to learn from other's experience. Stories that affirm what is already represented in our mental models are remembered more easily, are given more weight, and are more likely to be passed on.

(Triantafyllakos, Palaigeorgiou, & Tsoukalas, 2008) present the argument that collaborative design is actually a narrative, a way to deconstruct the design process. They propose the theory that output of collaboration is actually a design story, in that it is the result of a chain of events and the interaction of a collection of characters. Further, the way that output is communicated is also a story in that it is the story of the item being designed, the people that will use it, and the affect caused by the use of the item. Designers and customers become readers and actors in the design process story and co-authors of the product. The design team leader becomes the narrator, and, in the delivery of the product, becomes the critical reader of the story.

Chatham (Chatham, 1978) discusses the author as the manipulator of narrative elements including character, setting, and events to construct the design story that is recounted and revised throughout the design process. Chatham states that narrative transmission is concerned with the manner in which the story's events are presented during the recounting of the narrative. The most important aspect of the narrative transmission in Chatham's model is the organization of the story's events, when the narrative begins, climaxes, and ends. Different design processes may have different stories, and different end products.

The designer is responsible for structuring the process such that the design stays on cost and schedule while accommodating the design team's creativity and innovation throughout the narrative (Triantafyllakos, Palaigeorgiou, & Tsoukalas, 2008, p. 212). In this role the team leader is responsible for the maintenance of the team's inner monologue as it progresses through the various design decisions and events.

(Erickson, 1996) talks about story as an integral part of the design process, a technique to generate discussion, inform the user, and persuade the users. Design is a collaborative activity between the designer and the customer, and has become a distributed social process as enterprise applications have grown in scope and structure. Erickson uses story as a change management tool, to make the user feel in control of his situation when technology is overwhelming the situation The metric is not the story itself, but the fact that the audience relates to it and responds, offering opinions on topics that they may not have otherwise offered inputs about. Stories help define what is important, what the user's environment is like, and set the stage for more formal design methods. Finally, stories are memorable, in that a good story is talked about with others and they are relatively informal. They are not expected to be precise, so they discuss the issues, not the minituae that delay the design process.

Computer Assisted Storytelling and Analysis

In (Nissan, 2008), narrative is proposed as a communication medium among avatars in a virtual world or robots performing a collective function. Nissan observes that "narrative is pervasive; even the reasoning process unfolds in a narrative way (p. 518). "At a formal level, narrative governs argument in that arrangement, the ordering or internal progression of a discourse, depends upon a narrative structure in which a premise is elaborated, developed, proved or refuted.

Narrative as arrangement is in this sense intrinsic to logic as well as to dialectic and rhetoric" (Goodrich, 2007, p. 348 in Nissan, 2008).

Further on, Nissan quotes (Ryan, 2005, p.347) in the definition of a story:

> 1. The mental representation of a story involves the construction of the mental image of the world populated with individual agents (characters) and objects (spatial dimension).
> 2. The world must undergo not fully predictable changes of state that are caused by non-habitual physical events: either accidents ("happenings") or deliberate actions by intelligent agents (temporal dimension).
> 3. In additon to being linked to physical states by causal relations, the physical events must be assoicated with mental states and events (goals, plans, emotions). This network of connections gives events coherence, motivation, closure, and intelligibility and turns them into a plot (logical, mental, and formal dimension).

Essentially, a story lives in five distinct dimensions (spatial, temporal, logical, mental, and formal). In working with robots or virtual avatars, Nissan poses that what is habitual or routine can be captured in a behavioral specification language, and what in non-routine is significant. The non-routine activity can then be processed against known patterns of activity (p.518).

At this point Nissan moves into a discussion on the analysis of common folk tales. Folk tales are inspired by some historical event that has been adapted to the current situation. In this manner, events are handed down through the oral tradition and are transformed by being brought into harmony with the thematic patterns of the current day. For example, Rumpelstilskin was originally documented by the Brothers Grimm in the 1812 Edition of Children's and Household Tales. The folk tale exists across the Scandinavian and European countries in various forms. Further, we can recognize the situational pattern in current society:

> "Rumpelstiltskin Syndrome" is an analogical reference to the role of the king in the story of Rumpelstiltskin. Common practice in middle management is to impose unreasonable work demands on subordinates. Upon completion of the task or tasks in question, equal or higher work demands are then imposed; moreover, no credit, acknowledgement, or overt appreciation is demonstrated by way of recognition" (Beatie, 1976).

Nissan makes the case that robotic operations can be defined by a taxonomy of situations (p. 524). This taxonomy can apply the thematic patterns of folktales, which have already been extracted in the work of Vladimir Popp (Popp, 1968); who proposed a mathematical model for the thematic patterns in folktales, which can be adapted for machine generated narrative. Popp's work facilitated the creation of story grammars and automatic story processing. Popp postulated that there were twenty-two specific folktale functions, or actions, and nine more that were considered preparatory functions in the story introductions. Aarne and Thompson (Uther, 2004) classified folktale narratives into categories of stories based on the actions and moral lessons to be learned.

(Cavazza M. and Pizzi, 2006) provides a critical overview and introduction to interactive storytelling systems, which integrates artificial intelligence techniques to generate narrative action sequences and animation. The authors translate Popp's characterization of folktale functions into narrative events, such as transgression, deception, wedding, struggle, and punishment. These functions form primitives that are used to construct sequences of events (p. 73), and provide a formalism for narrative structure. The four primary points of Popp's work are:

1. Narrative functions are the basic primitives of folktales. They are stable and invariant elements, independent from the characters that execute them, and from the modalities of their execution.

2. There are a limited number of such functions, which form the primitives.
3. Functions will always occur in the same order, although any given folktale may use a proper subset.
4. The order does not allow backtracking.

Beyond this structure, Greimas (Cavazza M. and Pizzi, 2006, p. 74) built a role-based analysis of narratives. The concept of an *actant*, was added to define characters. Actants were defined in pairs of oppositional characters, such as Subject v. Object. With the addition of Greimas' actants and Popp's functions, we now have the actions and characters required to build a story.

In (Brooks, 1996), the observation is made that stories tend to be written in a linear fashion, and are perceived by the audience within the context of their cultural experience. Stories tend to be told sequentially, with a beginning, middle, and end state. However, they are not usually created sequentially. Authors start with an idea, and may begin in the middle or at the end of the story. The finished product is refined over several drafts; revised at the request of editors, directors, or producers; and eventually goes to press, where the intended audience votes with their wallets on the author's success or failure. Figure 15 illustrates the sequential nature of story production.

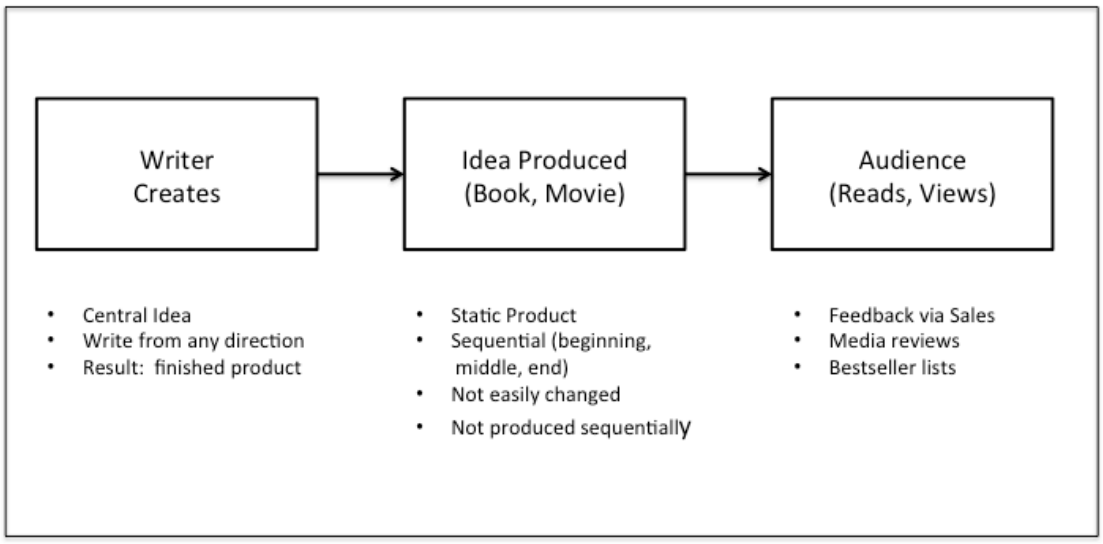


Figure 15 – The traditional model of story creation.

The notion of computational narrative explores the customization of the storytelling experience for each audience member. In a computational narrative model, the audience actively participates in the navigation of the story. For example, a game such as Adventure or Dungeon and Dragons takes different execution branches depending on the user's decisions. With a computational narrative model, the story can be tailored to reflect each participant's perceptions.

Kevin M. Brooks at the MIT Media Lab conducted one of the early experiments in computational narrative (Brooks, 1996). Brooks decomposed a story into 3 atomic components:

1. Events,
2. People, or characters, and
3. Things.

These three components are built into a narrative, which describes the organization of information. A narration explains how the narrative is expressed to the intended audience. For example, a movie may start as a narrative created by the screenwriter.

As it moves through production, various other experts such as costumers, dialogue coaches, and the director may alter the narrative over several iterations. The resulting end product is the narration. Figure 16 illustrates the iterative nature of narration creation.

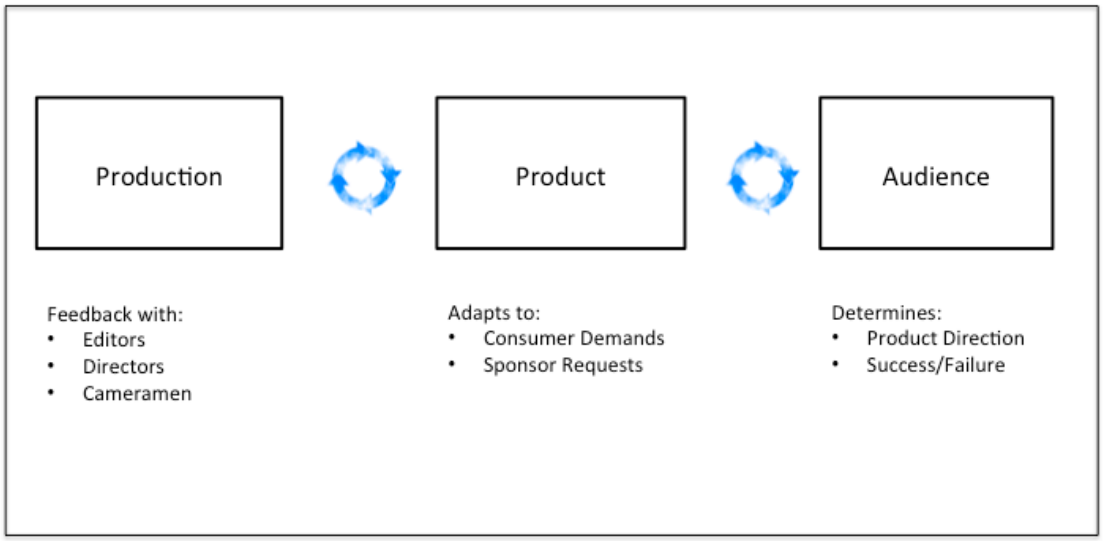


Figure 16 – The actual way narrative is created.

Brooks believed that a story could connect the creator with the audience through a computational model. In Brooks' model, a narration was created through four different processes:

1. *The representation process* – which defined the components of the story.
2. *The presentation process* - determining how the components are revealed to the audience.
3. *The reasoning process* – which applies logical inferences about the components based on the representation.
4. *The reasoning engine* – which coordinates the processes of reasoning, representation, and presentation, and provides the results to the audience.

Through the use of behavior-based artificial intelligence, the reasoning engine in Brooks' model was capable of adapting either to the audience's response or to the creator's manipulation of the elements, based on the inference engine's recalculation

of the story line. Figure 17 illustrates the computational narrative model of story creation.

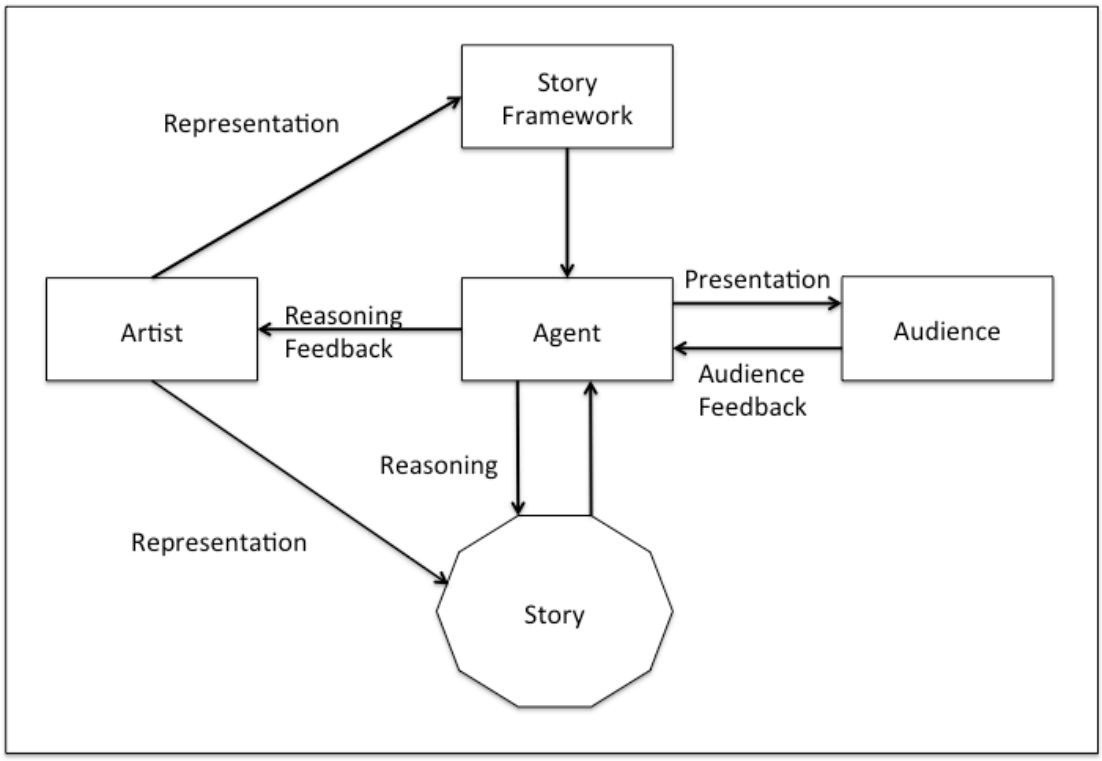


Figure 17. The creation of a narrative, computer generated.

Brooks further defined three components of a computational narrative:

1. The representational environment – which reasons about the world presented in a given story.
2. The structural environment, or story framework – which provides the basic story events, such as characters, conflicts, resolution, diversions, and endings.
3. The presentation environment – which presents the results of the interpretation of the story framework and the representation to the audience, and delivers feedback to the artist for subsequent structure and story manipulation.

The computational narrative model thus defined was applied as the basis for the Agent Story environment, an early prototype for interaction between the creator and the audience in storytelling environments.

From the preliminary perspective provided by Brooks, research continued into how to capture collaboration among authors and tell stories from alternative viewpoints. Mazalek and Davenport have explored the use of interaction platforms to define the structural environment and help the audience develop a frame of reference (Mazalek A. and Davenport, G., 2003). With their colleague Ishii, Mazalek and Davenport (Mazalek A. et al., 2003) developed a Tangible Viewpoints system to allow collaborative authors to choose a story's direction through the manipulation of pawn-like tokens on a game screen. With this model, the data is represented in 3-dimensions, and the collaborators can maintain a frame of reference for each character of interest (Mazalek A. et al., 2002). Case study experiments of how the Tangible Viewpoints model works in actual collaborations are included in their work.

Zagalo and Szilas proposed refinements to the basic framework of a story. In (Zalago, Barker, & Branco, 2004), emotion as a feedback mechanism is introduced, and a feedback notation for detection, categorization, and intensity is feedback provided by the story agents. Szilas looks at action as a narrative structure element that augments the story framework as a refinement mechanism based on audience interaction (Szilas, 2004).

SUMMARY

To conclude our review of the literature, we have a large and diverse set of security policies that have formal models. We have several policy authoring tools that explore templates, constrained natural languages, and graphical authoring environments. Further, we have a collection of analytical tools to disassemble narrative text and story elements into their basic components for textual analysis.

However, the policies discussed are highly formal, the policy authoring environments constrain the user in syntax and expression, and the narrative analysis tools have not been applied to security policies.

Chapter 3

Methodology

This chapter describes the research methodology, the research methods to be employed, development and validation of the experiment, results presentation, and resources required to complete the experiment. Recall, the high-level goal is to determine the feasibility of defining a top-level narrative-based security policy that can (a) support a significant degree of formalism; (b) accommodate verification of completeness and (c) traceability to lower-level functional system specifications. The research objective is the definition of the logical security principles that comprise the system security policy such that:

(a) they can govern system behavior,
(b) are enforceable throughout the system design process, and
(c) are comprehensible to the system stakeholders.

An end-user should never have to be concerned with the format of an access control list or firewall rule: a general statement of access management should be traceable to lower-level implementation specific constraints. Our hypothesis is that the security policy elements required to implement complex security meta-policies can be best expressed as story or narrative elements, which can be decomposed into sub-policies and system requirements. The eventual result of such decomposition is that the policy enforcement mechanisms can be explicitly traced back to the specific story elements within the top-level policy statement.

Overview of the Research Methodology

There are few research methodologies that have been successfully applied to the domain of security policy. Unlike traditional computational research, there are no

performance measures associated with security policy. That is, one cannot say that a given policy results in a faster solution N-percent of the time, or that one policy requires more execution statements than another. Therefore the research methodology selected for this project is a qualitative research model, in particular a grounded theory methodology as described by Creswell (2013). Creswell differentiates narrative studies as those creating a portrait of an individual (p. 122), and ethnography as the study of a culture-sharing group on a large scale (p. 122). Neither of these qualitative methodologies applies to security policy research. Examination of most security policy research indicates an emphasis on formal proof of a given security policy model (Bell D. , 1994) (Bell D. & LaPadula, L., 1976) (Bertino, Ferrari, & Perlasca, 2001) (Gligor, 1995) (Lin, 2000). These models essentially are logical case studies that emulate specific elements of behavior. Phenomenology does not apply, as this research does not focus on the lived experiences of individuals around a phenomenon, such as a war or a cultural revolution (Cresswell, 2013).

This leaves the qualitative methodology of grounded theory. Our objective was to generate a substantive theory and validate it through systematic procedures for data collection, analysis and categorization, and specification of the context and conditions under which the theory operated (p. 123).

Specific Research Method(s) to be Employed

Grounded theory methodology (GTM) is a systematic, qualitative procedure to extract information (Chakraborty & Dehlinger, 2009). GTM can be further described as "a qualitative research method that uses a systematic set of procedures to develop an inductively derived grounded theory about a phenomenon" (Strass & Corbin, 1998). A worked example of the application of GTM to enterprise system requirements is presented in Chakraborty & Dehlinger (p. 333), who apply GTM to:

- Present a structured, qualitative analysis method to identify enterprise requirements
- Provide a basis to verify enterprise requirements via high-level enterprise architecture objectives
- Allow for the representation of business strategy in a requirements engineering context
- Enable the traceability of enterprise architecture objectives in the requirements engineering and design phases.

Figure 18 illustrates the coding phases of GTM in the context of requirements engineering, as interpreted by Chakraborty & Dehlinger (p. 335).

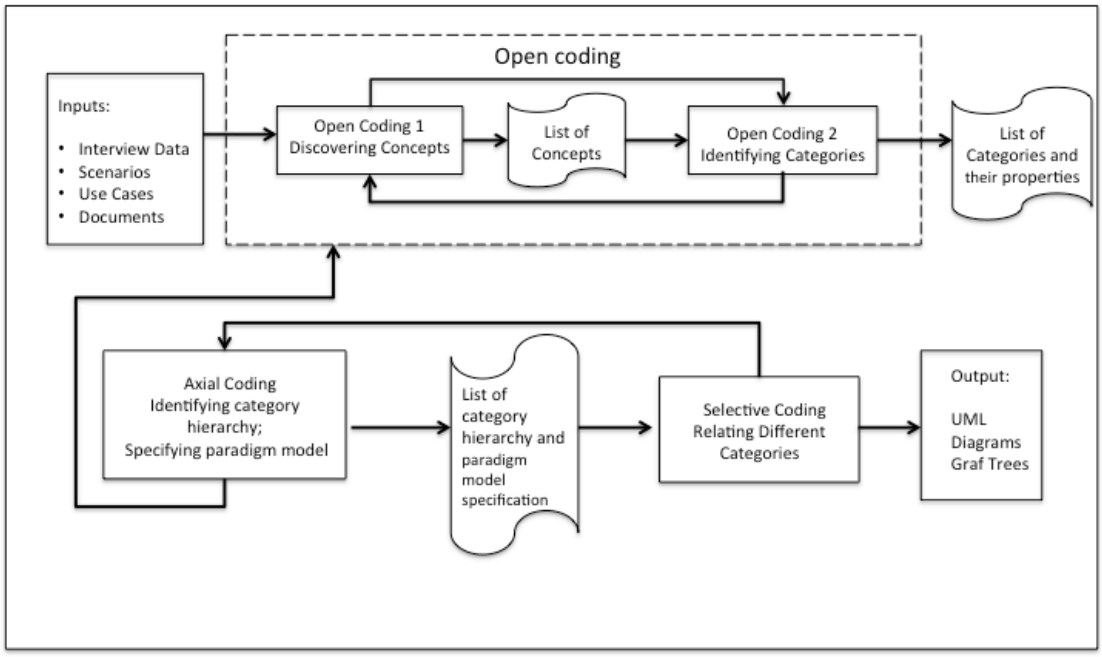


Figure 18. Coding phases of GTM in the context of requirements engineering (p. 335).

Similarly, Lehmann (2010) discusses the application of GTM to Information Systems, which are defined as action networks of technology and people. Lehmann contends that GTM was originally created to uncover social theory from empirical data generated across a broad spectrum of contexts and activities, and to create theoretical foundations which were nonexistent (p.1). Indeed, GTM is defined as the

discovery of theory from data – systematically obtained and analyzed in social research (Glaser & Strauss, 1967). Essentially, GTM is a simple research sequence: "I gathered data and once the data was arranged in neat piles, I wrote them up" (Stoller, 1987). The arrangement of the data, or its coding, are defined in three steps in Miles and Huberman (Miles, 1994):

1. Commonalities in the data are captured in descriptive codes to more clearly capture the essential attributes of the phenomenon;
2. As more data and codes are available, interpretive codes are abstracted from the idiographic confines of the concrete incidents to help understand what is going on behind the data;
3. Lastly, inferential pattern codes, now abstract of space and time and etic to the substantive range of the research, are conceptualized: they are explanatory and often predictive.

The important element of GTM is the analysis, not necessarily the data itself. Strauss & Corbin (Strauss, 1990) state:

> Concepts are the Basic Units of Analysis. A theorist works with conceptualizations of the data, not the actual data per se. Theories can't be built with actual incidents or activities as observed or reported; that is, from "raw data." The incidents, events, and happenings are taken as, or analyzed as potential indicators of phenomena, which are thereby given conceptual labels… As the researcher encounters other incidents, and when after comparison to the first, they appear to resemble the same phenomena, then these, too, can be labeled (in the same way). Only by comparing incidents and naming like phenomena with the same term can a theorist accumulate the basic units for theory.

Figure 19 illustrates this cycle. In GTM, samples are analyzed until all the categorizations and properties are validated.

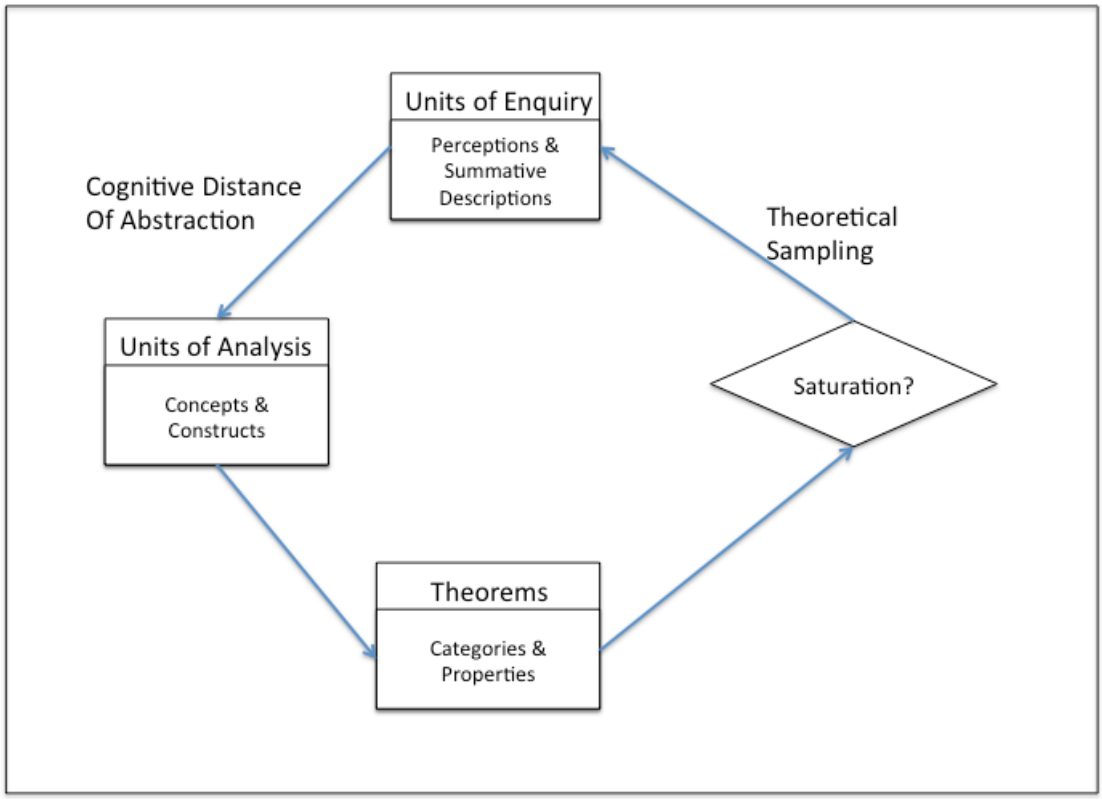


Figure 19. The analysis cycle of GTM (Lehmann, p. 5).

Research Design

We applied GTM to the research questions posed in Chapter 1. Using this methodology, the workflow associated with the research is presented in Figure 20.

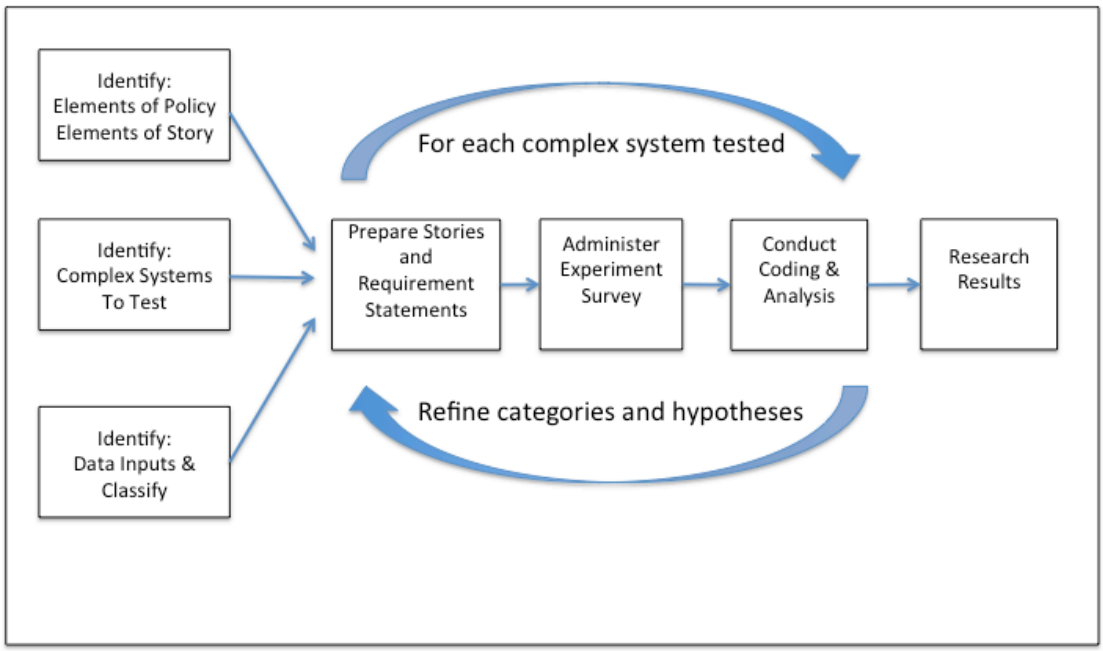


Figure 20. Experiment Workflow

Identification of the elements of security policy and the elements of story have been conducted as part of the preliminary research for this experiment. (Carnielli & Pittarello, 2009) present a case study to validate story analysis against the plot of an actual novel addressing the contextual data involved in an autobiograpical story. They use the approach defined by (Segre & Kemeney, 1988) which defines four levels of narrative analysis:

1. Discourse – the linguistic, stylistic, and metric features of the text;
2. Story – the set of actions as they are presented to the reader by the author;
3. Fabula – the set of actions logically and chronologically ordered;
4. Narrative model—structure characterized by invariants common to a set of texts.

Carnielli and Pittarello apply these levels to create a decision tree of "scenes" that define the spatial elements of the physical environment. "Situations" are a set of

actions the user may choose from that will trigger the next set of actions within the story (p. 93).

We propose to apply the narrative structure of Segre and Kemeney in conjunction with a story designed to elicit the policy elements required for RAdAC access control decisions. Our objective is to capture the policy elements for use by the risk decision element of the RAdAC model. Figure 21 illustrates the RAdAC model, with the policy elements to be elicited highlighted.

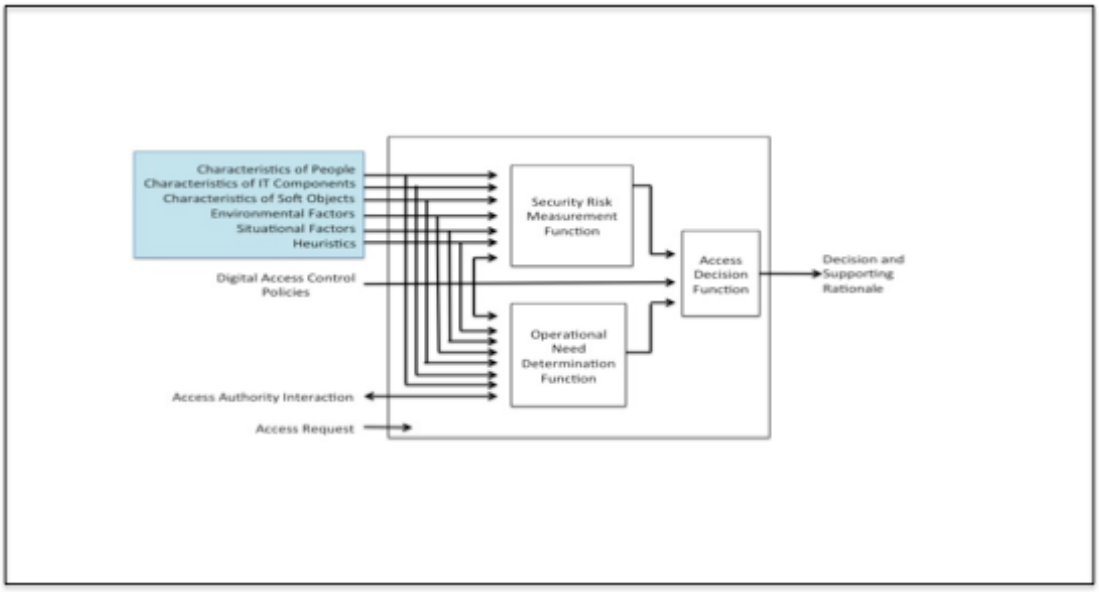


Figure 21. RAdAC model, with policy elements highlighted.

Table 2 presents the definitions of these elements from (Government, Global Information Grid Information Assurance Reference Capabilities Document, 2004 p. 3-15).

Table 2. Risk Attributes of RAdAC Model

Attribute	Definition
Characteristics of people who create and consume information	Unique identifier, citizenship, security clearance level, source, organization, community of interest membership(s), rank, length of service, job title, system privileges, operational position
Characteristics of IT Components that create information and enable people, applications, or services to create, share, or use information	Unique device identifier, operating system, hardware features, owning organization, network connectivity, location, certified system administrator, current system certification
Characteristics of Soft objects such as files or databases that are shared	Data, applications, services, identifier, sensitivity level, releasability, protection quality, source, originator, intended usage
Environmental factors	Physical location, adversarial threat level, operational need
Situational factors	National, enterprise wide, or local indicators of a situational condition such as the threat level associated with a particular type of attack (cyber, terrorist, nuclear)
Heuristics	Knowledge acquired from past sharing and access decisions such as characteristic profiles for all other factors, plus weighting factor in the overall access decision.

Identification of Complex Systems for Experimentation

The next step in the experimental process is the selection of complex system architectures that can be subject to analysis. In the book Thinking in Systems, Donella H. Meadows defines a system as "an interconnected set of elements that is coherently organized in a way that achieves something….a system must consist of three kinds of things: elements, interconnections, and a function or purpose" (2008, p. 11). (Meadows, 2008) Complex systems apply various enforcement mechanisms to enforce a comprehensive security policy. For example, a cloud-based system

might employ firewalls, routers, directory services, and various application mechanisms to enforce a security-as-a-service architecture. For the purposes of this set of experiments subject matter experts on the given system(s) and their policy enforcement mechanisms must be accessible to the researcher. This will support validation of the story and the narrative requirements statements. With the policy attributes and story components identified, selection of the systems was the next critical item in the process. Three distinct system architectures were selected for experimentation.

Architecture 1 is an enterprise mail infrastructure. It supports a global Fortune 500 company with over 15,000 mail clients deployed. This infrastructure is used to support both internal and external electronic mail, and is directly connected to the Internet. Figure 22 illustrates the enterprise mail infrastructure.

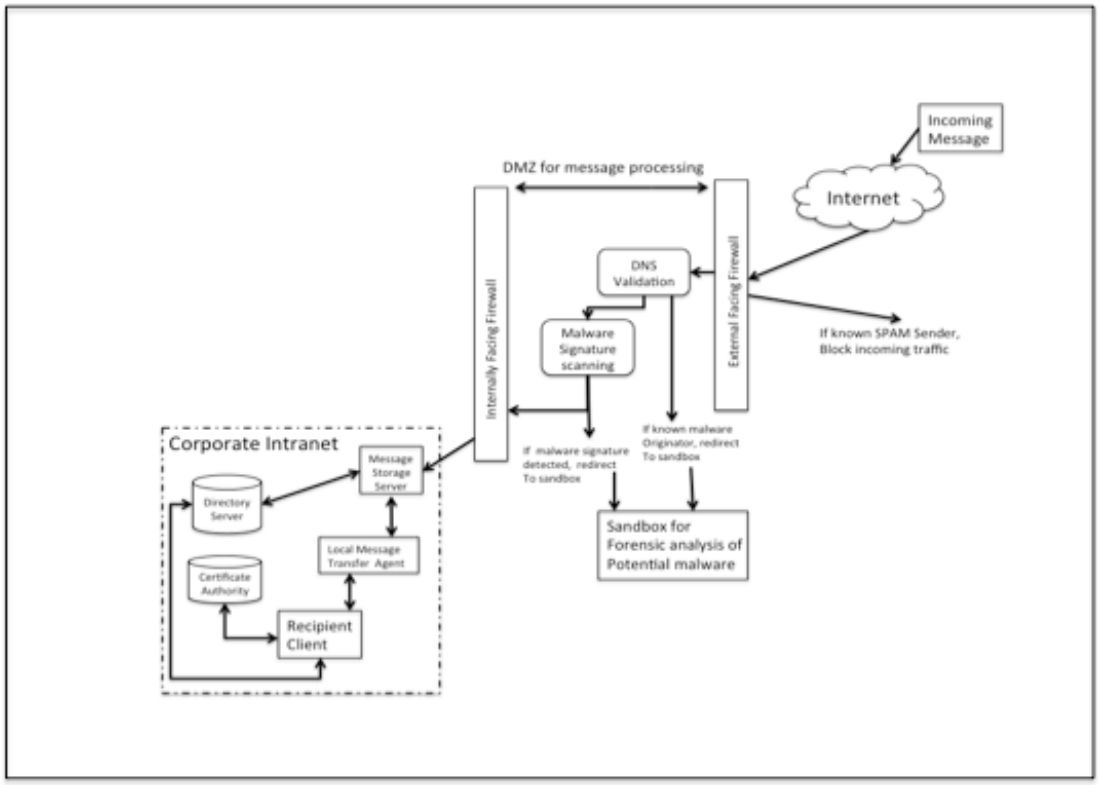


Figure 22. Representative enterprise e-mail architecture.

Architecture 2 is a network infrastructure responsible for connecting over 15,000 individual sites for a mission critical systems control application. In this architecture, the network device health and status information is segregated from the device data streams. That is, information on the health of the network and its connections is transmitted to a management node, while information used by the network applications is transmitted to analytical nodes for processing and visualization. Figure 23 illustrates this architecture.

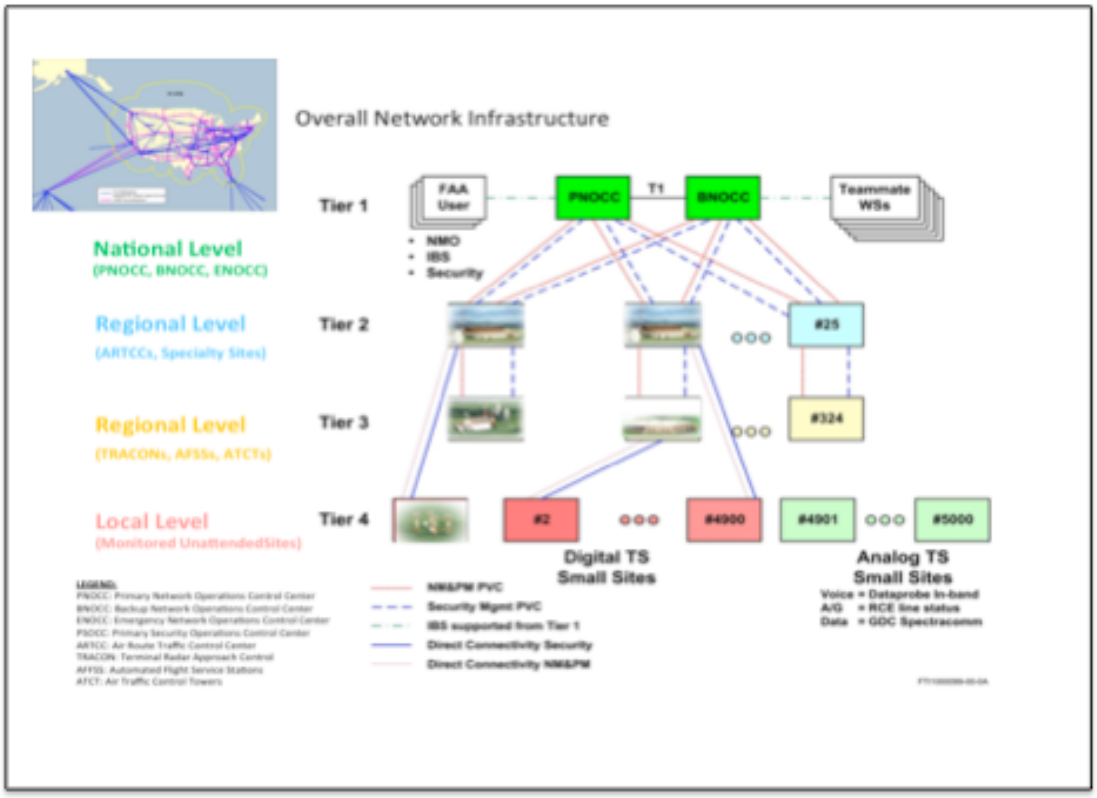


Figure 23. Network Infrastructure Architecture for 14,000-site network topology.

The third system architecture is an enterprise services architecture that crosses from a secure domain to an untrusted domain in a publish-subscribe model. Architecture 3 reflects the circumstance where an information provider wishes to make data products available to the public, but wants to make absolutely certain that

the subscribers cannot "reach back" into the enterprise intranet and extract additional unauthorized data. Figure 24 illustrates the system architecture for this system.

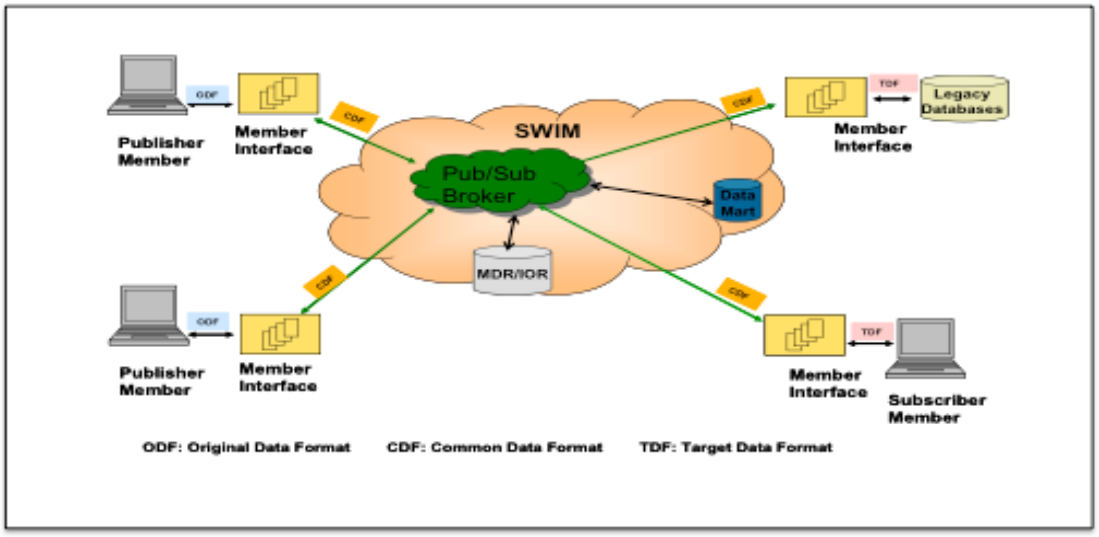


Figure 24. System architecture for one-way publish-subscribe message broker.

Identification of Data Inputs and Classification

Planned data inputs for the experiment consisted of:

1. The story narratives designed to address the attributes of risk adaptive access control policies. Story narratives will be written with the assistance of system subject matter experts. These subject matter experts will review the security policy defined in the narrative to correct any errors in policy statement.
2. A requirements specification addressing the attributes of risk adaptive access control policies. Requirements statements will be written with the assistance of system subject matter experts. These subject matter experts will review the requirements statements to correct any omissions in the policy requirements specification.
3. Comprehension questions developed to assess the reader's understanding of the policy. To eliminate author bias, the questions will be independently developed by a security subject matter expert with no knowledge of the overall experiment.
4. Demographic questions for the experiment participants. These questions will address age, computer system usage, general occupation, and years of experience. The demographic questions are for classification purposes only; they will not be used to identify the subjects by name.

A trial experiment was conducted with a small group of cooperative subjects, in these cases members of the local Information System Security Association (ISSA) chapter. Using the planned data inputs proved problematic in a group setting; some subjects simply decided to leave early, and chose not of participate. Others wanted to debate potential responses as if the survey was a certification exam. Further, some subjects got lost in reading a one page narrative and lost interest. This experience led to adjustments to the original plan to facilitate data coding.

Finding a sufficiently sized group that would not be considered biased by virtue of their employer made it simpler to conduct the survey online anonymously. Anonymity of survey participants was protected by not asking for names and not tracking IP addresses of participants.

A standard set of demographic questions was developed, to accommodate analysis both on an aggregate experiment level and at the level of architecture description, either narrative or requirements based.

A standard set of definitional questions was developed for the same reason, and also facilitated a common understanding of the terminology.

Finally, instead of a single consolidated story narrative, a selection of brief, one paragraph scenarios were used. These were designed to accommodate a short user attention span, and were matched with requirements specification language that reflected approximately the same scenario.

This completed the inputs to the experiments, and provided the framework for data categorization and subsequent analysis of the results.

Sampling

To conduct the research involved, three distinct research samples were developed. These research samples are all purposeful (Bloomberg & Volpe, 2012). The candidates selected are used to determine the cognitive value of using the security policy story. Three samples are employed, one for each of the complex system architectures used as data inputs. The sample subjects were made familiar with the same complex system architecture for each experiment.

Sampling Strategy

Sample personnel were originally planned to be the application user community that relies upon the given complex system. These personnel are the end users of the complex system architecture. To the sample personnel, the devices used to create the actionable data displayed on their monitors are transparent. Sample participants met the following basic criteria:

- Have at least 1year of experience as an end user of the system.
- Possess positions requiring use of the system on a daily basis.

Unfortunately, the communities of system users available for this experiment could have potentially been biased by virtue of their employers. In one of the scenarios, the users could have been the organization that operates the system. In another, the sample population would have been consumers of the system. To avoid these biases, an adjustment was made to the original plan: instead of using known populations, a more random population of subjects was solicited. The survey instrument was made available in two Linked In user communities: The CyberWarfare User Community and the DIACAP/FISMA/5800.2 authorization and approval community. These communities were selected because they have large membership populations (over 1,000 members each) and are fairly active (usually 5-10 different message threads have posts daily).

Research Design

The research was designed as a series of three experiments, one that matches each of the complex system architectures identified as data inputs to the research. Each experiment followed the same experiment protocol.

Preliminary Trial

Prior to conducting the three experiments, a preliminary trial was conducted with a smaller sample set of subjects. The purpose of the preliminary trial is to ensure the directions are clear and that the subjects are not confused by the instructions or the questions. The preliminary trial used a simple architecture of a print server on a local area network segment. Only authorized users within a given range of office space are allowed to print on the print server. The preliminary trial was expected to find and resolve any ambiguities in the questions and the instructions prior to initiating the complex architecture experiments. As was discussed above, several areas of the experiment were refined as a result of lessons learned during the preliminary trial.

Experiment 1 – Enterprise Messaging Architecture

Knowledge Workers/Participants

Participants in the study were a group of 50 Information Technology Users. This number of users is consistent with the sample size used in (Carnielli, 2009), which used a single sample of 35 users for one experiment. These engineers were recruited via announcement in the Linked In communities. The surveys were created in SurveyMonkey™, a survey design and analysis environment that allows creation of multiple choice and short completion surveys and provides a hosting platform for population surveys as services based computing application.

The subjects may have security policy implementation knowledge, but probably have not derived an information security policy. That is, if required, the

security policy to be implemented in their normal job function is one that has been provided to them. For example, they have implemented specific firewall rules, but have not had to derive the rules from network traffic analysis. The participants may have been responsible for device security configuration, which includes applying vendor patches, vulnerability scanning on a routine basis, and password management of administrative accounts.

Experiment Design

Experiment participants were provided with one of the two narratives: either a story-based security narrative or a requirements specification narrative. Both narratives reflected the same system architecture. The subjects were asked to read the narrative, and then take a brief survey. The survey consisted of demographic questions to facilitate subject classification, definitional questions to ensure a common understanding, and security policy questions developed to test their comprehension of the policy narrative.

Training

The study participants did not require training. The questions answered after reading the narrative were multiple choice categorizations or brief answer, similar to the questions asked in a traditional marketing survey or academic testing environment.

After the data collection session, the results from the experiment were analyzed. The two groups (requirements specification and story narrative) were compared to determine which narrative provided a more understandable (or memorable) security policy statement.

Experiments 2 and 3

Experiments 2 and 3 were conducted in the same fashion as Experiment 1. The difference is that these experiments were conducted with different complex

system architectures. By varying the system architectures applied to the experiments, any existing knowledge bias should have been eliminated because it is highly unlikely that the same subject sample would have prior knowledge of all three architectures. Sample sizes were identical for each experiment, and were divided between the requirements specification narrative and the story narrative.

Instrument Development and Validation

There were 6 experimental instruments developed. All instruments shared a common set of demographic and definitional questions, designed to facilitate coding of the populations. Each of the three system architectures required two system descriptions, one in narrative format and one in requirements specification format.

Each system architecture was tested against a 50 subject population group, with 25 participants receiving the narrative and 25 participants receiving the requirements specification language. The original plan was to randomly select subjects for each group until the population was full, and then begin populating the next system architecture. Unfortunately, Survey Monkey does not permit the notion of connected surveys. The experimenter had to monitor the response count, and change the survey link manually when the maximum subject threshold was reached for a given instrument.

Formats for Presenting Results

Results from the experiments were subject to categorization and open coding of results. Initial coding applied demographic information gathered from the participants to determine correlations between the participant's level of expertise and the accuracy of the resulting narrative. The initial code rubric was a simple matrix of the number of iterations and the types (minor/major) of clarifications required as a result of the iteration.

The 3 experiments were executed with 50 people each, divided into 2 groups per experiment, for a total sample size of 150 participants. Results from the preliminary trial were not included in the experimental analysis.

Aggregate results of the experiment are presented as trending analysis data and summary tabular information in Section 4. Best presentation mechanisms were determined as the results and experiments were refined throughout the analytical process.

RESOURCE REQUIREMENTS

The resources required to complete this experiment are minimal: a single personal computer for data preparation and result correlation. Items to be prepared include the data collection forms, the requirements narrative and the story narrative. Data collection forms were created using Microsoft Word and transferred to the Survey Monkey℠ platform to generate the survey instrument. The requirements specification narrative was created using traditional specification language samples such as those provided in NIST 800-53 (NIST, 2013). The story was created with Microsoft Word, after experimentation with Inspiration, an author's outlining toolkit, or the Writer's Dream Kit, an alternative story development environment. Ultimately the story creation environment was a function of personal preference. Someone less familiar with requirements definition and narrative might have been perfectly happy using Inspiration or the Writer's Dream Kit. All of these tools were accessible to the experiment designer.

SUMMARY

In conclusion, the experimental discussion has presented an overview of the issues associated with security policy specification and logical correspondence. Given that security policies have become more expressive with additional constraints and rule-based capabilities, reaching the eventual binary access control decision has

become a more complex task. Additionally, the need to provide dynamic security policies in the event of cyber attack scenarios means policy modifications must be rapidly propagated throughout an infrastructure. This does not leave time to resolve security policy conflicts when an entire mission critical network may be compromised.

We present an alternative approach, specifying security policy through storytelling. Stories provoke discussion, help our collective memory recall the success or failure of past policy attempts, and encourage alternative solutions. As security policies become more robust in their emulation of the "real world, " it will become more difficult to prove logical soundness with existing formal methods. Given that the ability to perform natural language processing is improving, it may be more feasible to perform policy analysis through the use of narratology. The use of computer-assisted narrative for policy management is an area that has not been explored to date.

Further research in ontologies and natural language authoring environments needs to be conducted to determine if policies can be expressed and analyzed with these capabilities. Mankind shares a common social history in story; our cultural differences change the context. As our social media converges in the global communications network, it will be interesting to determine if our stories and our policies converge as well.

CHAPTER 4

RESULTS

This chapter presents the data analysis and research results of the experiment. Recall, from Chapter 3, the objective of our experiment was to prove our hypothesis, that the security policy elements required to implement complex security meta-policies can be best expressed as story or narrative elements, which can be decomposed into sub-policies and system requirements. We have chosen to use the Grounded Theory Methodology (GTM) to validate the hypothesis, and designed our experiment to collect data to further facilitate our information coding. Bloomberg and Volpe (2012) illustrate the data analysis process in Figure 24 (p. 140).

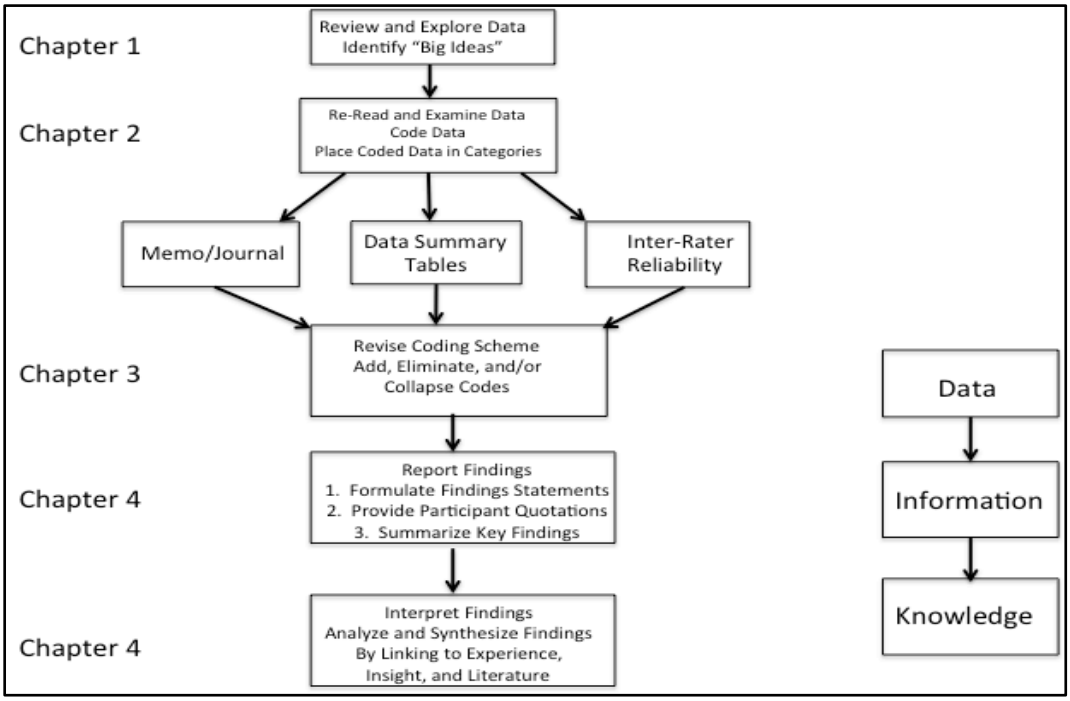


Figure 25 The Qualitative Data Analysis Process mapped to data transformations and dissertation chapters.

Data Analysis

Strauss and Corbin (Strauss, 1990) have identified seven criteria as a guideline to determine the quality of a grounded theory study. The applicability of the criteria to the data analysis is presented in the following sections.

Criterion Number One: how was the original sample selected?

The original sample was selected from local security practitioners in the Melbourne, Florida area. However, the consideration of sample bias was raised because most of the practitioners work for a single employer. To mitigate this bias, an alternate sample was selected. Linked In℠ is a popular professional social networking site that supports the notion of communities of interest. A notice was posted in two of these communities of interest: the Cyber Warfare Community of Interest and the DIACAP/FISMA/FEDRAMP Authorization and Approval Community of Interest. These groups were used because they are large communities of users, with over 1000 users each. The notice is illustrated in Figure 25.

Call for participation:

A graduate student at Nova Southeastern University is conducting a survey on access control policy specification. Participation consists of answering survey questions. The survey will take no more than 25 minutes of your time for all portions of the survey. All responses are anonymous. If you have any questions or concerns, please contact hronda@nova.edu. Please click on the following link to participate in this research project: www.surveymonkey.com/XYZ

Figure 26. Participation Solicitation Notice.

The criteria for sample selection were straightforward: if a candidate clicked on the survey and provided consent the candidate was added to the sample. When a given survey was full, the solicitation was modified to reflect the next sample group, until the full 150-subject sample was completed. The candidates were purely random and

anonymous. The only common element was that they were members of the appropriate community of interest.

Criterion Number Two: What major categories emerged?

The major categories had begun emerging in the literature survey, particularly in the discussions of security policy and computer assisted storytelling. From the various security policy formalisms, the notion of subjects, objects, and actions has become an accepted standard practice. That is, subjects are uniquely identified users or processes acting on the behalf of user. Objects are data files, or containers, that reflect computing resources available to the user. Actions are explicit acts, or instructions, that are conducted upon data containers, such as read, write, copy, and delete operations.

From the computer assisted storytelling realm, the concept of principal actors, actions, and catalysts emerged. Principal actors can be considered the primary characters in a story. Actions become the activities the principal actor engages in over the course of the story, and catalysts become the motivating factors that modify the activities. For example, in Miguel Cervantes' Don Quixote, the Don is the primary character, and he engages in a series of somewhat misguided quests over the course of the story. Finally, at the end of the story, the Don has the revelation that his actions have been misguided, and his actions are altered to reflect this change in perception.

These major categories were applied to the creation of the survey instrument. For clarity, the instrument was divided into a definitional component and a situational component. The definitional component was identical for both the narrative and the

requirements-based situational components. This was a refinement from the trial administration, where the experiment subjects raised explicit questions about contextual information and environmental dependencies.

Given that the comprehensive set of security controls in NIST 5800.4 (U.S. Government, 2013) is well over 250 pages of text, a single family of security controls was used as the basis for the survey instrument, the family of access control. This selection was made to align with the initial coding of security policy models and computational narrative constructs. A comprehensive survey instrument that addressed the entire family of access control requirements would have been prohibitively long for a voluntary participation survey. Further, creation of a narrative scenario would have become a small novel, challenging the subject's reading comprehension and attention span.

The thought of asking the subject to create a "security story" was also considered, but was discarded. While the human brain is wired to accept narrative information, it is not a universal talent to create and tell a story. Rather than bias the study with the possibility of bad prose from subjects who could not write, or force the subject to use unfamiliar authoring environments, the experiment became a comparative assessment of comprehension between a traditional requirements specification and small security vignettes.

Criterion 3: What were some of the events, incidents, actions and so on, (as indicators) that pointed to some of these major categories?

The major categories of coding were extracted primarily from categorizations that were made within the context of security policies (subjects, objects, actions) and the context of narrative components (characters, circumstances, actions). In the domain of security policies, earlier policy models were limited to <subject, object, action> triples. Over time, subjects and objects were further refined with specific security-relevant attributes. For example, classification of the data container and the clearance of the subject were the earliest security relevant attributes incorporated into the security policy domain (Bell, D. & LaPadula, L. 1976). The most robust model defined to date is that of risk-adaptive access control (RaDAC) (McGraw, 2004), where various attributes affect the security policy and the user's access is modified based on the values of these attributes. For example: a user's access may depend on the time of day, which type of device is being used, and where a user is located. Data that can be accessed when the user is on the corporate campus may not be accessible from a smart phone or tablet device when the user is on an airplane. This evolutionary refinement of the respective disciplines influenced the coding of data as the study progressed.

Criterion 4: On the basis of what categories did theoretical sampling proceed? Guide data collection? Was it representative of the categories?

Theoretical sampling was guided by three distinct requirements:

1. A need to collect demographic information about the survey respondents, to allow categorization of the sample.
2. A need to establish common definitional terms, or to at least provide respondents the opportunity to agree or disagree with the terminology applied in the study.
3. A need to support comparative analysis of narrative discourse and requirements specification.

These requirements influenced the data collection process, in that they shaped the data questionnaires as defined in Appendices C-J. The author defined the use of a narrative discourse or a requirements specification as the independent variable of the experiment, as it was the only section of the questionnaire that was not identical across all populations.

Criterion 5: What were some of the hypotheses pertaining to conceptual relations (that is, among categories) and on what grounds were they formulated and tested?

At the highest level, our hypothesis was that it was feasible to define a top-level narrative-based security policy that can (a) support a significant degree of formalism; (b) accommodate verification of completeness and (c) traceability to lower-level functional system specifications. The research objective is the definition of the logical security principles that comprise the system security policy such that:

(a) they can govern system behavior,

(b) are enforceable throughout the system design process,

(c) and are comprehensible to the system stakeholders.

Security policy research has usually been based upon the ability to accommodate verification of completeness and traceability to low-level functional system specifications through the use of formal methods and logical proof (Bell, D. E., 2005). Unfortunately, most users of computer systems do not have degrees in formal logic, and have not proven a theorem since high school geometry. To facilitate logical correctness, the author hypothesized that a grammatically sound English sentence could be used to communicate the concepts of subjects, objects, actions, and attributes to the end user. In many systems today, the end user is left to their own devices when establishing a security policy for an application or a given device. For example, Internet access to private residences does not come with caveats to enable wireless access protection through encryption, thereby allowing the entire neighborhood to piggyback on a single user's connection. Personal computers brought the Defense Information System Network (DISN) to its knees when military personnel brought work home on USB drives and unknowingly brought malicious code back to the office on Monday morning.

Our data collection activities addressed these issues by comparing the requirements defined in traditional requirements specifications with narrative discourse designed to present scenarios where access could be granted or denied. The objective was to explain the policy implementation within the context of everyday communication and language, not within the confines of symbolic logic and

set theory mathematics. Oftentimes the only way to extract policy rules for a complex system is to define usage scenarios, present them to the user community, and hope for consensus on the resulting behavior.

Criterion 6: Were there instances when hypotheses did not hold up against what was actually seen?

In a perfect world, the state of the art in all research areas for an interdisciplinary study such as this one would be sufficiently mature to accommodate the hypotheses at hand. Unfortunately, the world is not perfect. While we can express security policy in terms of subjects, objections, actions, and attributes, we cannot have both formalism and functional information systems. While some improvement in theorem proving technology has been made over the last 20 years, the feasibility of proving code bases of over 1 million lines of code is still beyond the state of the art.

Similarly, computer-assisted storytelling is very much in its infancy. While the research community understands the components of a story-telling computer, our ability to develop an ontology accommodating the vast nature of human language is somewhat lacking. There have been attempts to apply neuroscience to understand how a human catalogs experiences and develops comprehension, but the ability to accurately measure and monitor these capabilities is just emerging into the mainstream research community. The computer-generated stories of today are of the "See Spot. See Spot run." variety: simplistic and lacking the rich contextual background that should accommodate concepts such as RaDAC. When these areas

were discovered in the course of the research, they were noted and documented in this dissertation.

Criterion 7: How and why was the core category selected (sudden, gradual, difficult, easy)? On what grounds?

The core category of this research was selected gradually, over time, and with some difficulty. Conceptually, the notion of a security policy being understandable is quite simple. Our society has standards for sharing, and not sharing, information that have not quite become as soundly established in the realm of cyberspace. Understanding the sociological basis of story and how the human brain is designed to comprehend narrative story was an emerging interdisciplinary process. Defining security policy in the context of storytelling became less complex when the categorizations of story structure and components were aligned with the structure of security policy.

Findings

Bloomberg and Volpe (2012) state:

> You, the researcher, are the storyteller. Your goal is to tell a story that should be vivid and interesting while accurate and credible. In your report, the events, the people, and their words and actions are made explicit so the reader can experience the situation as real in a similar way to the researcher and experience the world of the participants. (p 148)

To that end, the population for this study was drawn from two LinkedIn℠ communities of interest: the CyberWarfare community and the DIACAP/FISMA/FEDRAMP community. Both of these communities have large subscriber bases of active practitioners in the field of cyber security. The demographics of the survey population can be found in Appendix K. For the

purposes of this study, there was no statistically significant difference among the subjects' basic background that would skew the results relative to interpretation of a requirements based system specification or a story-based specification.

From a demographic perspective, the average survey respondent has been working in the security field from 8-10 years, has at least a Bachelor's degree, and works in civil government, technology, or the aerospace industry. Forty percent (60 of the 150 respondents) hold at least one security certification, the most common being the Certified Information System Security Professional (CISSP). Twenty percent of the respondents are involved in system authorization and approval. Given the Federal Government's tiered model of certifications and years of experience acceptable for various career levels, these demographics are not remarkable.

Part Two of the survey provided definitional context to address security policy definitions. These five questions were designed to eliminate ambiguity in the context of access control. Subjects were asked to agree or disagree with the definition, and provided the opportunity to comment on their answers. Prior to the first question, the questions were caveated with the following paragraph:
"There are several types of access controls that can be used in cyberspace. We are not concerned with physical access controls for this research project, but with access controls in information systems." This caveat was incorporated into the survey to make it clear that the research was not about physical access management to a computer facility or data center, but about the access management that occurs within a computer system.

In general, the subject population agreed with the definitions as written. This is not surprising, since the terms and their definitions came from known, generally accepted glossaries of terminology such as the NIST Information Assurance Glossary. The percentage of each population group that agreed or disagreed with the respective definitions is illustrated in Tables 3, 4, 5, 6, and 7

Table 3. Definitional Question Results: Discretionary Access Control

Discretionary Access Control is a type of access control that restricts access to objects based on the identity of the subjects or groups to which the subjects belong. The access controls are discretionary because subjects with certain privileges are capable of passing those privileges to any other subjects, either directly or indirectly. Do you agree or disagree with this definition? If you disagree, please comment.						
Percentage of Respondents Replying						
	Arch. 1–R	Arch. 1-N	Arch. 2-R	Arch. 2-N	Arch.3-N	Arch.3-R
Agree	75	80	80	85	85	90
Disagree	25	20	20	15	15	10

Those that disagreed with the definition of discretionary access control commented about restricting access to objects based on the user's identity, given that the definition did not specify a unique user identity. Another comment was that "certain privileges" was ambiguous, especially if those privileges allowed the passing of access rights to others. An example was given of passing access to a member of a North Atlantic Treaty Organization (NATO) country, in that granting access to one country member of NATO usually means the entire Allied Forces would have access by the next day. Those that disagreed wanted a more explicit definition of the term to avoid ambiguity.

Table 4 presents the results of the definitional question about execution domains. As defined in the NIST Information Assurance Glossary, a domain is the cross product of the resources and users defined by the system security policy. For example, a trusted domain can be defined as the domain of intranet users, and an untrusted domain would be the set of extranet users. The trusted domain is defined by the access control boundary of an organization, for example, a firewall that is at the intranet/extranet boundary.

Table 4. Definitional Question Results: Domain

A domain is an environment or context that includes a set of system resources and a set of system entities that have the right to access the resources as defined by a common security policy, security model, or security architecture. Do you agree or disagree with this definition? If you disagree, please comment.						
Percentage of Respondents Replying						
	Arch. 1–R	Arch. 1-N	Arch. 2-R	Arch. 2-N	Arch.3-N	Arch.3-R
Agree	80	75	85	90	85	95
Disagree	20	25	15	10	15	5

Again, there was general agreement on the domain definition. There was some disagreement on the use of security policy or security model as the boundary definition; in that the terminology was not as explicitly specified as it would be in a security architecture. This would be a case where the subject may have been seeking less ambiguity in the definition, in which case the lower level of abstraction would be preferred.

Table 5 presents the results from the definitional question of Role-Based Access Control (RBAC). This question generated the greatest consensus, with over ninety percent of the respondents agreeing with the definition as stated. A large

percentage of the subject population is employed in the government and/or technology fields, where RBAC is widely used to facilitate system administration activities.

Table 5. Definitional Question Results: Role Based Access Control

Access control based on user roles (i.e., a collection of access authorizations a user receives based on an implicit or explicit assumption of a given role). Role permissions may be inherited through a role hierarchy and typically reflect the permissions needed to perform defined functions within an organization. A given role may apply to a single individual or to several individuals. Do you agree or disagree with this definition? If you disagree, please comment.						
Percentage of Respondents Replying						
	Arch. 1–R	Arch. 1-N	Arch. 2-R	Arch. 2-N	Arch.3-N	Arch.3-R
Agree	90	95	85	90	95	95
Disagree	10	5	15	10	5	5

Table 6 summarizes the definitional question on the term security policy model. This question generated the greatest disagreement percentages among the participants. Comments received on this definition were that the subject did not understand flow control, and that "other aspects of information security policy" should have been explicitly stated. The author believes the issue may be that a security policy model is usually an early system design artifact, and that several system architectures essentially reuse existing security policy models. Tailoring of abstract security policy models such as the Bell-LaPadula Model (Bell, D. & LaPadula, L., 1976) is considered a detailed design activity at a lower level of abstraction.

An alternative explanation may be found in the interpretation of the demographics of the population. Approximately 15 percent of the population

characterized itself as a security architect; another 15 percent consider themselves security engineers. That means 30 percent of the population routinely address system design and requirements as part of their assigned duties. The remaining population functions as security analysts, security operations personnel, and system administrators. In these job categories, the employee is responsible for the actual implementation of the security policy on given devices, such as firewalls, routers, and servers, or is responsible for locating policy breaches and reverse engineering these activities. As such, they are not usually concerned with the soundness of the policy model, but with the enforcement of the model as translated through the design abstractions.

Table 6. Definitional Question Results: Security Policy Model

An abstraction representing the basic properties or characteristics of an entity with respect to safeguarding information; typically associated with internal data structures (e.g., records, buffers, files), within the information system and used to enable the implementation of access control and flow control policies, reflect special dissemination, handling, or distribution instructions, or support other aspects of the information security policy. Do you agree or disagree with this definition? If you disagree, please comment.						
Percentage of Respondents Replying						
	Arch. 1–R	Arch. 1-N	Arch. 2-R	Arch. 2-N	Arch.3-N	Arch.3-R
Yes	70	75	65	70	80	75
No	30	25	35	30	20	25

Table 7 summarizes the findings on Mandatory Access Controls. There was minimal disagreement on the definition of mandatory or non-discretionary access controls. Comments made by the survey subjects asked for more explicit definitions of an authorized user and/or what was considered an access control. For example, if a user fails an identification and authentication challenge, this could be considered a form of

mandatory access control, in that the system would not necessarily allow the user to perform an activity. The example was provided of Amazon allowing anyone to browse the on-line store, but only an authenticated user being allowed to purchase merchandise.

Table 7. Definitional Question Results: Mandatory Access Control

Mandatory Access Controls or Non-discretionary access controls are access controls imposed by the information system that cannot be altered without explicit action from an authorized user. Do you agree or disagree with this definition? If you disagree, please comment.						
Percentage of Respondents Replying						
	Arch. 1–R	Arch. 1-N	Arch. 2-R	Arch. 2-N	Arch.3-N	Arch.3-R
Agree	85	80	90	90	85	80
Disagree	15	20	10	10	15	20

As in the definitional results presented in Table 6, the results for Table 7 may have more to do with the job functions of the respondents than their understanding of the terms. Operators and administrators do not usually think about policy as mandatory or discretionary; policy is policy and is enforced by the system. Especially at the less experienced career levels, the security policy is mostly an abstraction, a design concept and definition that is memorized for industry certification exams. There is very little thought about policy until one is asked to derive a policy based upon the documentation provided by a client, which in most organizations is an activity conducted by a security architect or a system security engineer.

Narrative v. Requirements Specification

The third and final section of the survey instrument addresses the question of traditional requirements specification language compared to narrative requirements.

For the purposes of comparison, Table 8 presents the two sets of specifications for Architecture 1 in their entirety side by side.

Table 8. Architecture 1, Requirements Specification v. Narrative Language

Requirements Specification Language	Narrative Language
System Description: For the next set of questions, the system is defined as an enterprise mail infrastructure. It supports a global Fortune 500 company with over 15,000 mail clients deployed. This infrastructure is used to support both internal and external electronic mail, and is directly connected to the Internet. For this discussion, we are only concerned about the access control policy. The system in question is implementing a new type of access control, risk adaptive access control, which considers the context of the system environment to make access control decisions. To send mail, a user shall be a registered user in the enterprise. A registered user shall be defined as a user with a unique identity. A registered user shall authenticate to the system with an authentication token. A mailbox shall be created for all registered users of the messaging system. The system shall allow for a maximum mailbox size of 200 MB. A warning message shall be generated if the mailbox size exceeds 150 MB. When the maximum mailbox size is reached, a user shall not be allowed to send messages until the mailbox size is reduced. Approval from the Messaging System Administrator shall be required to extend the maximum mailbox size beyond 200 MB.	System Description: For the next set of questions, the system is defined as an enterprise mail infrastructure. It supports a global Fortune 500 company with over 15,000 mail clients deployed. This infrastructure is used to support both internal and external electronic mail, and is directly connected to the Internet. For this discussion, we are only concerned about the access control policy. The system in question is implementing a new type of access control, risk adaptive access control, which considers the context of the system environment to make access control decisions. To send mail, a user must be registered in the enterprise, and have a unique user identity and authentication token as well as a mailbox. This is the non-discretionary access control policy of the messaging system. When a user joins the company, he is issued a unique user account and an authentication token so he can access the company systems. A mailbox is also created for him. When a user leaves the company, the user account is disabled, but the mailbox remains active. A maximum mailbox size is set, and if a user exceeds it, they cannot send or receive mail. Administrative assistants can send mail on the behalf of managers if they have been granted that privilege. A department manager travels a lot and

A user shall not send mail on the behalf of another user. A user can function in an administrative role and send mail on the behalf of another user provided the administrative role is bound to both messaging system users. Messages sent on the behalf of a user shall also be sent to the users normally in this role, as well as the intended recipient. The system shall deliver a message within 15 minutes of the user hitting "send." If the system cannot deliver a message, a notice of non-delivery shall be sent to the user, with an explanation of why the message was not delivered.	has his administrative assistant read and respond to his routine email so he doesn't have to worry about them while he is on the road. The administrator always sends a copy of the message to the boss so he knows what the message response was. The administrative assistant cannot delete messages from the manager's account.

Note that in general, the same basic information is conveyed in both variants. In the traditional requirements specification language, more specificity about sizes of mailbox and delivery times is provided. In the narrative specification, the requirements are less specific, and focus more on the rationale behind the policy, such as a manager is on the road and his administrative assistant deals with routine correspondence.

For the purpose of analysis, the questions were divided into smaller segments. Segment one addressed the basic access control policy, and whether it was sufficient. In the case of the traditional requirements specification subjects, 70 percent of the respondents did not consider the requirements to be a minimum sufficient set of access controls. The respondents wanted more specificity: how a user was registered into the system, what qualified as a unique identity, what was an acceptable authentication token, and when a user was considered registered. In contrast, 65

percent of the narrative respondents believed the set of requirements was minimally sufficient. The 35 percent that did not consider the requirements to be sufficient echoed the comments of the narrative respondents in that more specificity was required.

The requirements specification discussion on maximum size of mailbox (Question 14 in the survey) was considered a valid set of requirements for a messaging system. Some commented on the inclusion of a role based access control mechanism incorporated into this question, in that approval from the Messaging System Administrator was required if the maximum mailbox size needed to be increased. In the narrative specification, the majority of respondents considered the narrative a sufficient set of access control requirements.

Questions 15, 16, and 17 were considered control questions. The same question was asked in both versions of the survey instrument. Question 15 resulted in 80 percent of respondents selecting the answer "no messages can be sent or received." This is the correct response based upon the information presented. Question 16 asks how many messages are sent when the administrative assistant sends a message for the department manager. The information provided indicates a correct answer is two, one to the recipient and one to the department manager. In reality, 35 percent of the respondents selected the answer that three messages were sent, citing the retention of a message in the administrative assistants' sent items mailbox. It should be noted that not all mail systems maintain a sent items mailbox.

The last question in this survey posed an interesting challenge to the respondents: "A user leaves the organization. Can the user still send email?" 85

percent of the narrative respondents said the user could not send email, because the user would not be registered within the enterprise. The requirements specification respondents did not agree, and 50 percent found the question ambiguous. The user could still send email, simply not from that email system within the enterprise. There were no restrictions on his ability to send email from alternative accounts, such as a personal email account.

The second survey architecture represents a network infrastructure, a considerably more complex system architecture example. While most system users are quite familiar with electronic mail, fewer understand how network infrastructure delivers packets, and how many defensive mechanisms are employed to reduce the probability of a successful attack. Table 9 presents the alternate specifications.

Table 9: Alternate Specifications, Network Infrastructure

Requirements Specification	Narrative Specification
For the next section, the system in question is a network infrastructure. The network infrastructure is responsible for connecting over 15,000 individual sites for a mission critical system control application. The network terminates at the service delivery point, which is similar to the cable service provider termination at a personal residence. Network device health and status information shall be segregated from the user data within the network. The network and security management node shall enforce the concept of least privilege. Two man control (security and network administrators) shall be required to add a component to the infrastructure: no one person can completely register a device on the network and configure it.	For the next section, the system in question is a network infrastructure. The network infrastructure is responsible for connecting over 15,000 individual sites for a mission critical system control application. In this architecture, the network device health and status information is segregated from the device data streams. That is, information on the health and status of the network is transmitted to a management node, while information used by the network applications is transmitted to analytical nodes for processing and visualization. The management node enforces least privilege and two man controls on the infrastructure components; no one person can completely register a device on the network and configure it. The networks terminate at service delivery points

	(SDPs) that are not necessarily end user servers or workstations.

For this set of analysis, the emphasis was on the notion of two-man control and separation of duty policies. When a security administrator calls in sick, there is only one administrator working, a network administrator. The question is raised as to whether the network administrator can completely configure a device and place it in operation. The majority (70%) of respondents said the answer was NO in both the narrative and the requirements specification survey instruments. When the YES answer was provided, it was expressed as a special circumstance, such as a device in need of repair or replacement on an emergency basis. It was not a question of whether or not the network administrator had the knowledge, it was a question of did the network administrator have the permissions to configure the security functions of the device. Upon further analysis, these respondents were almost exclusively security operations personnel, those who are responsible for the daily maintenance of a system. In these cases, actual experience outweighed the statement of security policy.

The next two questions in this survey focused on actual duties performed. In both surveys, 65% of the respondents correctly categorized job functions. Network routing tables are normally considered part of network administration duties, and audit log monitoring is part of security administration duties. Those who selected minority answers (that audit logs are part of network administration duties and routing table establishment is part of security administration duties) made those selections based on their individual experiences. There are systems where security

auditing is not implemented, but OpenFlow traffic monitoring is used to manage network bandwidth. Additionally, there are networks where routing tables are considered sensitive information; therefore routing table management could be considered a security administration function. Again, it is more an environmentally based answer than an answer that has its basis in written policy.

Finally, the question of patch management is presented in the final question on this survey instrument. The simple statement "Vulnerability remediation (patch management) fixes vulnerabilities in the infrastructure" is presented. The subject is asked to determine which job function is the best fit for this duty. Of the requirements specification respondents, 70 percent placed vulnerability remediation as a security function. Narrative respondents were less definitive, with 50% commenting that the function could be a network administration function, depending upon the vulnerability in question. One could state that the narrative respondents took a more collaborative approach to the task than a strict separation of duty model would indicate.

Finally, we examine the third architecture, a service-oriented architecture with a content creator/content subscriber model that crosses a trusted boundary. The two system specifications are presented in Table 10.

Table 10. – Alternate Specifications, Publish Subscribe Model.

Requirements Specification	Narrative Specification
For the next set of questions, the system shall support a content creator/content subscriber model. All content created within an enterprise shall be stored in a data repository. When new content is available, the subscriber shall be notified. The information broker validates information flow as illustrated in Figure 23.	For the next set of questions, the system is defined as a content creator/content subscriber model. All content is created within an enterprise and stored in a data repository. When new content is available, subscribers are notified. The information broker validates that the subscriber is within the enterprise, or an external subscriber. If the subscriber is an external subscriber, the information is pushed through a firewall and made available on an outwardly facing message bus. No information is passed back through the firewall to the internal message bus. An external subscriber can collect information from the external message bus only, and cannot alter the original content of the internal message bus

Architecture Three provides the highest degree of complexity, in that there is cross-domain information flow between trusted and untrusted subscribers. In the constraints of this study, the respondents that received the requirements specification form picked more constrained responses. For example, when asked how many types of subscribers existed, 85 percent of requirements specification respondents picked two: trusted and untrusted. Of the narrative respondents, there were a higher percentage of unknown responses, 45 percent. The comments were that there was not information to determine the types of subscribers, and that more information was needed about the content to determine the subscriber type.

The remaining questions had a distinct and similar pattern: the majority of narrative respondents (averaging 80% over the four questions) were able to select answers to the questions consistent with the security policy described in the narrative. One could state that these respondents had the benefit of additional narration that was not provided to the requirements specification respondents. Alternatively, the requirements specification respondents had the benefit of a system diagram illustrating information flow across the domains. With the information provided, it would appear that the narrative specification respondents were able to more precisely place the system security policy within the context of prohibited/allowed user behaviors than a system diagram would allow.

Summary

In this section, we have discussed the criterion of a grounded theory methodology experiment, and illustrated how this experiment conforms to the criteria described in (Glaser & Strauss, 1968). The experiment is not a strict interpretation of grounded theory methodology, which, in the constructs of Bloomberg and Volpe (2012) , involve small populations, repeated interviews, and shared experiences among the population. Using this strict interpretation of qualitative experiment design would have minimized the population size, and sacrificed anonymity of the user population. It also would have proven manpower intensive to perform one-on-one interviews with 150 experimental subjects.

The initial experimental objective, that narrative could be applied more effectively as a requirements language than traditional specification language was successfully illustrated: as the systems became more complex the narrative

specifications appeared to deliver a more precise model of system behavior as defined in the narrative specification and derived from the question responses. The use of an anonymous survey as a measuring instrument may lack a preferred depth of data in that subjects are not interviewed; the survey does provide sufficient contextual information to infer correct user behaviors from the described security policy.

What does this indicate for the experiment's hypothesis, that security policies can be explained in relatively simple English and explicitly traced to lower-level implementations? It appears that simple English, logically and explicitly described, provides a sufficient system requirements specification when compared to traditional requirements specification languages. That is, a top-level narrative can be traced to lower level abstractions to define appropriate system activities comparable to the policy objectives one would see in a more formal specification language.

Chapter 5

CONCLUSIONS

This chapter places the experimental findings within the context of current research, and presents thoughts for further study.

Implications

Reeder, et al (2007) states:

> Security and privacy management tasks were previously left to expert system administrators who could invest the time to learn and use complex user interfaces, but now these tasks are increasingly left to end-users. Two non-expert groups of policy authors are on the rise. First are non-technical enterprise policy authors, typically lawyers or business executives, who have the responsibility to write policies governing an enterprise's handling of personal information (Karat, J., Karat C.-M, Brodie, C., & Feng, J, 2005). Second are end-users, such as those who wish to set up their own spam filters, share files with friends but protect them from unwanted access (Cao, X., & Iverson, L, 2006) (Good, N.S. & Krekelberg, A., 2003) (Maxion, R.A. & Reeder, R.W., 2005), or share shipping information with Web merchants while maintaining privacy (Cranor, L.F., Guduru, P., Arjula, J., 2006).

As the use of information systems has spread throughout society, the need for less expert policy interfaces has increased. One can no longer assume that accomplished system administrators are formulating policy and propagating its enforcement throughout the user's environment. Today, a casual information technology user can maintain a variety of devices such as smartphones, tablets, laptops, desktops, and file servers to support ubiquitous connectivity. Yet the security policy creation environments across these devices are not consistent or user friendly. Rather, security by obscurity continues to be the rule rather than exception.

There have been attempts to define structured policy authoring environments. LeMay, Fatemieh, and Gunter (2007) state that "many of the challenges that arise during the development and maintenance of an access control policy are caused by the inability of the policy administrator to correctly translate high-level business requirements into low-level access control policies that can be implemented in an Access Decision Function (ADF)(p. 205)." There has also been research on the subject of making policy authoring domain friendly, in that different user environments use different terminology, with a hierarchy of direction implicit in the terms (Johnson, Karat, Karat, Grueneberg). For example, in the medical profession, directions written by a doctor take precedence over those written by a nurse. While these attempts to define and refine policy are useful, they apply a very different approach: one of bottoms-up policy refinement. That is, security policy is defined within the context of the devices and/or applications being used to enforce it. It is not authored as a top-down function of an information system. While it is useful to understand how policy is implemented within a complex system, it is not necessarily the best approach to policy when dealing with non-expert system users.

Our research has taken a top down approach to security policy, treating security policy as a top-level system requirement that must be allocated throughout the design process. With a complex system architecture, this is the preferred approach to system design, in that functionality is allocated to specific components, and, as technology evolves, replacement elements duplicate or improve upon the functionality allocated to a specific element.

Research to date on policy authoring has addressed hands-on observation of 20-30 subjects and focused on how the subjects performed the task, what errors they commonly made, and how to correct those errors (Johnson, et al, 2010), (Reeder, et. al, 2007). The policy authoring tools developed used constrained language subsets or templates manifested as pull-down menus.

Instead of making the user adapt to a new syntax and subset of language, we have chosen to explore the use of traditional English language and the constructs of storytelling and narrative decomposition as an alternative. In our study, we have synthesized the ongoing research in neuroscience, computer assisted storytelling, and narrative understanding to present a top down approach to security policy. Further, through the application of an adapted grounded theory methodology, we were able to do so with a somewhat larger sample of participants than previous studies have accommodated.

While our sample size was larger, we also had to revise the study protocol to adapt to the data delivery method. To accommodate an on-line survey instrument, one must adapt to the user interface conventions of web-based access methods: namely, that users do not read long narratives online. This modification to the original experiment design forced the data to be delivered in smaller "vignettes", addressing single topic areas in each question set as opposed to the complex system requirements.

Recommendations

There are several areas for research refinement that could be addressed in future research.

User population description

One area we would have liked to address is the potential difference in policy specification from the casual system user's perspective. The subject pool for the experiment was a pool of experienced security practitioners: not a pool of general computer users. This was a question of available and cooperative subjects to perform the experiment. The author had access to security practitioners who had practical work experience, and did not have access to a general population of casual computer users. In one respect, this simplified the study, in that the user population understood the fundamental concepts of security policy. On the other hand, this complicated the study, because definitional questions were added to ensure all practitioners had a relatively common understanding of the terms used in the specifications. A general population of "casual" computer users would be difficult to qualify. With the proliferation of tablet computers, notebooks, "smart phones" and gaming devices, a common set of definitions and/or experiences might make the demographics and categorization of such a group problematic.

Further research must be conducted to determine if this type of experiment is feasible for the general class of system users with larger sample sizes. This may be the only way to determine if simple security policies can be understood and subsequently enforced upon larger user communities. Locating a collection of willing participants that do not reflect a bias because of their employer or student status will make selection of such a student pool more difficult. Such experimentation may be better focused in the realm of product manufacturing. Cell phones and tablet computers tailored to the elderly and the preschool markets have

begun to come to market. It would not be far fetched for a more simplistic user interface to abstract the operating system's complexities from the end user: witness the popularity of windows-oriented interfaces over character-based interfaces.

Description of general user situations

The study addressed three distinct system architectures: enterprise e-mail, network infrastructure, and a publish-subscribe cross-domain solution. These types of architectures are representative of the types of general complex system problems that exist in current information technology environments. However, with the exception of e-mail, they are not the types of systems that end users encounter every day. Most of the activity in complex system architectures is transparent to the end-user. That is, the systems deliver information to the end user subject to the user's operational constraints and the system's policy enforcement capabilities.

The end user does not see the system components of web-based electronic purchase transactions such as inventory management, order fulfillment, and payment processing. The results of those functions are delivered to the end user, but are not presented in detail. For example, the user may see a message that a credit card was rejected. The details of the rejection process and its workflow are not presented to the end user; only a message to contact the issuing organization is displayed.

A series of usage scenarios needs to be created to determine the applicability of this methodology to the casual system user. In these cases, the security policy and its enforcement is transparent to the system user, who is not provided sufficient information to determine why his request was not fulfilled. While this is deliberate to avoid sharing information that could be used to compromise the system, it does

distance the user from understanding the role of security policy enforcement. If the user does not understand that a policy exists the user could not be expected to understand its enforcement.

Application of Ontological Based Deconstruction

As the technologies for natural language have matured, the ontologies that support natural language processing have become more robust. One of the constraining functions of computer-assisted storytelling has been the ability of the computer to understand the user's contextual information. For example, the number "42" in most contexts is just a random number; to those who have read "The Hitchhiker's Guide to the Galaxy" the number 42 is the meaning of life, the universe, and everything.

With the emergence of large storage area networks, it is possible for computers to rapidly search large quantities of data and determine the probability a given fact fits a given user context. For example, when IBM's Watson architecture played Jeopardy!, Watson did extremely well on questions that required rapid fact-based searches. It was less successful on questions that required some degree of background or contextual information. This is consistent with security policy enforcement as well: binary answers are easy; answers that depend on a collection of circumstances are more difficult to determine.

Raskin et al (2001) developed a preliminary ontology of information security terms. While this ontology presented a synthesis of cryptographic terms and information security terms, it predates a large body of technology and terminology that has become commonplace. For example, tablet computing, notebooks, and cloud

computing did not exist at the time the Raskin ontology was created. As such, the development of security-specific ontologies must also address the current and emerging technology baseline. Otherwise, security will continue to be an afterthought in the system development process and lag behind in technology innovation and acceptance.

A security specific ontology must address information sharing mechanisms and unique user identification and authorization techniques. Without the language constructs to address constrained information sharing, identity management, and information provenance, there will be little functional enforcement of theoretical policy constructs. This will become an increasing problem as the "Internet of Things" emerges and temporary social media become more prominent in our quest for connectedness.

For example, a given hotel chain may offer a "lobby network" for registered guests to help them find dinner partners in a strange town. Participation in such a network may be voluntary, but would only be available to paying guests. Specification of the constraints associated with such social information sharing would need to address both membership in the group of registered guests and the location of the hotel. When a guest checks out, the hotel must remove his access to the group, or remove him from the location after his anticipated stay is over. Such constraints can be gleaned from the user check-in experience, but only if the capability is integrated into the guest registration and billing infrastructure.

As another example, consider the electronic health record. Today's medical technology allows home-based monitoring of various medical information types that

are valuable to the treating physician. This information can be stored and used to create general trending data to adjust medication and treatment plans for patients. The devices, the information stored, the patient, and the physician all have some access rights to this information. The question is how to limit those rights within the constraints of patient privileged information and patient identity. As an example, there are instances when reporting of infectious diseases such as influenza to public health officials is in the interests of the common good, and such data is usually reported without patient identification. Again, definition of the information sharing policy must be simple enough to be understood, but complex enough to address the potential usage or value chain associated with the information.

The Transitivity of Security Policy Enforcement

One of the key features of a partially ordered set representation of security policy is the transitive nature of security policy enforcement. That is, the results are the same if $(a + b) + c$ or if $a + (b + c)$. As we move towards risk adaptive access controls, the chain of variables increases: $(a + b) + (c + d) + e$. As the number of variables increases, the results of policy decisions may result in inconsistencies based on the order of enforcement or evaluation. Risk Adaptive Access Control mechanisms are just emerging from the research laboratories, and have not been implemented in a commercial-off-the-shelf product architecture to date. More experimentation with both the policy representation and the order of evaluation must be accomplished with multi-variant security policy logic. These activities are essential as we move towards autonomic systems, where the system is responsible for security policy enforcement and takes action without human intervention. As the

Internet of Things (IoT) emerges, sensor networks will incorporate logic to take actions when specific threshold values are reached. In these cases, a system may shut down, throttle back production, or decrease network bandwidth based upon the programmed policies. If the results of policy evaluation are not transitive, then the failure conditions may not be consistent. If this is the case, a system may be shut down or removed from service based upon false information, or, worst case, lives may be lost because the system believed everything was fine. Experimentation is necessary to determine both policy correctness and operational policy implementation correctness. There is considerable room for error between a logical policy model and the software that embodies that policy model in an operational system.

Evolution of Storytelling Technologies

In "Do Story Agents Use Rocking Chairs", Brooks (1996) hypothesized a computer-generated storytelling architecture and stated that it was going to take a considerable amount of effort to make such a system a reality. The question of automated storytelling has been deconstructed into a series of questions about contextual information: what a given phrase means within a given context and how the language is arranged within a sentence. With the advent of Attribute Based Access Control (ABAC) models, we have the capability to describe the access control related context of a given object in a standardized context. What remains to be accomplished is the integration of security context with storytelling capabilities to determine if the actions are consistent with the user's intent. In the non-cyber world it is relatively easy to determine if a user has violated access controls: physical alarms go off, there are signs of forced entry, and items are missing. In the cyber

world, data can be missing and the owner may not be aware that it has been taken, or worse, altered in some way that would lead to a wrong conclusion. The complexity of system architectures and network infrastructures add to the complexity of security policies in that there are that many more devices that have the opportunity to alter the security policy or misinterpret it in transit. For example, an application layer firewall may prohibit communication that should be permitted because a particular protocol is not supported. Considerably more research must be conducted in this area to determine if a fully automated security policy generation environment can be created.

Summary

In conclusion, we have presented the case for policy elicitation through the use of structured storytelling as a less intimidating, more descriptive technique that would lend itself to contextual security information such as that required for risk adaptive access control model. We believe this would be a more comprehensible system security policy definition; one that could elicit the information required for various security policy models and supports a degree of formalism that would be amenable to formal methods and logical proof if desired.

Our review of the literature surveyed a large and diverse set of security policies that have formal model as well as several policy authoring tools that explore templates, constrained natural languages, and graphical authoring environments. Further, we presented a collection of analytical tools to disassemble narrative text and story elements into their basic components for textual analysis. The policies discussed are highly formal, the policy authoring environments constrain the users

ability to select in syntax and expression, and the narrative analysis tools have not been applied to security policies.

The experimental discussion presented an overview of the issues associated with security policy specification and logical correspondence. Security policies have become more expressive with additional constraints and rule-based capabilities, which have made reaching the eventual binary access control decision a more complex task. Additionally, the need to provide dynamic security policies in the event of cyber attack scenarios means policy modifications must be rapidly propagated throughout an infrastructure. This does not leave time to resolve security policy conflicts when an entire mission critical network may be compromised.

As an alternative approach, we offer the specification of security policy through storytelling. Stories provoke discussion, help our collective memory recall the success or failure of past policy attempts, and encourage alternative solutions. As security policies become more robust in their emulation of the "real world, " it will become more difficult to prove logical soundness with existing formal methods. Given that the ability to perform natural language processing is improving, it may be more feasible to perform policy analysis through the use of narratology. The use of computer-assisted narrative for policy management is an area that has not been explored to date.

Further research in ontologies and natural language authoring environments needs to be conducted to determine if policies can be expressed and analyzed with these capabilities. Mankind shares a common social history in story; our cultural differences change the context. As our social media converges in the global

communications network, it will be interesting to determine if our stories and our policies converge as well.

Our experiment adapted the qualitative grounded theory methodology to the specification of security policies. We demonstrated correspondence to Glaser and Strauss' criteria for a grounded theory experiment (1968). The experiment is not a strict interpretation of grounded theory methodology, which, in the constructs of Bloomberg and Volpe (2012), involves small populations, repeated interviews, and shared experiences among the population. Using this strict interpretation of qualitative experiment design would have minimized the population size, and sacrificed anonymity of the user population. It also would have proven manpower intensive to perform one-on-one interviews with 150 experimental subjects.

The initial experimental objective, that narrative could be applied more effectively as a requirements language than traditional specification language was successfully illustrated. As the system architectures became more complex the narrative specifications appeared to deliver a more precise model of system behavior, as defined in the narrative specification and derived from the question responses. The use of an anonymous survey as a measuring instrument may lack a preferred depth of data in that subjects are not interviewed; the survey does provide sufficient contextual information about the system to infer correct user behaviors from the described security policy. While system requirements specification language also describes system behavior, in our results the requirements specification respondents wanted additional detail to remove ambiguity from the requirements.

What does this indicate for the experiment's hypothesis, that security policies can be explained in relatively simple English and explicitly traced to lower-level implementations? It appears that simple English, logically and explicitly described, can provide a sufficient system requirements specification when compared to traditional requirements specification languages. That is, a top-level narrative can be traced to lower level abstractions defining appropriate system activities comparable to the policy objectives one would see in a more formal specification language.

In the early stages of this dissertation, the vision was the adaptation of a natural language storytelling environment to the creation of security policy. As the research survey continued, it was apparent that machine-assisted storytelling was in its infancy. There are writer's aids that assist in defining plot and story characters, but none that can comprehensively address the dimensions of a security specification. Using storytelling to address the nineteen requirements families covered in NIST's Security Controls Catalog would require authoring a large book. It may be more effective to model security policy in a game based simulation, where information context can be visualized.

In today's computer system environment, security policy can be enforced at any point from the user's device back to the storage area network. Each device has its own syntax, it's own commands, and it's own enforcement mechanisms. The integrated totality of the protection mechanisms is the responsibility of the system architect, who is ultimately responsible for allocating portions of the policy to the various devices. The security architect is responsible for translation of the security

policy at the policy definition point interface to the system. Until the policy definition point can be simplified, the users of computer systems will depend upon the security decisions of the system architects to provide the protections required for system security and integrity.

Appendix A

Preliminary Glossary of Information Security Terminology

The following terms were proposed in (Raskin, Hempelmann, Triezenberg, and Nirenburg, 2001) as a preliminary attempt to reconcile the terminology of the security community with the ontologies under development at that time. To the author's knowledge, the list has not been updated or published since the original paper.

Absolute rate	AS-400	Boot sector virus
Access control	Associativity	Bootstrap virus
Access control list	Assurance	Bounds register
Access control matrix	Assymetric encryption	Break
Access log	Attack	Brute force attack
Access triple	Attribute	Buffer
Accountability	Audit	Buffer overflow
Accuracy	Audit log	Caesar cipher
Address	Audit operations	Call bracket
Adjudicable	Audit options	Capability
Aggregate query	Authenticate	Career criminal
Aggressive scheduler	Authentication	Category
Algorithm	Authenticity	CERT
Amateur	Automatic retaliation	Certificate
Analog	Availability	Certificate distribution center
Analyzability	Backdoor	Certificate revocation list
Anklebiter	Backup	Certification authority
Anonymity	Base register	Certified code
Applet	Bastion host	Certified mail
Arbiter	Block cipher	CGI script

Change log
Channel
Checksum
Chinese wall policy
Chinese Wall Model
Cipher
Cipher block chain
Ciphertext
Classification
Clearance
Client
Clique problem
Code
Collision
Columnar transposition
Commit
Commitment
Common Criteria
Commutativity
Compartment
Complexity
Composite
Compression
Computing system
Conceal
Concurrency-control
Confidentiality
Configuration management
Confusion
Connectivity
Conservative scheduler
Constrained data item
Contract signing
Control
Controlled sharing
Cookie
Copy
Copyright
CORBA
Core
Core dump
Correct
Coupling
Cover story
Covert
Covert channel
Covert timing channel
Cracker
Credentials
Criteria creep
Cryptanalysis
Cryptanalyst
Cryptography
Cryptology
Cryptosystem
Cycle
Data
Data encryption standaard
Database
Database management system
Datagram
Decideability
Decipher
Decode
Decrypt
Degausser
Dependability
Diagram
Diffusion
Digest
Digital
Digital signature
Digital signature scheme
Directory
Disaster
Disclosure
Distributivity
Divisible by

Domain
Dominance
Dongle
Double transposition
Driver
Effectively secure
Effectiveness
Egoism
Egoless programming
Electronic code book mode
Element
Encapsulation
Encipher
Encode
Encrypt
Equivalent
Error code
Error propagation
Ethic
Etiquette
Evaluation
Evidence
Executive
Exhaustive attack
Expandability
Exposure
Fabrication
Fair use
Fairness
Fence register
Field
Field check
File protection
Filter
Fire
Firewall
Flood
Flooding
Frequency distribution
Front end
Guard
Guest
Hack
Hardware
Hash
Heat
Hierarchy
Host
Identity
Impersonate
Index of coincidence
Inductance
Inference
Information
Information hiding
Information leak
Integrity
Intercept
Internal consistency
Interpretation drift
Interruption
Intruder
Inverse divide
Inverse mod
Isolation
Join
Kasiski method
Kernel
Key
Key distribution server
Keyless cipher
Knapsack
Lattice model
Layering
Least privilege

License
Limited privilege
Link
Local name space
Logic
Logic analyzer
Logic bomb
Lucifer
Macro
Macro virus
Maintain
Malicious code
Master key
Measure of roughness
Mechanism
Memory-resident virus
Mental poker
Message digest
Microwave
Modern
Modification
Modular arithmetic
Module
Modulus
Monitor
Monoalphabetic cipher
Multiplex
Mutual suspicion
Need-to-know
Network
Node
Nondeterminism
Notarization
Notary
Novelty
Nucleus
Object
Object request broker
Oblivious transfer
One-time
Open design
Optical fiber
Oracle machine
Originality
Packet
Packet sniffer
Paging
Parasitic virus
Parity
Password
Patent
Payload
Peer code review
Peer design review
Permission
Permutation
PGP (Pretty Good Privacy)
Physical
Plaintext
Policy
Polyalphabetic cipher
Polymorphic (virus)
Polynomial
Port
Precise
Prime number
Privacy
Probably password
Problem
Product cipher
Program
Project
Property
Protect
Protected object
Protocol
Query
Rabbit

Random access memory
Read only memory
Receiver
Record
Recover
Reducibility
Redundancy
Relation
Relative prime
Reliable
Religion
Relocation
Repeater
Replay
Resident virus
Resource
Reuse
Reverse engineer
Ring bracket
Risk
Rogue program
Routing
Salami attack
Satellite
Satisfiability problem
Schema
Secrecy
Secure
Security audit
Segment
Segmentation
Self-enforcing protocol
Semantic sugar
Sender
Sensitive
Sensitive data
Separation
Server
Service program
Session
Session key
Shadow program copy
Shared file
Shared resource matrix
Shell theft
Shredder
Shrink wrapped software
Side effect
Simple substitution
Single-user system
Socket
Software
Solvable problem
Spoof
Stream cipher
Stub
Subject
Subscheme
Substitutions
Suppress
Surge
Symmetric
Symmetric key exchange
Tamper
Tamperproofness
Target
Temporal
Terminal
Test
Theft
Time bomb
Time stamp
Topology
Trade secret
Traffic key
Transformation procedure
Transient virus
Transmission medium

Transposition	Vernam cipher
Trapdoor	View
Trigram	Vigenere tableau
Tripwire	Virtual
Trojan horse	Virtualization
Trusted	Virus
Unbypassability	Virus scanner
Unconditionally secure	Virus signature
Understand	Vulnerability
Unicity distance	Window
Unix	Wiretap
Usage restriction	Workstation
User	Worm
Validation	Write-down
Verification	

Appendix B

Institutional Review Board Certifications

NOVA SOUTHEASTERN UNIVERSITY
Office of Grants and Contracts
Institutional Review Board

MEMORANDUM

To: Ronda Henning
Date: May 22, 2013

Re: *Security Policies that Make Sense for Complex Systems: Comprehensible Formalism for the System Consumer*

IRB Approval Number: wang05151302

I have reviewed the above-referenced research protocol at the center level. Based on the information provided, I have determined that this study is exempt from further IRB review. You may proceed with your study as described to the IRB. As principal investigator, you must adhere to the following requirements:

1) CONSENT: If recruitment procedures include consent forms these must be obtained in such a manner that they are clearly understood by the subjects and the process affords subjects the opportunity to ask questions, obtain detailed answers from those directly involved in the research, and have sufficient time to consider their participation after they have been provided this information. The subjects must be given a copy of the signed consent document, and a copy must be placed in a secure file separate from de-identified participant information. Record of informed consent must be retained for a minimum of three years from the conclusion of the study.

2) ADVERSE REACTIONS: The principal investigator is required to notify the IRB chair and me (954-262-5369 and 954-262-2020 respectively) of any adverse reactions or unanticipated events that may develop as a result of this study. Reactions or events may include, but are not limited to, injury, depression as a result of participation in the study, life-threatening situation, death, or loss of confidentiality/anonymity of subject. Approval may be withdrawn if the problem is serious.

3) AMENDMENTS: Any changes in the study (e.g., procedures, number or types of subjects, consent forms, investigators, etc.) must be approved by the IRB prior to implementation. Please be advised that changes in a study may require further review depending on the nature of the change. Please contact me with any questions regarding amendments or changes to your study.

The NSU IRB is in compliance with the requirements for the protection of human subjects prescribed in Part 46 of Title 45 of the Code of Federal Regulations (45 CFR 46) revised June 18, 1991.

Cc: Protocol File

From: Ling Wang, Ph.D.
Institutional Review Board

Security Policy Definition

Consent Form

The purpose of this research project is to examine access control policy specifications and determine which types are most readily understood. This is a research project being conducted in the Graduate School of Computer and Information Systems at Nova Southeastern University. You are invited to participated in this research project because you are a member of the information security community.

Your participation in this research study is voluntary. You may choose not to participate. If you decide to participate in this research study, you may withdraw at any time. If you decide not to participate in this study or if you withdraw at any time, you will not be penalized.

The procedure involves completing an online survey that will take approximately 25 minutes. Your responses will be confidential and no identifying information such as your name, email, or IP address is collected.

We will make our best efforts to keep your information confidential. All data is stored in a file that requires user identification and authentication for access. To further protect your information, the survey does not contain information that will personally identify you. The results of this study will be used for scholarly purposes only and may be shared with Nova Southeastern University representatives.

If you have any questions about the research study, please contact hronda@nova.edu. This research has been reviewed according to Nova Southeastern University IRB procedures for research involving human subjects.

1. ELECTRONIC CONSENT: Please select your choice below.

Clicking on the "agree" button below indicates that:

You have read the above information

You voluntarily agree to participate

You are at least 18 years of age.

If you do not wish to participate in the research study, please decline participation by clicking on the "disagree" button.

- ○ Agree
- ○ Disagree

CITI Collaborative Institutional Training Initiative

Human Research Curriculum Completion Report
Printed on 4/8/2013

Learner: Ronda Henning (username: rondahenning)
Institution: Nova Southeastern University
Contact Information 569 Lake Ashley Circle
W. Melbourne, FL 32904 U.S.A.
Department: SCIS
Phone: 321-795-9305
Email: hronda@nova.edu

6. SCIS:

Stage 1. Basic Course Passed on 04/08/13 (Ref # 10131769)

Required Modules	Date Completed	Score
Introduction	04/08/13	no quiz
History and Ethical Principles - SBR	04/08/13	4/5 (80%)
Defining Research with Human Subjects - SBR	04/08/13	4/5 (80%)
The Regulations and The Social and Behavioral Sciences - SBR	04/08/13	5/5 (100%)
Assessing Risk in Social and Behavioral Sciences - SBR	04/08/13	5/5 (100%)
Informed Consent - SBR	04/08/13	5/5 (100%)
Privacy and Confidentiality - SBR	04/08/13	5/5 (100%)
Internet Research - SBR	04/08/13	5/5 (100%)
Nova Southeastern University	04/08/13	no quiz

For this Completion Report to be valid, the learner listed above must be affiliated with a CITI participating institution. Falsified information and unauthorized use of the CITI course site is unethical, and may be considered scientific misconduct by your institution.

Paul Braunschweiger Ph.D.
Professor, University of Miami
Director Office of Research Education
CITI Course Coordinator

Return

Appendix C

Demographic Questions

Security Policy Definition

***2. Demographic Information**

How long have you been an information security practitioner?

- ☐ 0-3 years
- ☐ 3-5 years
- ☐ 5-8 years
- ☐ 8-15 years
- ☐ over 15 years

3. What is the highest level of education you have completed?

- ○ Did not attend school
- ○ Graduated from high school
- ○ 1 year of college
- ○ 2 years of college
- ○ 3 years of college
- ○ Graduated from college
- ○ Master's Degree
- ○ Ph.D. (or equivalent)

4. Do you have any industry security certifications?

- ○ Yes
- ○ No

Security Policy Definition

5. If so, which (check all that apply)

- [ ] CISSP
- [ ] CISSP-ISSAP
- [ ] CISSP-ISSEP
- [ ] CISSP-ISSMP
- [ ] CSSLCP
- [ ] CISM
- [ ] CISA
- [ ] CGEIT
- [ ] CRISC
- [ ] Security+
- [ ] IAM
- [ ] IEM
- [ ] Other (please specify)

6. Which of the following best describes your job function?

- ○ Security Analyst
- ○ Security Engineer
- ○ Security Architect
- ○ Security Certification and Accreditation
- ○ Security Operations
- ○ Security Administrator
- ○ Security Manager
- ○ Security Educator
- ○ Security Researcher
- ○ Other (please specify)

Security Policy Definition

7. What is the principal industry of your organization?

- ○ Advertising & Marketing
- ○ Agriculture
- ○ Airlines & Aerospace (including Defense)
- ○ Automotive
- ○ Business Support & Logistics
- ○ Construction, Machinery, and Homes
- ○ Education
- ○ Entertainment & Leisure
- ○ Finance & Financial Services
- ○ Food & Beverage
- ○ Government
- ○ Healthcare & Pharmaceuticals
- ○ Insurance
- ○ Manufacturing
- ○ Nonprofit
- ○ Retail & Consumer Durables
- ○ Real Estate
- ○ Telecommunications, Technology, Internet & Electronics
- ○ Utilities, Energy, and Extraction

Appendix D

Definitional Questions

Security Policy Definition

Definitional Information

This portion of the survey explores definitions and types of access controls.

There are several types of access controls that can be used in cyberspace. We are not concerned with physical access controls for this research project, but with access controls in information systems.

8. Discretionary Access Control is a type of access control that restricts access to objects based on the identity of the subjects or groups to which subjects belong. The access controls are discretionary because subjects with certain privileges are capable of passing those privileges on to any other subjects, either directly or indirectly. Do you agree or disagree with this definition. If you disagree, please comment.

- Agree
- Disagree

Comment

9. A domain is an environment or context that includes a set of system resources and a set of system entities that have the right to access the resources as defined by a common security policy, security model, or security architecture. Do you agree or disagree with this definition? If you disagree, please comment

- Agree
- Disagree
- Comment

Please comment if you disagree

Security Policy Definition

10. Access control based on user roles (i.e., a collection of access authorizations a user receives based on an explicit or implicit assumption of a given role). Role permissions may be inherited through a role hierarchy and typically reflect the permissions needed to perform defined functions within an organization. A given role may apply to a single individual or to several individuals. Do you agree or disagree with this definition? Please explain if you disagree.

- Agree
- Disagree

Please comment if you disagree

11. An abstraction representing the basic properties or characteristics of an entity with respect to safeguarding information; typically associated with internal data structures (e.g., records, buffers, files), within the information system and used to enable the implementation of access control and flow control policies, reflect special dissemination, handling, or distribution instructions, or support other aspects of the information security policy. Do you agree or disagree with this definition? If you disagree, please comment.

- Agree
- Disagree

Please comment if you disagree

12. Mandatory Access Controls or Non discretionary access controls are access controls imposed by the information system that cannot be altered without explicit action from an authorized user. Do you agree or disagree with this definition. If you disagree, please comment.

- Agree
- Disagree

Please comment if you disagree

Appendix E

Messaging System Questions: Requirements Model

Security Policy Definition

System Description

For the next set of questions, the system is defined as an enterprise mail infrastructure. It supports a global Fortune 500 company with over 15,000 mail clients deployed. This infrastructure is used to support both internal and external electronic mail, and is directly connected to the Internet. For this discussion, we are only concerned about the access control policy. The system in question is implementing a new type of access control, risk adaptive access control, which considers the context of the system environment to make access control decisions.

13. To send mail, a user shall be a registered user in the enterprise.
A registered user shall be defined as a user with a unique identity.
An registered user shall authenticate to the system with an authentication token.
A mailbox shall be created for all registered users of the messaging system.

Do you agree or disagree that this is a minimum sufficient set of access controls? If you disagree, please comment

- ○ Agree
- ○ Disagree

If you disagree, please comment

14. The system shall allow for a maximum mailbox size of 200MB.
A warning message shall be generated if the mailbox size exceeds 150 MB.
When the maximum mailbox size is reached, a user shall not be allowed to send any messages until the mailbox size is reduced.
Approval from the Messaging System Administrator shall be required to expand the maximum mailbox size beyond 200 MB.

Are these valid requirements for a messaging system?

- ○ Yes
- ○ No

If no, please comment

Security Policy Definition

15. A user shall not send mail on the behalf of another user.
A user can function in an administrative role and send mail on the behalf of a user provided the administrative role is bound to both messaging system users.
Messages sent on the behalf of a user shall also be sent to the user normally in this role, as well as the intended recipient.

An admin is asked to send a message for the department manager.

How many messages are actually sent?

- ○ None
- ○ One
- ○ Two
- ○ Three

Other (please specify)

16. The system shall deliver a message within 15 minutes of the user hitting "send."

If the system cannot deliver the message, a notice of non-delivery shall be sent to the sender, with an explanation for why the message was not delivered.

A user has exceed their mailbox size, but does not know it and hits "send". What will the result be?

- ○ The message will be sent and delivered to the recipient
- ○ A notice of non-delivery will be sent to the sender
- ○ The sender has 15 minutes to clean up his mailbox; then the message will be sent.
- ○ Unable to determine based on provided information.

Comment if desired

Security Policy Definition

17. A user leaves the organization. Can the user still send mail?

- ○ Yes
- ○ No
- ○ Maybe

Comment if desired

Appendix F

Messaging System Narrative Format

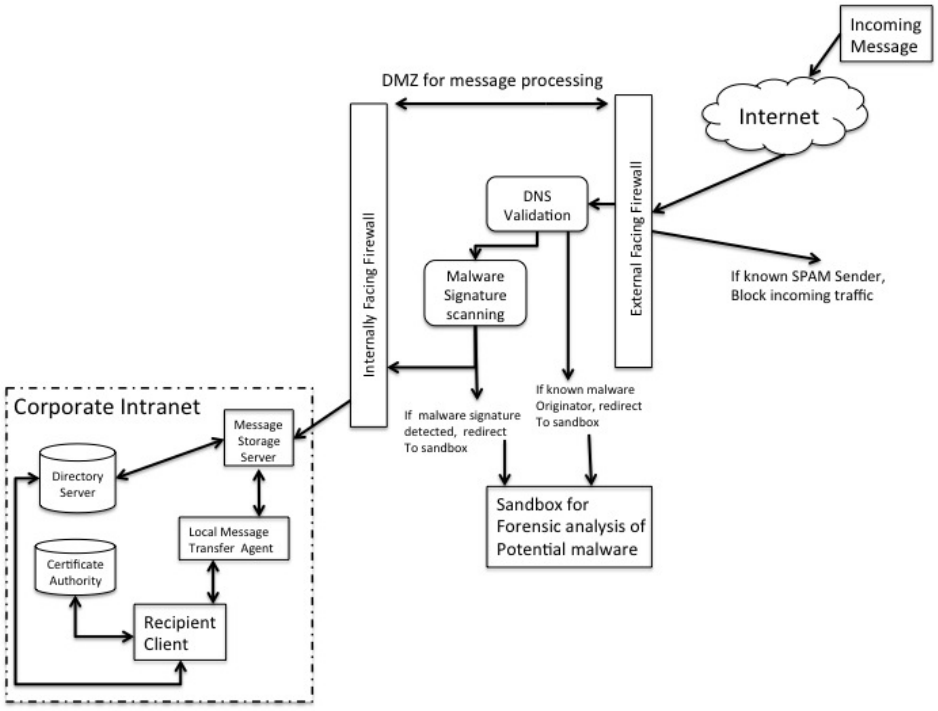
Security Policy Definition

System Description

For the next set of questions, the system is defined as an enterprise mail infrastructure. It supports a global Fortune 500 company with over 15,000 mail clients deployed. This infrastructure is used to support both internal and external electronic mail, and is directly connected to the Internet. For this discussion, we are only concerned about the access control policy. The system in question is implementing a new type of access control, risk adaptive access control, which considers the context of the system environment to make access control decisions.

Figure 1 is the basic architecture of the Enterprise Email system. For the purposes of this research, please refer to this figure for the next set of questions.

To send mail, a user must be registered in the enterprise, and have a unique user identity and authentication token as well as a mailbox. This is the non-discretionary access control policy of the messaging system.



Security Policy Definition

13. Do you agree this is a minimally sufficient set of access controls? If you disagree, please comment.

- ○ Agree
- ○ Disagree

if you disagree, please comment

Security Policy Definition

Messaging Narrative

When a user joins the company, he is issued a unique user account and an authentication token so he can access the company systems. A mailbox is also created for him. When a user leaves the company, the user account is disabled, but the mailbox remains active. A maximum mailbox size is set, and if a user exceeds it, they cannot send or receive mail. Administrative Assistants can send mail on the behalf of their managers if they have been granted that privilege. A department manager travels a lot and has his administrative assistant read and respond to his routine email so he doesn't have to worry about them while he is on the road. The administrator always sends a copy of the message to the boss so he knows what the message response was. The administrative assistant cannot delete messages from the manager's account.

14. Do you agree that this is a sufficient set of access controls for email? If you disagree, please comment.

- ○ Agree
- ○ Disagree

Please comment if you disagree

15. The department manager exceeds the maximum mailbox size. What happens?

- ○ He can only send email.
- ○ He can only receive email.
- ○ All incoming messages are rejected.
- ○ All outgoing messages are rejected.
- ○ No messages can be sent or received.

If no, please comment

Security Policy Definition

16.

An administrative assistant is asked to send a message for the department manager. How many messages are actually sent?

- None
- One
- Two
- Three

Other (please specify)

17.

A user has exceed their mailbox size, but does not know it and hits "send". What will the result be?

- The message will be sent and delivered to the recipient
- A notice of non-delivery will be sent to the sender
- The sender has 15 minutes to clean up his mailbox; then the message will be sent.
- Unable to determine based on provided information.

Comment if desired

18. A user leaves the organization. Can the user still send mail?

- Yes
- No
- Maybe

Comment if desired

Appendix G

Network Infrastructure, Requirements Format

Security Policy Definition

System Description

For the next section, the system in question is a network infrastructure. The network infrastructure is responsible for connecting over 15,000 individual sites for a mission critical systems control application. The network terminates at a site service delivery point, which is similar to the cable service provider termination at a personal residence.

Network device health and status information shall be segregated from the user data within the network.
The network and security management node shall enforce the concept of least privilege.
Two man control (security and network administrators) shall be required to add a component to the infrastructure: no one person can completely register a device on the network and configure it.

13. Joe, a network administrator, and Jeff, a security administrator, were supposed to work the third shift. Jeff called in sick for the night.

Can Joe add a device to the infrastructure and place it in operation?

- ○ Yes
- ○ No

Other (please elaborate)

14. Establishing network routing tables is a function of which role?

- ○ Security Administrator
- ○ Network Administrator

Other (please elaborate)

15. Monitoring of audit logs is a function of which role?

- ○ Security Administrator
- ○ Network Administrator

Please elaborate

Security Policy Definition

16. Can a security administrator view a network administrator's status information?

- ○ Yes
- ○ Yes, but he can't change it
- ○ No

Other (please specify)

17. User data can only enter the network through a Service Delivery Point

- ○ True
- ○ False

Comment if desired

18. Vulnerability remediation(patch management) fixes vulnerabilities in the network infrastructure. This is a function that is best handled by

- ○ The security administrator
- ○ The network administrator

Comment if desired

Appendix H

Network Infrastructure, Narrative Format

Security Policy Definition

System Description

For the next section, the system in question is a network infrastructure. The network infrastructure is responsible for connecting over 15,000 individual sites for a mission critical systems control application. In this architecture, the network device health and status information is segregated from the device data streams. That is, information on the health of the network and its connections is transmitted to a management node, while information used by the network applications is transmitted to analytical nodes for processing and visualization. The management node enforces least privilege and two man controls on the infrastructure components: no one person can completely register a device on the network and configure it. The networks terminate at service delivery points (SDPs) that are not necessarily end user servers or workstations.

13. Joe, a network administrator, and Jeff, a security administrator, were supposed to work the third shift. Jeff called in sick for the night.

Can Joe add a device to the infrastructure and place it in operation?

- ○ Yes
- ○ No

Other (please elaborate)

14. Establishing network routing tables is a function of which role?

- ○ Security Administrator
- ○ Network Administrator

Other (please elaborate)

15. Monitoring of audit logs is a function of which role?

- ○ Security Administrator
- ○ Network Administrator

Please elaborate

Security Policy Definition

16. Can a security administrator view a network administrator's status information?

- ○ Yes
- ○ Yes, but he can't change it
- ○ No

Other (please specify)

17. User data can only enter the network through a Service Delivery Point

- ○ True
- ○ False

Comment if desired

18. Vulnerability remediation(patch management) fixes vulnerabilities in the network infrastructure. This is a function that is best handled by

- ○ The security administrator
- ○ The network administrator

Comment if desired

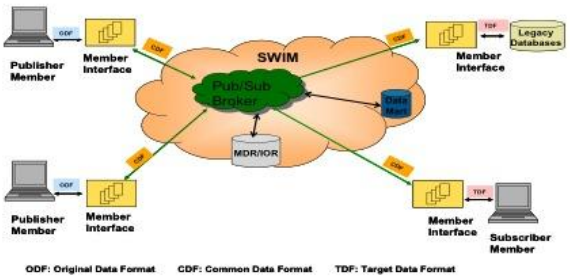
Appendix I

Publish Subscribe Model, Requirements Format

Security Policy Definition

System Description

For the next set of questions, the system shall support a content creator/content subscriber model. All content shall be created within an enterprise and shall be stored in a data repository. When new content is available, subscribers shall be notified. The information broker validates information flow as illustrated below.



Security Policy Definition

13. Given the available information, how many types of subscribers exist in this system?

- ○ None
- ○ One
- ○ Two
- ○ Unknown (please comment)

Please elaborate on unknown responses

14. An external subscriber's network connection is terminated in the middle of a file transfer. How would the subscriber tell the originator to retransmit?

- ○ The subscriber doesn't have to, the network takes care of it.
- ○ Send an email or phone the publisher and ask for another.
- ○ Go to the outward facing website and download the file again.
- ○ None of the above

If none, please comment

15. An internal subscriber is creating a new report and accesses content that can be released to external subscribers. Is the internal subscriber allowed to modify the content?

- ○ Yes
- ○ No
- ○ Unknown

if Unknown, please elaborate

Security Policy Definition

16. An external provider "overlays" additional information on top of content retrieved from the externally facing website and uses it to generate a new product. Is this allowed?

- ○ Yes
- ○ No
- ○ Other

If Other, please elaborate

17. An internal user wants to communicate with a specific external subscriber. Can the system as described keep other external subscribers from viewing the communication?

- ○ Yes
- ○ No
- ○ Unknown

If Unknown, please elaborate

Appendix J

Publish Subscribe Model, Narrative Format

Security Policy Definition

System Description

For the next set of questions, the system is defined as a content creator/content subscriber model. All content is created within an enterprise and stored in a data repository. When new content is available, subscribers are notified. The information broker validates that the subscriber is within the enterprise, or an external subscriber.

If the subscriber is an external subscriber, the information is pushed through a firewall and made available on an outward facing message bus. No information is passed back through the firewall to the internal message bus. An external subscriber can collect information from the external message bus only, and cannot alter the original content on the internal message bus.

13. Given the available information, how many types of subscribers exist in this system?

- ○ None
- ○ One
- ○ Two
- ○ Unknown (please comment)

Please elaborate on unknown responses

14. An external subscriber's network connection is terminated in the middle of a file transfer. How would the subscriber tell the originator to retransmit?

- ○ The subscriber doesn't have to, the network takes care of it.
- ○ Send an email or phone the publisher and ask for another.
- ○ Go to the outward facing website and download the file again.
- ○ None of the above

If none, please comment

15. An internal subscriber is creating a new report and accesses content that can be released to external subscribers. Is the internal subscriber allowed to modify the content?

- ○ Yes
- ○ No
- ○ Unknown

if Unknown, please elaborate

Security Policy Definition

16. An external provider "overlays" additional information on top of content retrieved from the externally facing website and uses it to generate a new product. Is this allowed?

- ○ Yes
- ○ No
- ○ Other

If Other, please elaborate

17. An internal user wants to communicate with a specific external subscriber. Can the system as described keep other external subscribers from viewing the communication?

- ○ Yes
- ○ No
- ○ Unknown

If Unknown, please elaborate

APPENDIX K

Detailed Question Analysis

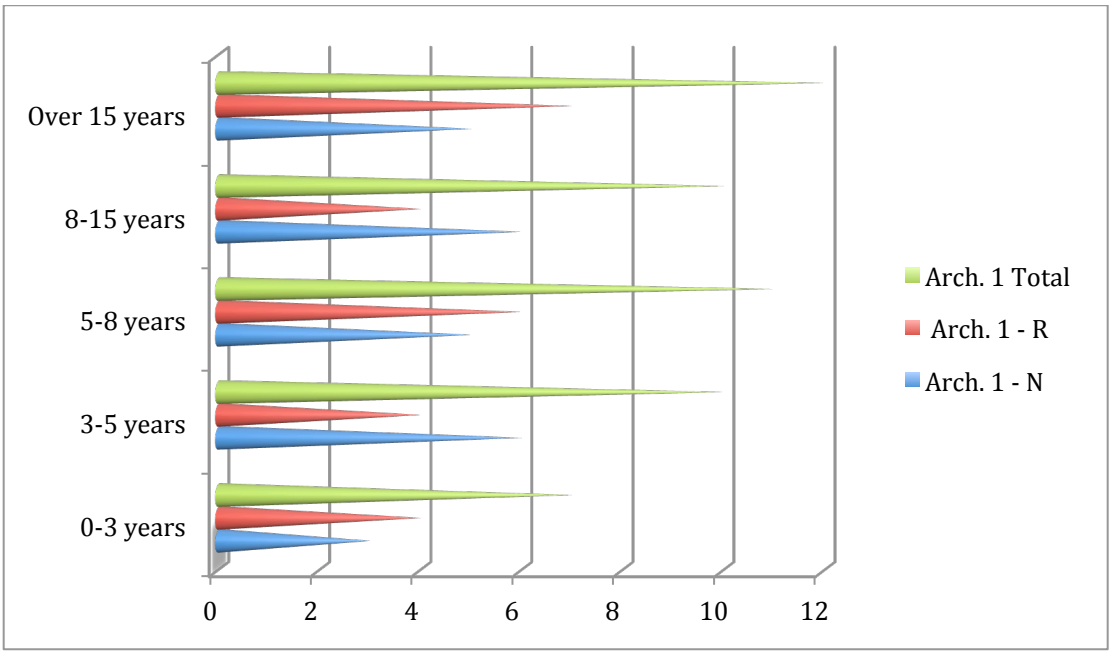


Figure K-1a. Demographic Information, Question 2, Architecture 1.

Total Sample Size = 50 subjects. 25 for Requirements Format, 25 for Narrative Format.

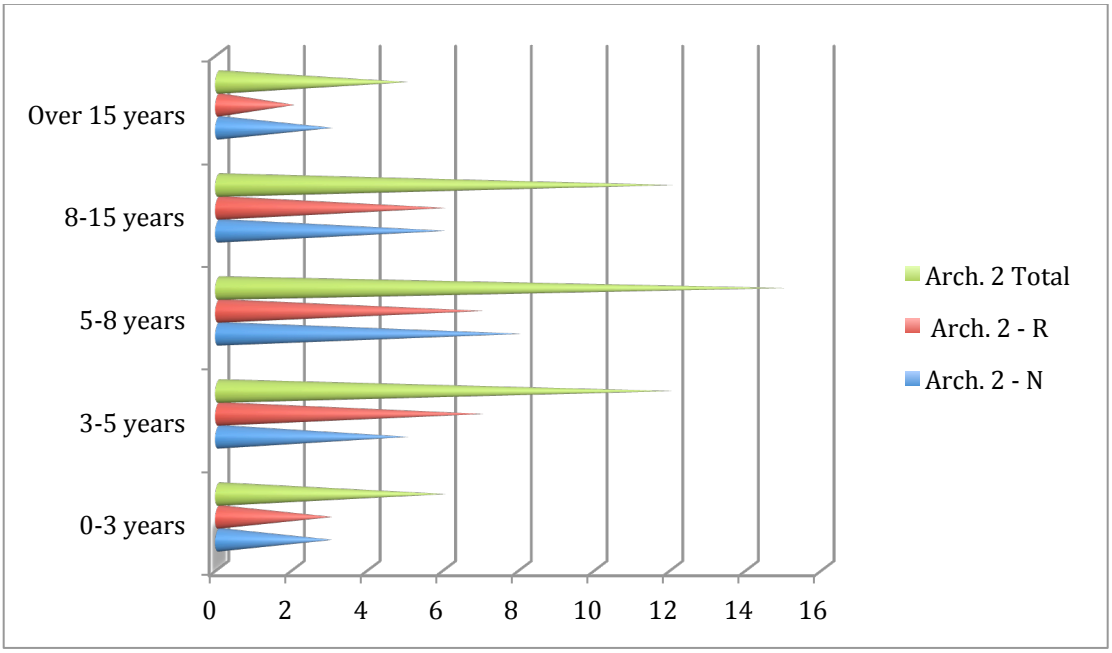


Figure K-1b. Demographic Information, Question 2, Architecture 2.

Total Sample Size = 50 subjects. 25 for Requirements Format, 25 for Narrative Format.

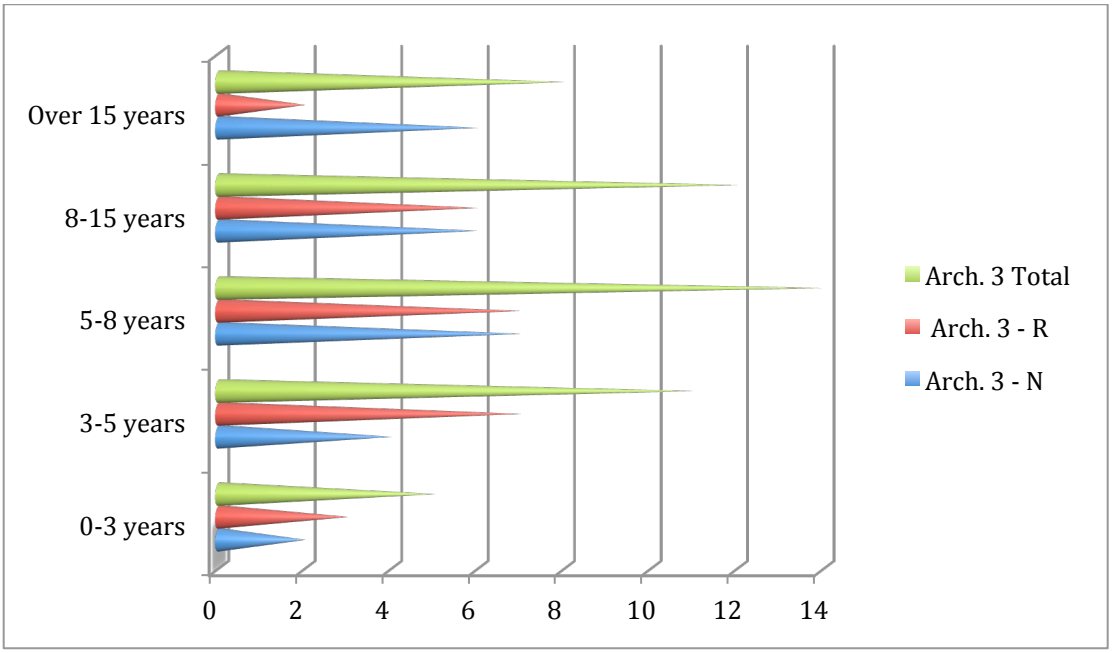


Figure K-1c. Demographic Information, Question 2, Architecture 3

Total Sample Size = 50 subjects. 25 for Requirements Format, 25 for Narrative Format.

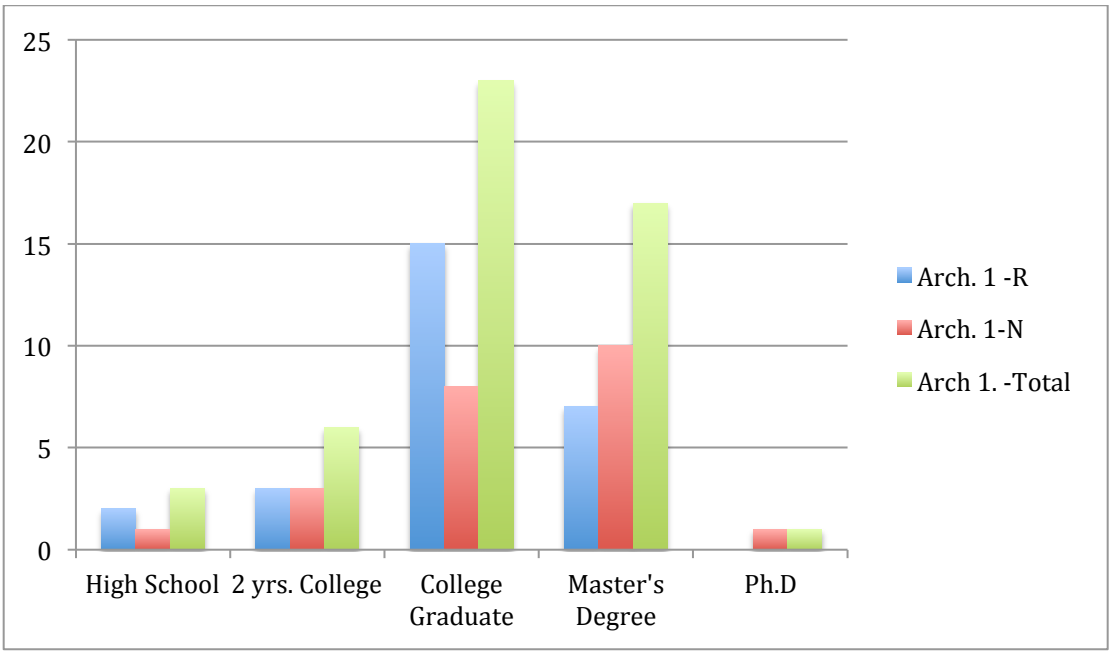


Figure K-2a. Demographic Information, Question3, Architecture 1

Total Sample Size = 50 subjects. 25 for Requirements Format, 25 for Narrative Format.

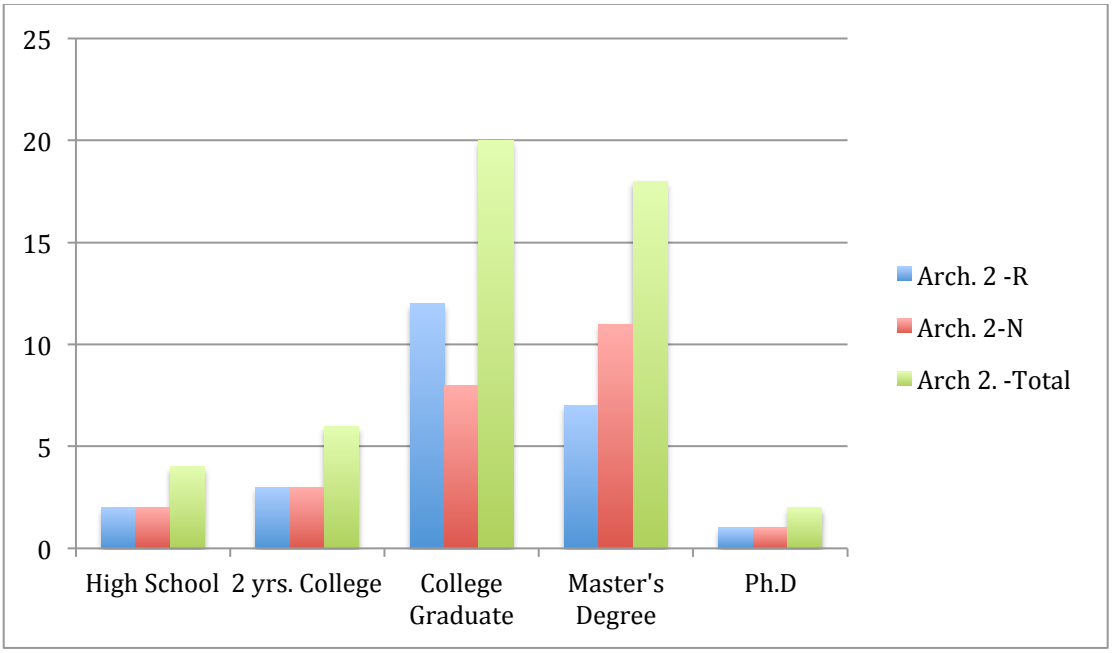


Figure K-2b. Demographic Information, Question3, Architecture 2

Total Sample Size = 50 subjects. 25 for Requirements Format, 25 for Narrative Format.

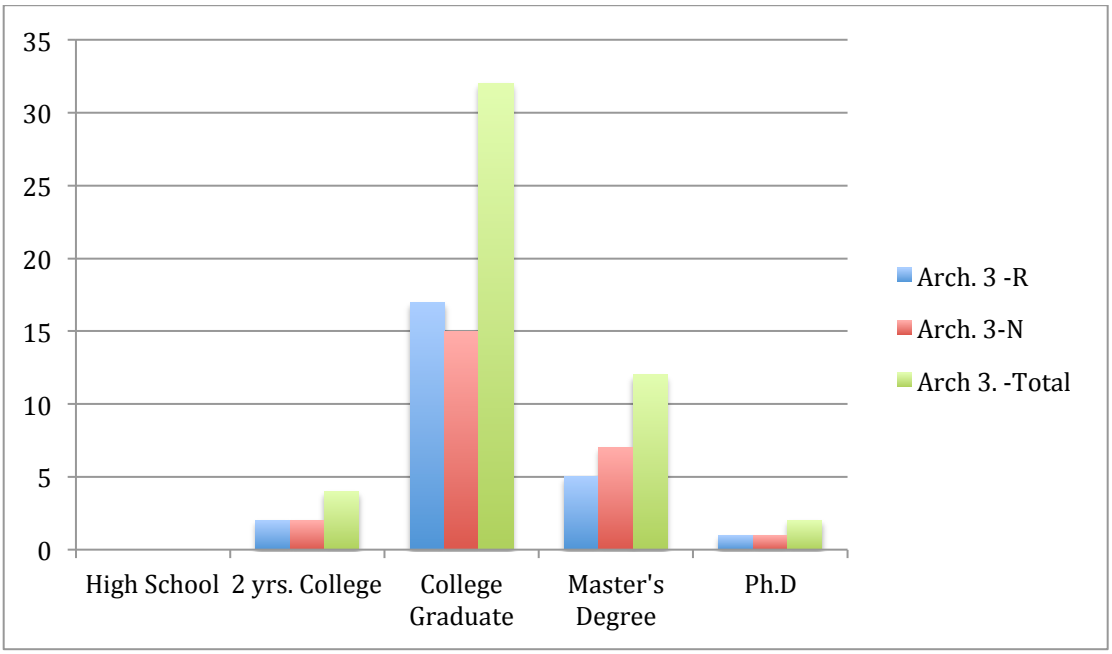


Figure K-2c. Demographic Information, Question3, Architecture 3

Total Sample Size = 50 subjects. 25 for Requirements Format, 25 for Narrative Format

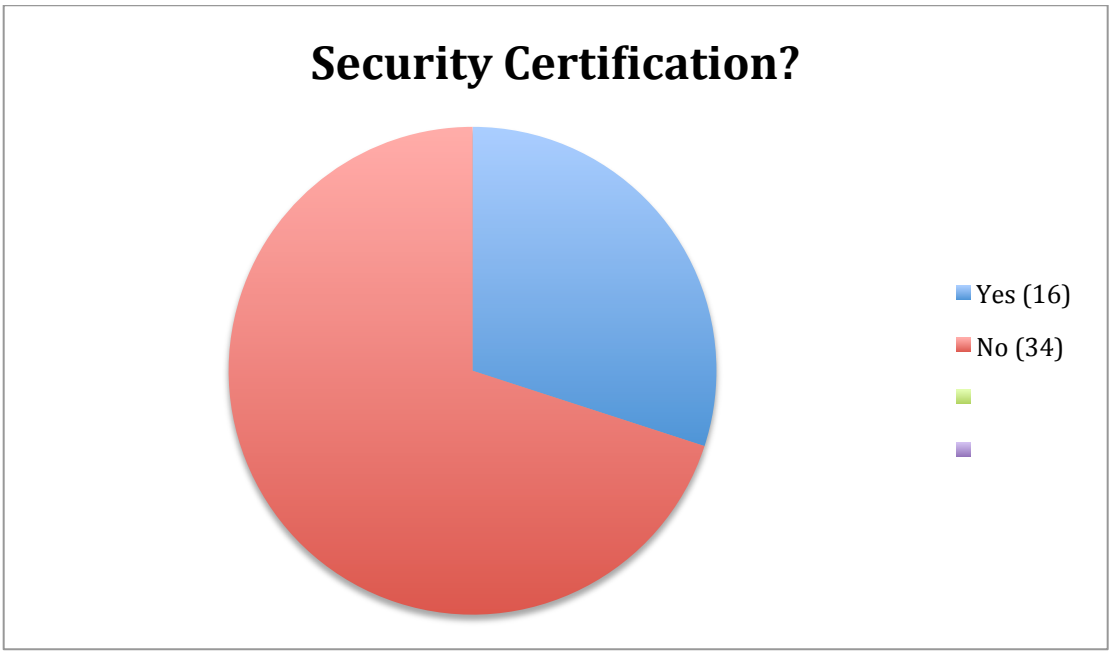


Figure K-4a: Respondents with Security Certifications, Architecture 1 (Total 50 Subjects)

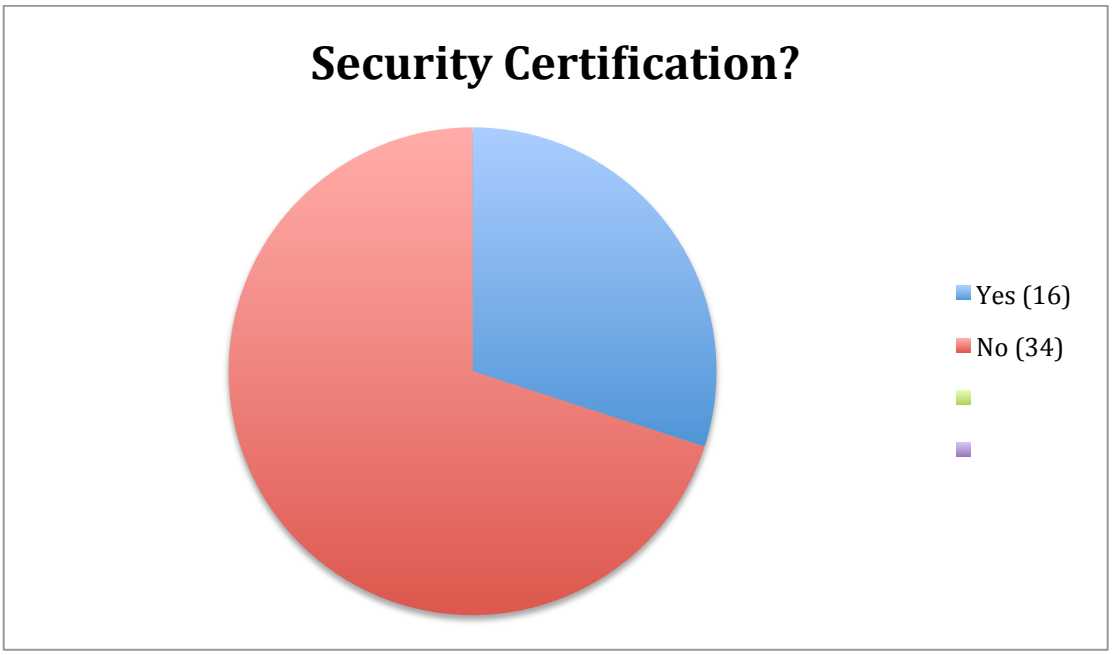


Figure K-4b: Respondents with Security Certifications, Architecture 2 (Total 50 Subjects)

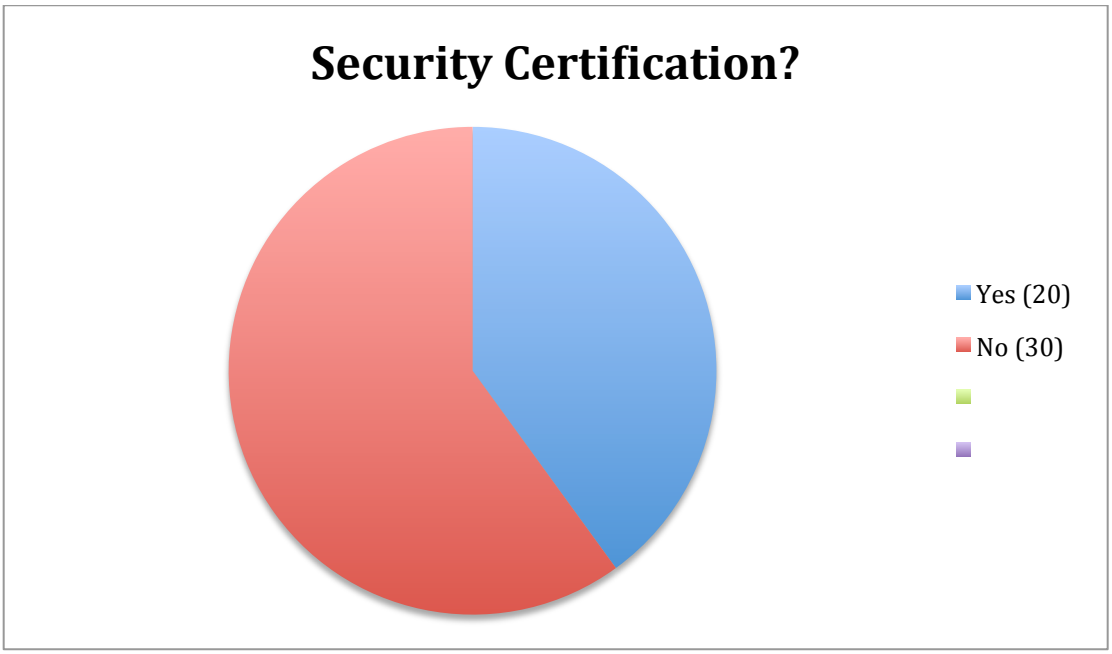


Figure K-4c. Respondents with Security Certifications, Architecture 3 (Total 50 Subjects)

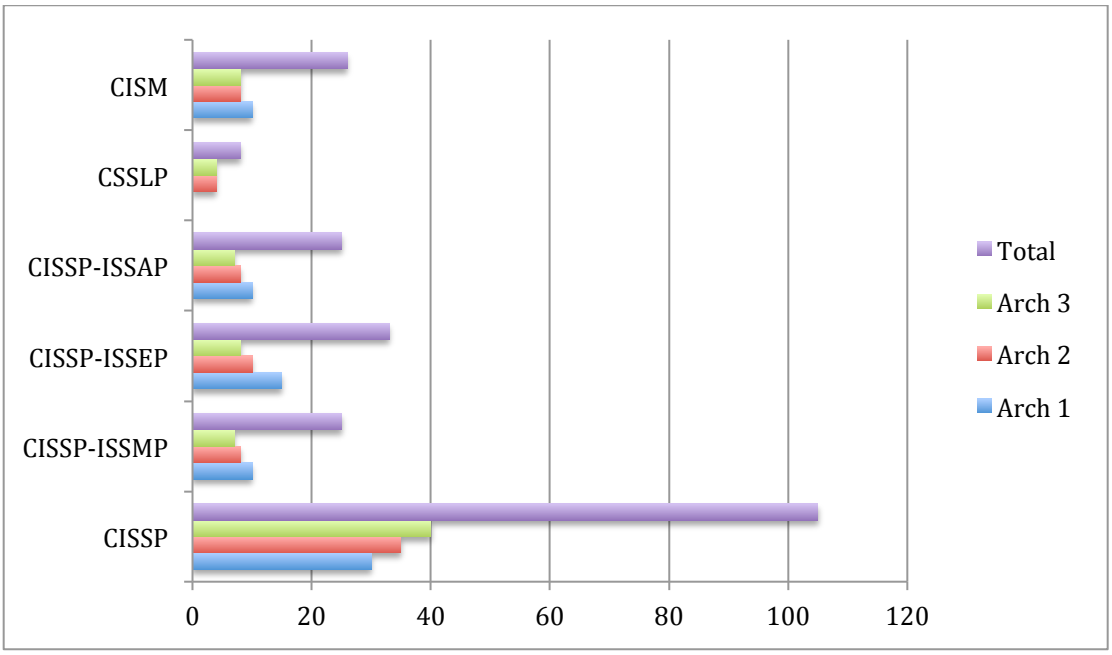


Figure K-5. Types of Certifications Held (all architectures).

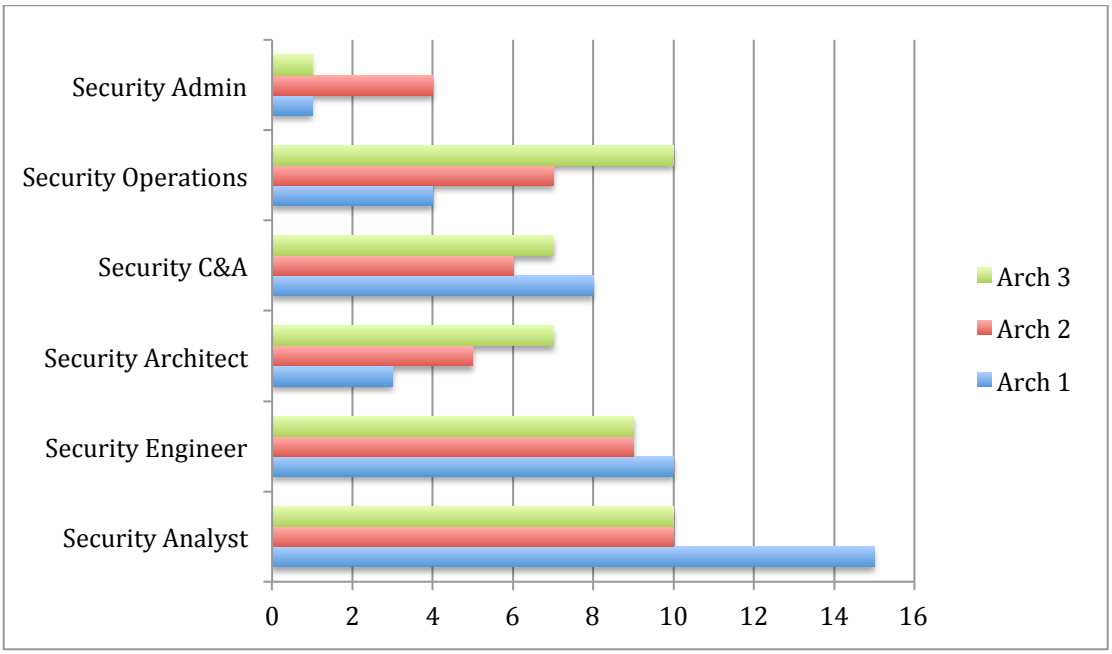


Figure K-6. Job Function of Respondents (self-categorized)

150 respondents, 50 each in architecture 1, architecture 2, and architecture 3.

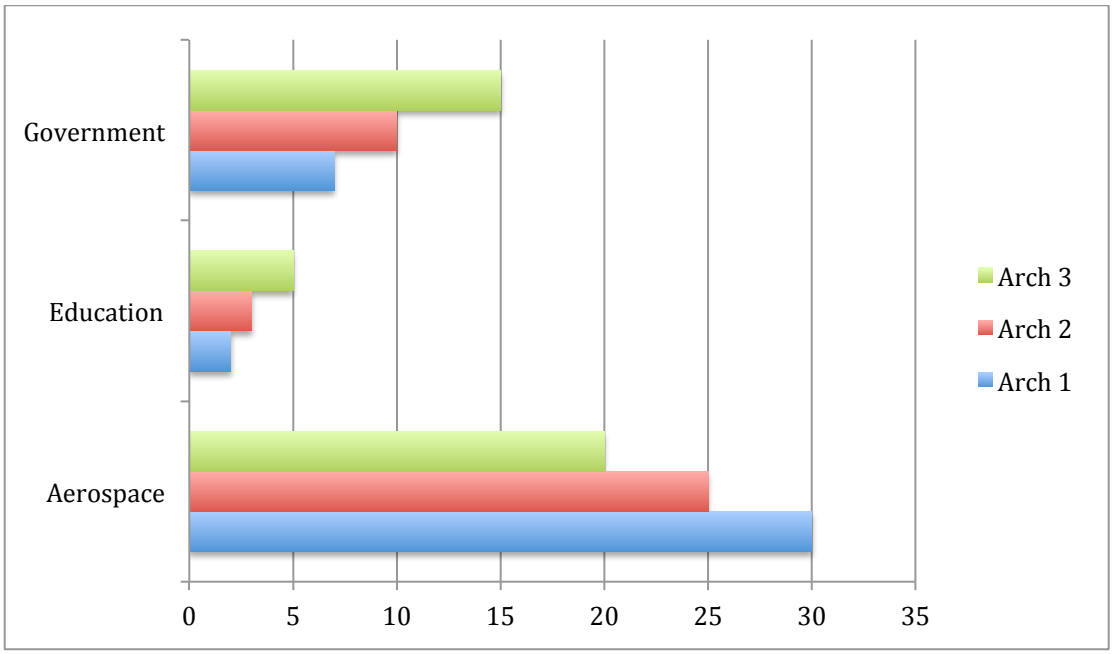


Figure K-7. Principal Industries of Responders.

References

Alexander, P. (2006). System specification using rosetta. In R. Vauaghn, & M. Warkentin (Ed.), *Enterprise Information Systems Assurance and System Security: Managerial and Technical Issues*. Starkville, MS, US: Idea Group.

Ardanga, C. A. et al. (2006). Supporting location-based conditions in access control policies. *ASIACCS '06*. Taipei, Taiwan: ACM Press.

Asgharpour, F. L. (2007). Mental Models of Compuer Security Risks. *Workshop on the Economics of Information Security (WEIS)*. Washington, DC: IEEE Press.

Atallah, M., & McDonough, C. R. (2000). Natural Language Processing for Information Assurance and Security: An Overview and Implementation. *New Security Paradigms Workshop* (pp. 51-65). New York, NY: ACM Press.

Ayuso, D., Varda, S., & Weischedel, R. (1987). An environment for acquiring semantic information. *Association for Computational Linguistics Annual Meeting* (pp. 32-40). Association for Computational Linguistics.

Baskerville, R. & Siponen, Milo. An information security meta-policy for emergent organizations. *Logistics Information Management , 15* (5/6), 337-346.

Baskerville, R. (1993). Information systems security design methods: Implications for information systems development. *Computing Surveys , 25* (4), 375-414. ACM Press.

Beatie, B. A. (1976). "Romances traditionales" and spanish traditional ballads: Menedez Pidal v. Vladimir Popp. *Journal Folklore Institute , 13* (1), pp. 37-55. Indianapolis, IN, University of Indiana.

Bell, D. & LaPadula, L. (1976). *Secure computer systems: Unified exposition and Multics interpretation*. US Air Force. Boston, MA: U.S. Government.

Bell, D. E. (2005). Looking back at the Bell-La Padula model. *Computer Security Applications Conference*. Tuscon, AZ: Applied Computer Security Associates.

Bell, D. (1994). Modeling the "multipolicy machine". *New Security Paradigms Workshop*. Little Compton, RI: ACM Press.

Bertino, E. B., Ferrari, E., & Perlasca, P. (2001). A logical framework for reasoning about access control models. *ACM SACMAT'01*. Washington: ACM Press.

Bertino, E., Catania, B., Damiani, M., & & Perlasca, P. (2005). GEO-RBAC: a spatially aware RBAC. *SACMAT'05*. New York, NY: ACM Press.

Bloomberg, L. D., & Volpe, M. (2012). *Completing Your Qualitative Dissertation: A Road Map from Beginning to End* (Second Ed. ed.). Los Angeles, CA, U.S.: Sage Publications.

Boyd, B. (2009). *On the Origin of Stories: Evolution, Cognition, and Fiction*. Boston, MA, US: Harvard University Press.

Bransford, J. & Brown, A. (2000). *How People Learn*. Washington, DC, US: National Academy Press.

Brooks, K. M. (1996). Do Story Agents Use Rocking Charis? The Theory and Implementation of One Model for Computational Narrative. *Fourth ACM Conference on Multimedia*. New York: ACM Press.

Burnside, M. & Keromytis, A.D. (2007). Arachne: Integrated Enterprise Security Management. *Information Assurance Workshop*. Washington, DC: IEEE Press.

Carnielli, E. & Pittarello, F. (2009). Interactive stories on the net: a model and an architecture for X3D worlds. *14th International Conference on Web 3D Technology* (pp. 91-99). New York, NY: ACM Press.

Cao, X., Iverson, J. (2006) Intentional access management: Making access control usaable for end-users. *Proceedings of the Second Symposium on Usable Privacy and Security (SOUPS 2006)*, (pp. 20-31) New York, NY: ACM Press.

Cavazza M. and Pizzi, D. (2006). Narratology for Interactive Storytelling: A Critical Introduction. In S. Gobel, & a. I. Malkewitz, *Technologies for Interactive Storytelling and Digital Entertainment* (pp. 72-83). Berlin, DDR: Springer.

Chakraborty, S. & Dehlinger, J. (2009). Applying the Grounded Theory Method to Derive Enterprise System Requirements. *10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing* (pp. 333-338). Washington, DC: IEEE Press.

Chakraborty, S. & Ray, I. (2006). TrustBAC -- Integrating trust relationships into the RBAC model for access control in open systems. *SACMAT '06* (pp. 49-58). New York,NY: ACM Press.

Chatham, S. (1978). *Story and Discourse: Narrative Structure in Fiction and Film*. Ithaca, NY, US: Cornell University Press.

Choudhary. (2005). A policy based architecture for NSA RAdAC Model. *Sixth IEEE SMC Information Assurance Workshop* (pp. 294-301). Washington, DC: IEEE Press.

Constantine, L. & Lockwood, A.D. (1999). *Sofware for Use: A Practical Guide to Models and Methods of Usage*. Boston, MA, USA: ACM Press.

Covington, M. W., Srinivasan, S., Dey, A. K., Ahamad, M., & Abowd, G. D. (2001). Securing context-aware applications using environment roles. *SACMAT'01*. Washington: ACM.

Cranor, L.F., Guduru, P., & Arjula, M. (2006) User Interfaces for Privacy Agents. *ACM Transactions on Computer-Human Interaction* 13(2*)*, 135-178. New York, NY, ACM Press.

Creswell, J. W. (2013). *Qualitative Inquiry and Research Design: Choosing Among Five Approaches* (Third Ed.). Los Angeles, CA, U.S.: Sage Publications.

Edwards, W. K. (2005, December). Putting Computing in Context: An Infrastructure to Support Extensible Context-Enhanced Collaborative Applications. *Transactions on Human-Computer Interaction, 12* (4), pp. 446-474.

Edwards, W. K. (1997). Representing activity in collaborative systems. *6th IFIP Conference on Human Computer Interaction*. Sydney, AUS.

Erickson, T. (1996, July/August). Design as storytelling. *Interactions , 3* (4), pp. 31-35.

Ferraiolo, D. a. (1995). Role-based access controls. *Fifteenth National Computer Security Conference*. Baltimore, MD: U.S. Government.

Glaser, B. G. (1967). *The Discovery of Grounded Theory: Strategies of Qualitative Research*. Chicago, IL, USA: Aldine Publishing Company.

Gligor, V. (1995). Characteristics of role-based access control. *First ACM Workshop on Role-based access control*. Washington, DC: ACM Press.

Good, N.S., and Krekelberg, A. Usability and Privacy: a Study of Kazaa P2P File-sharing, *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2003)*, New York, NY, April 2003, (pp. 137-144) ACM Press, New York

Goodrich, P. (2007). Narrative as Argument. In D. Herman, M. Jahn, & M.-L. Ryan, *Routledge Encyclopedia of Narrative Theory* (pp. 348-349). London, UK: Routledge.

Gottschall, J. (2012). *The Storytelling Animal: How Stories Make Us Human*. New York, NY, U.S.: Houghton Mifflin Harcourt Publishing.

Government, U.S. (2004). *Global Information Grid Information Assurance Reference Capabilities Document*. U.S. Government, Department of Defense. Washington, DC: U.S. Government.

Government, U.S. (2011). *Special Publication 800-53 rev. 4, .* National Institute Of Standards and Technlogy, Commerce. Washington, DC: U.S. Government.

Hafmann, U., & Kuhnhauser, W. (1999). Embedding security policies into a distributed computing environment. *SIGOPS Operating System Review , 33* (2), pp. 51-64.

Han, Y. L. (2000). An object-oriented model of access control based on role. *SIGSOFT Software Engineering Notes , 25* (2), pp. 64-68.

Haven, K. (2007). *Story proof, the science behind the startling power of story*. Westport, CT, US: Libraries Unlimited.

Hengartner, U. & Steenkiste, P. (2004). Implementing access control to people location information. *SACMAT'04*. New York, NY: ACM Press.

Hoagland J.A., Pandey, R. & Levitt, K.N. (1998). *Security policy specification using a graphical approach*. University of California-Davis, Computer Science. Davis, CA: University of California.

Hosmer, H. (1991). Metapolicies I. *Special Workshop on Data Management Security and Privacy Standards*. San Antonio: ACM SIGSAC.

Hosmer, H. (1993). The multipolicy paradigm for trusted systems. *New Security Paradigms Workshop*. New York, NY: ACM Press.

Hulsebosch, R., Salden, A., Bargh, M., & Ebben, P.W.G. (2005). Context Sensitive Access Control. *SACMAT '05*. Stockholm, Sweden: ACM Press.

Jaeger, T. & Tidswell, J. (2001). Practical safety in flexible access control models. *Transactions on Information and System Security, 4* (2), 158-190. New York, NY: ACM Press.

Jajodia, S., Samarati, P., & Subrahmanian, V.S. (1997). A logical language for expressing authorizations. *IEEE Symposium on Security and Privacy* (pp. 31-42). Washington, DC: IEEE.

Johnson, M., Karat, J., Karat, M.-C., & and Gruenberg, K. (2010). Optimizing a Policy Authoring Framework for Security and Privacy Policies. *Symposium on Usable Privacy and Security (SOUPS)*. New York, NY: ACM Press.

Johnson, M., Karat, J., Karat, M.-C., & and Gruenberg, K. (2010). Usable Policy Template Authoring for Iterative Policy Refinement. *Policies for Distributed Systems and Networks* (pp. 18-24). Washington, DC: IEEE.

Karat, J. Karat, C.-M, Brodie, C., Feng, J. (2005) Privacy in information technology: Designing to enable privacy policy management in organizations. *International Journal of Human-Computer Studies* 63(1-2), 153-174.

Karp, A., Haury, H. & Davis, M.H. (2009). *From ABAC to ZBAC: the Evolution of Access Control Models*. Hewlett-Packard Laboratories. Cupertino, CA: Hewlett-Packard.

Klunder, D. (n.d.). *Electronic Tolling*. (A. C. Union, Producer) Retrieved August 15, 2009, from www.acluwa.org: http://www.acluwa.org/library_files/TollingPositionPaper.pdf

Kuhn, D. R. (2010, June). Adding attributes to role-based access control. *Computer* , pp. 79-81. IEEE.

Lehmann, H. (2010). Grounded Theory and Information Systems: Are We Missing the Point? *43rd Hawaii International Conference on System Sciences* (pp. 1-11). Washington, D.C.: IEEE Computer Society Press.

Lei, H., Daby, M., Davis, J., Banavar, G., & Ebling, M. (n.d.). The design and application of a context service. *Mobile Computing and Communications Review , 6* (4), pp. 45-55.

LeMay, M., & Fatemieh, O. & Gunter, C.A. (2007). PolicyMorph: Interactive Policy Transformations for a Logical Attribute-Based Access Control Framework. *SACMAT '07* (pp. 205-214). New York, NY: ACM Press.

Lin, T. (2000). Chinese wall security model and conflict analysis. *24th International Computer Software and Applications Conference* . Washington, DC: ACM Press.

Lueger, G. *Artificial Intelligence, structures and strategies for complex problem solving (Fourth ed.)*. Essex, England: Pearson Education Ltd.

MacGraw, R. (2009). Risk Adaptive Access Control. *NIST Workshop on Privilege Management*. Washington, DC: U.S. Government.

Matthew Burnside, A. D. (2007). Arachne: Integrated Enterprise Security Management. *Workshop on Information Assurance*. West Point, NY: IEEE.

Maxion, R. A. & Reeder, R. W. (2005) Improving User-Interface Dependability through Mitigation of Human Error. *International Journal of Human-Computer Studies* 63(1-2), 25-50.

Mazalek, A. & Davenport, G. (2003). A Tangible Platform for Documenting Experiences and Sharing Multimedia Stories. *ACM SIGMOD Workshop on ExperientalTelepresence ETP '03*. NEW York: ACM Press.

Mazalek, A. et. al. (2002). Tangible Viewpoints: A Physical Approach to Multimedia Stories. *Conference on Multimedia Systems*. New York, NY: ACM Press.

McGraw, R. (2004). Securing Content in the Department of Defense's global information grid. *DoD Secure Knowledge Management Workshop*. Buffalo, NY: U.S. Government.

Meadows, D. H. (2008). *Thinking in Systems: A Primer*. (D. Wright, Ed.) White River Junction, VT, US: Chelsea Green Publishing Co.

Miles, M. &. (1994). *Qualitative Data Analysis: An Expanded Sourcebook, 2nd Ed*. Newbury Park, CA, USA: Sage Publications.

National Institute of Standards and Technology . (2004). *Minimum Essential Requirements for Federal Information and Federal Information Processing Systems*. National Institute of Standards and Technology, Commerce. Washington, DC: U.S. Government.

Nissan, E. (2008). Nested Beliefs, Goals, Duties, and Agents Reasoning about their own or Each Other's Body in the TIMUR Model: A formalism for the Narrative of Tamerlane and the Three Painters. *Journal of Intelligent Robotic Systems , 52*, 515-582.

NIST. (2010). *A Report on the Privilege (access) Management Workshop*. National Institute of Standards and Technology, Commerce. Washington, DC: U.S. Government.

NIST. *Information Assurance Glossary*. Committee on National System Security Information, National Institute of Standards and Technology. Washington, DC: U.S. Government.

OpenGIS Consortium. (1999). *Open GIS simple features specification for SQL*. Technical, OpenGIS Consortium.

Owen, T., Wakeman, I., Keller, W., Weeds, J., & ≈Weir, D. (2005). Managing the Policies of non-technical users in a dynamic world. *Sixth International conference on Policies for Distributed Systems and Networks*. Washington, DC: IEEEE.

Oxford English Dictionary. (n.d.). *Definition Lookup*. (o. e. dictionary, Producer, & Oxford University Press) Retrieved April 7, 2012, from www.oxforddictionaries.com: http://oxforddictionaries.com/definition/security?region=us&q=security

Pan, C., Mitra, P., & Liu, P. (2006). Semantic access control for information interoperation. *SACMAT '06*. New York, NY: ACM Press.

Park, J. C., Neven, T., & Diosomito, J. (2004). A composite RBAC approach for large, complex organizations. *SACMAT'04*. Yorktown Heights, NY: ACM.

Pereira, C., & and Sousa, P. (2004). A method to define an enterprise architecture using the Zachman Framework. *ACM Symposium on Applied Computing* (pp. 1366-1371). New York, NY: ACM Press.

Popp, V. (1968). *Morphology of the Folktale*. (,. L. Wagner, Ed.) Austin, TX, US: University of Texas Press.

Raskin, V., Hempelmann, C. F. Nierenberg, S. & Trienzenberg, K. (2001). Ontology in information security: a useful theoretical foundation and a methodological tool. *New Security Paradigms Workshop*. New York, NY: ACM.

Raskin, V., Hempelmann, C., & Triezenberg, K. (2004). Semantic Forensics: An Application of Ontological Semantics to Information Assurance. *TextMean '04, Proceedings of the Second Workshop on Textual Meaning and Interpretation*. Stroudsburg, PA: Association of Computational Linguistics.

Reeder, R., Karat, C.-M., Karat, J., & Brodie, C. (2007). Usability Challenges in Security and Privacy Policy-Authoring Interfaces. In e. a. C. Baranauskas (Ed.), *INTERACT 2007*. New York, NY: IFIP.

Rees, J. B. (2003). PFIRES: a policy framework for information security. *Communications of the ACM , 46* (7), pp. 101-106.

Ryan, M.-L. (2005). Narrative. In D. Herman, M. Jahn, & Ryan, M.-L. (eds.) *Routledge Encyclopedia of Narrative Theory* (pp. 344-348). London, UK: Routledge.

Sandhu, R. (2004). A logical specification for usage control. *9th Symposium on access control models and technologies*. Yorktown Heights, NY: ACM Press.

Sandhu, R. C. (1996). Role-based access control models. *Computer , 29* (2), pp. 38-47. Washington, DC: IEEE.

Schaefer, R. (2009, Nov). The Epistemology of Computer Security. *SIGSOFT Software Engineering Notes , 34* (6), pp. 1-19.

Schell, R. (1979). Computer Security -- the Achilles' heel of the electronic air force. *Air University Review , XXX* (2), pp. 16-33.

Schell, R. (2001). Information security: science, pseudoscience, and flying pigs. *Annual Computer Security Applications Conference* (pp. 205-216). New York, NY: ACM Press.

Schneider, F. (2000). Enforceable security policies. *Transacations on Information and System Security , 3* (1), pp. 30-50. ACM Press.

Segre, C., & Kemeney, T. (1988). *Introduction to the Analysis of Literary Text*. Bloomington, IN, US: Indiana University Press.

Shank, R. G. (I972, October). Conceptual dependency; a theory of natural language understanding. *Cognitive Psychology, 3* (4), pp. 555-631.

Sherwood, J., Clark, A., & Lynas, .D. (2008). *Enterprise Security Architecture: A Business-Driven Approach*. San Francisco, CA, USA: CMP Books.

Shi, L., & Chadwick, D. (2011). A controlled natural language interface for authoring access control policies. *Symposium on Applied Computing*. New York, NY: ACM Press.

Stoller, P. &. (1987). *In Sorcery's Shadow: A Memoir of Apprenticeship Among the Songhay of Niger*. Chicago, IL, US: University of Chicago Press.

Strauss, A. & Corbin, J. (1990). *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Newbury Park, CA, USA: Sage Publicaitons.

Strembeck, M. & Gustaf, N. (2004). An integrated approach to engineer and enforce context constraints in RBAC envinronments. *Transactions on Information and System Security, 7* (3), pp. 392-429.

Szilas, N. (2004). Minimal Structures for Stories. *First Workshop on Story Representation, Structure, and Context*. New York, NY: ACM Press.

Thomas, R. (1997). Team-based access control (TMAC). *Second ACM Workshop on Rule-based Access Control* (pp. 13-19). New York, NY: ACM Press.

Tolone, W., Ahn, G., Pai, T., & Hong, S.P. (2005). Access Control in collaborative systems. *37* (1), pp. 29-41.

Triantafyllakos, G., Palaigeorgiou, G., & Tsoukalas, I. (2008). Collaborative Design as Narrative. *Indiana Univeristy Conference Proceedings, Participatory Design Conference* (pp. 210-213). Indiana Univeristy/ACM Press.

U.S. Government. (2001). Defense authorization act, government information security reform act (GISRA). *Congressional Record* . Washington, DC, USA.

U.S. Government. (2003). *The 9/11 Commission report, final report of the national commission on terrorist attacks upon the United States*. New York, NY, USA: W.W. Norton & Co., Inc.

Uther, H.-J. (2004). *The types of international folktales: a classification and bibliography, Based on the Sytem of Anitti Aarne and Sith Thompson, Part 3 Appendices*. Helsinki, Finland: Academia Scientiarum Fennica.

Wash, R. & Rader, E. (2011). Influencing Mental Models of Security: A Research Agenda. *New Security Paradigms Workshop*. New York, NY: ACM Press.

Wang, H. S., Livny, M. & McDaniel, P.D. (2004). Security policy reconcililation in distributed computing environments. *Fifth International Workshop on Policies for Distributed Systems and Networks*. Washington, DC: IEEE.

Wang, H., Zang, Y., & and Cao, J. (2006). Ubiquitious computing environments and its usage access control. *INFOSCALE '06*. New York, New York: ACM Press.

Winsborough, W. H. (2004). Security Perspective of Policy. *Fifth International Symposium on Policies for Distributed Systems and Networks*. Washington, DC: IEEEE.

Zalago, N., Barker, A., & Branco, V. (2004). Story Reaction Structures to Emotion Detection. *First Workshop on SRMC*. New York, NY: ACM Press.