

# MODELING SECURITY REQUIREMENTS FOR INFORMATION SYSTEMS DEVELOPMENT

**Murray E. Jennex**  
**San Diego State University**  
mjennex@mail.sdsu.edu

## Abstract

*There are numerous modeling methods for aiding system analysts in identifying information systems (IS) user requirements, including rapid application development, data flow diagrams, entity relation diagrams, and use case diagrams. Additionally, there are several tools that can be used to identify security requirements including checklists, threat and risk analysis, security policies, and security development methods. However, these methods are not integrated into an overall design methodology that can be used to ensure security requirements are identified and then implemented. This paper proposes using barrier analysis and the concept of defense in depth to modify Siponen and Baskerville's [17] integrated design paradigm into a more graphical and easier to understand and use methodology.*

## 1. Introduction

There are numerous modeling methods for aiding system analysts in identifying information systems (IS) user requirements, including rapid application development, data flow diagrams, entity relation diagrams, and use case diagrams. Additionally, there are several tools that can be used to identify security requirements including checklists, threat and risk analysis, security policies, and security development methods. However, these methods are not integrated into an overall design methodology that can be used to ensure security requirements are identified and then implemented. Siponen and Baskerville [17] attempted to resolve this by proposing a security design paradigm that relied on meta-notation to abstract and document integrated security requirements into IS development methods. However, this paradigm has not been widely adopted.

This paper proposes using barrier analysis and the concept of defense in depth to modify Siponen and Baskerville's [17] integrated design paradigm into a more graphical and easier to understand and use methodology. In addition to the meta-notation proposed by [17], this paper proposes the use of barrier diagrams in conjunction with barrier analysis to provide an integrative approach to adding

security into systems design, and to ensure that adequate levels or layers of security are in place at all stages of the software development life cycle. Barrier Analysis is a concept developed by William Haddon, Jr., M.D. in 1973 [9]. Barrier Analysis is most widely known in the Nuclear Energy arena, and has been improved upon by the System Safety Development Center, a training division of the Department of Energy, [4]. Barrier Analysis is a method of identifying hazards or threats, and determining the effectiveness of the preventative/mitigating factors that are constructed to prevent the occurrence of the hazard/threat. Barrier Analysis can also be used after an event has occurred to determine the root cause and to help develop barriers to prevent repeat occurrences [7].

To document the validity and usefulness of the proposed methodology, barrier analysis and defense in depth was tested by a group of graduate students as part of their systems design project. The goal was to determine if the concept of barrier analysis and barrier diagrams could be effectively used in information systems design, and whether or not this methodology is useful in discovering and implementing security requirements. The pilot study was done to determine if further studies and research should be performed to demonstrate that this is a useful methodology that should be adopted as an industry standard practice for ensuring that security requirements are thoroughly discovered, documented, followed and tracked throughout the systems development life cycle.

## 2. Background

Information and systems security is a continuing problem. According to a survey performed by the Computer Security Institute (CSI) and the FBI, over 50 percent of respondents of large corporations and U.S. Government agencies reported security breaches during 2004, with reported financial losses due to these violations of over \$141 billion [5]. The losses included not only lost revenue, but also costs relating to cleanup, data loss, liability issues and most importantly, loss of customer trust [1]. While this is a declining trend seen since 2001, these figures, coupled with the research finding that current hacking tools require decreased intruder technical

knowledge [14] suggests that there are greater numbers of potential hackers [1] and suggests that despite the overwhelming efforts made on the part of organizations by means of security policies, practices, risk management, technology, security architecture and design, security for information and systems is still a serious concern.

## 2.1. IS Security Design Paradigms

There are two main paradigms for designing security solutions in Information Systems as defined by [2]. The mainstream paradigm is based on the use of checklists while the integrative paradigm uses engineering processes or logical abstractions and transformational models to combine viewpoints and functions into a single security model. These paradigms are discussed further below.

### 2.1.1. Mainstream Paradigm

The mainstream paradigm is focused on risk identification, analysis, and assessment to identify security needs and then uses checklists, best practices, and/or cookbook approaches to select known solutions to mitigate the identified risks. According to [17] there are three underlying flaws with these approaches:

1. By design, the checklist approach is template in nature, and does not address the unique and individual security needs of an organization. Furthermore, when developers encounter a situation that requires a decision on the part of management, the checklist approach cannot offer a solution.
2. Developmental duality, the conflict created by the disparate requirements of creating security and IS development, is a problem with the use of checklists, risk management and formal development.
3. The social nature of the organization is ignored with the “mechanistic and functionalistic” characteristics of checklists and formal method development.

### 2.1.2. Integrative Paradigm:

Given the limitations of mainstream approaches for security design, the need for more integrative approaches has produced integrative paradigms such as information and database modeling approaches, responsibility approaches, business process approaches and the security-modified IS development approach. There are four basic weaknesses with the existing integrative approaches, pushing the need for further development in the integrative approach arena [17]:

1. The current integrative approaches lack a comprehensive modeling support for security for the three levels of modeling, which are organizational, conceptual and technical.

2. Most of the existing approaches are difficult or sometimes impossible to integrate into the IS development process, leading to the problem of developmental duality.
3. These approaches stifle the creativity and autonomy of the developer, sometimes limiting the developmental approach the developer normally would choose to use.
4. Emerging IS methods create an ongoing gap between IS development and the implementation of the necessary security, since the methodology is not always implemented the same way in practice.

Siponen and Baskerville [17] adds meta-notation to the development process. Meta-methodology seeks to provide a means for rapidly developing computer-aided systems analysis and software engineering. Meta-notation is considered to be a key feature of most methods and meta-methods. The meta-notation includes five areas: security subjects, security objects, security constraints, security classifications and security policy. By addressing each of these five dimensions in the development process security is addressed in IS development in an integrative approach [17]. In a typical use case the actor becomes the security object, and the security classification is added to the use case [17]. Additionally, the security policy and preconditions are included in the use case to show the application of security in the use case model. This insures that security is addressed for each actor, that the appropriate policy is in place, and addressed appropriately as part of the IS design.

Lee, et al. [15] also have proposed an integrative approach. Their approach integrates mainstream security approaches with standard software engineering approaches and the software development lifecycle. This approach provides a roadmap between lifecycle processes, security engineering, and lifecycle data for the supply, development, and operations and maintenance processes. However, this is still using standard, checklist and is only successful in limiting the development duality issue discussed above with the other issues still being valid concerns.

## 2.2. Threat Analysis

A threat is defined as a set of circumstances that has the potential to cause loss or harm [16]. These circumstances may be caused intentionally, unintentionally, or even by natural events. Another perspective comes from Jennex [13] where threats are the capabilities and intentions of adversaries to exploit an information system; or any natural or unintentional event with the potential to cause harm to an information system, resulting in a degradation of an organization’s ability to fully perform its mission.

Risk analysis is the identification, categorization, and assessment of threats.

There are many methods for identifying threats including the use of Courtney's [6] exposure groups, Fisher's [8] Exposure-Identification Structure, and Hutter's [12] tree diagramming process. The most classical of these are Courtney's [6] exposure groups where six groups of threats are identified: accidental disclosure, accidental modification, accidental destruction, intentional disclosure, intentional modification, and intentional destruction. Jennex [13] identifies risks based on location and intention and offers five basic threat groups that are somewhat consistent with [6]: external accidental, external intentional, internal accidental, internal intentional, and acts of God (natural events such as equipment failures, fires, earthquakes, etc.). Each of these five threat groups has three classes of risk: destruction of data, unplanned modification of data, and unapproved disclosure of data. This paper uses the Jennex [13] threat groups. These threat groups are complete for penetration type attacks, however they do not account for attacks designed to prevent legitimate external users from accessing the system. Denial of Service attacks is a class of attacks that are external to the system and which need to be protected against with the Internet Service Provider. Additionally, it is recommended that risk analysis be used to determine which threats and risks need to be protected against.

The second step of a threat analysis is to conduct vulnerability testing to determine actual methods that can be used to implement each threat. Vulnerability testing can involve auditing to determine if security is implemented the way it was designed, external and internal testing to determine if the security design prevents specific attacks, and using posted vulnerabilities to determine applicability to the organization.

### **2.3. Risk Analysis**

In addition to threat analysis organizations conduct a risk analysis in order to determine the financial impact of threats. Results of vulnerability testing as well as industry data are used to determine expected probability of a vulnerability. This multiplied by the consequence (in dollars) should the attack succeed to generate a cost for each risk. Risk analyses are used to determine which controls are cost effective to implement.

### **2.4. Barrier Analysis**

Barrier Analysis is a method of identifying hazards or threats and, determining the effectiveness of the mitigating factors that are currently in place. Barrier

Analysis can also be used after an event has occurred to determine the cause and to help develop barriers to prevent repeat occurrences.

Barrier systems can be classified as material or physical, functional, symbolic and immaterial [11]. For example, material or physical could be containment, such as walls, doors, or restriction of physical access. By functional, we mean preventing or hindering, such as with passwords, pre-conditions or delays. Symbolic refers to countering, regulating, indicating, permission or communication, such as coding of functions, procedures, signs, work permit or clearance. Finally, by immaterial, we mean monitoring or prescribing, such as with visual inspection, checklists, rules or restrictions [11].

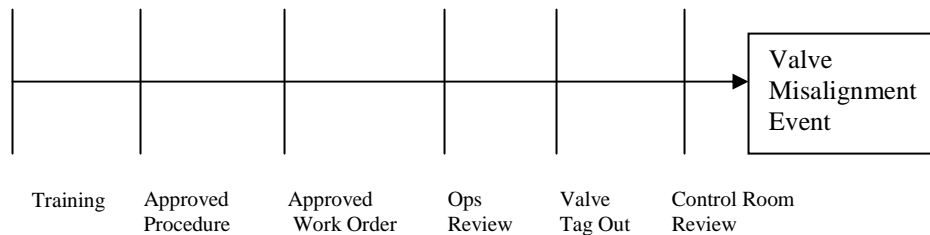
When we think of barrier analysis, it is the analysis of the placement or design of these types of barriers against possible risks or threats. Obviously, with higher risk, there will require more barriers between the object being protected, and the possible threat [13]. Barrier analysis seeks to demonstrate what is being protected and how. Furthermore, barrier analysis is used in the event of barrier failure to determine the cause, and to correct the problem by determining future preventive action [7].

The advantage of using barrier analysis is that it helps to identify the causal factors and the actions needed to correct the problems [7]. The disadvantage of barrier analysis is that the method does not insure that all failed barrier will be recognized, and the effects of the risks or threats that are applied in barrier analysis may not be properly identified [7].

## **3. Barrier Analysis And Defense In Depth As A Design Paradigm**

### **3.1. Barrier Diagrams**

In terms of the original design or purpose of barrier analysis and barrier design, barrier diagrams show the necessary ingredients for an accident, including the environmental condition that causes the harm, vulnerable people or objects that can be hurt by the condition, failure/lack of controls that are designed to keep them apart, and events that lead into the final accident [18]. Crowe [7] uses a simpler approach that utilizes the threat, the chain of barriers designed to prevent the threat, and the asset being protected. Barrier analysis is then used to assess the overall effectiveness of the barrier system, and of each individual barrier in preventing the event. Figure 1 illustrates a barrier diagram. This is the format chosen for use as a proposed method for modeling IS security.



**Figure 1, Sample Barrier Diagram [7]**

### 3.2. Defense in Depth

Hartman [10], suggests that any IS security design that relies on a single point of protection will probably be defeated. To counter this, it is suggested that defense in depth be used. Defense in depth is a concept that utilizes multiple compensating control mechanisms to prevent or reduce a threat [3]. Defense in depth uses a layered approach to increase the level of effort that would be required to damage the integrity of the system and or the data [10]. These threats could be intentional or unintentional, internal or external, or any combination of these options. The number of layers and the intensity of each defense layer are contingent on the level of the threat, and the importance of the data and the system. There must be a balance between the risk/threat, and the cost or overhead in protecting the system [1] [13].

### 4. Proposed Methodology

This paper proposes the use of barrier analysis and defense in depth as a requirement identification and design methodology and to modify the integrated approach [17]. This approach used meta-notation to add security detail to existing system development diagrams such as Use Cases [17]. This paper proposes using barrier analysis diagrams as a graphical method of identifying and documenting security requirements. Barrier diagrams are used to identify the necessary barriers to prevent events caused by credible threats identified through risk analysis. The defense in depth paradigm will be used to ensure that there are multiple security barriers between threats and events. Additionally, it is intended that the process follow the approach of integrating security design into the software development lifecycle [15].

Security requirements are identified in the requirements phase of system development by applying the security plan and through the use of vulnerability assessments and the generation of barrier diagrams. Vulnerability assessments are used to determine credible threats and key assets. Barrier diagrams are used to document those threats that need to be guarded against, those assets needing to be protected, key processes where threats could intervene, and key actors involved in those processes. Barriers will be identified based on stakeholder input, existing security policies, existing barriers in support and infrastructure systems, and using the defense in depth philosophy. Stakeholders for this phase include system analysts, users, management, and any existing security team or group. Data Flow Diagrams (DFDs), Entity Relation Diagrams (ERDs), and Use Cases should be used to assist in asset, critical process, and key actor identification. The final use of the barrier diagrams will be to gain concurrence from the users and other stakeholders that all threats and assets have been identified, and that adequate security requirements have been identified. The systems requirements document should include security requirements identified by the barrier diagrams and the actual barrier diagrams as a security model. Additionally, the organization should initiate any needed security policies to support the identified security requirements.

Security design specifications are identified in the design phase of system development. Analysts, developers, and security experts identify technologies and methods for implementing the security requirements of the identified barriers. Design specifications are determined using the security plan, policies and procedures; the existing security and technical infrastructure; and standard data integrity and fault tolerant design practices.

Specifications can be added to the diagram as meta-notation.

Developers and security experts use the barrier diagrams and security design specifications during the development phase to build security into the system. Barrier construction is determined using security policies, processes, and procedures; security infrastructure; checklists/best practices; and data integrity and fault tolerant construction practices. Implementation details are added to the diagrams as meta-notation.

Testing of the security barriers occurs during the system testing and implementation phases. The barrier diagrams should be used to generate test scripts and success criteria, user training requirements, and the implementation plan. Details of these plans and scripts should be linked by document reference to the barrier diagram meta-notation.

Integrity of the security barriers is maintained during the maintenance phase. The barrier diagrams are used to track continued implementation of security requirements and to verify that enhancements and fixes do not reduce the effectiveness of security barriers either individually or within the context of defense in depth.

Finally, barrier diagrams and analysis can be used in all phases to analyze security events. The analysis uses the diagrams to determine which barriers failed and what corrective actions or design implementations need to be taken to prevent recurrence of the event.

### 5. Barrier Diagram Example

A knowledge management system (KMS) will be used to illustrate how barrier diagrams and the defense in depth paradigm can be used. For this example it is assumed the KMS is Internet based to allow both internal and external access by employees, and the knowledge base is located in a single database on a single server.

The requirements phase involved the systems analyst, key knowledge users, management, and the security group. There is consensus that all knowledge in the knowledge base needs to be protected and that project experience related to the core business area is critical. There is also consensus that all five threat groups; external accidental, external intentional, internal accidental, internal intentional, and acts of God for all three degrees (except Acts of God which only has inappropriate destruction), inappropriate disclosure, modification, and destruction, need to be protected against. Identification of key knowledge base assets is accomplished through meta-notation in the form of comments added to the Entity Relation Diagram or Data Dictionary. The basic barrier diagram for the external intentional and accidental threats with inappropriate disclosure, modification or destruction is shown in Figure 2. Figure 3 is the diagram for the internal intentional and accidental threats with inappropriate disclosure, modification, and destruction. Figure 4 is the diagram for the Acts of God threats with inappropriate destruction. Requirements identified from the diagrams are added to the system requirements specification (SRS) with the threat table and barrier diagrams as supporting documentation. Finally, the current security plan and policies are reviewed to ensure policies exist for the identified security requirements.

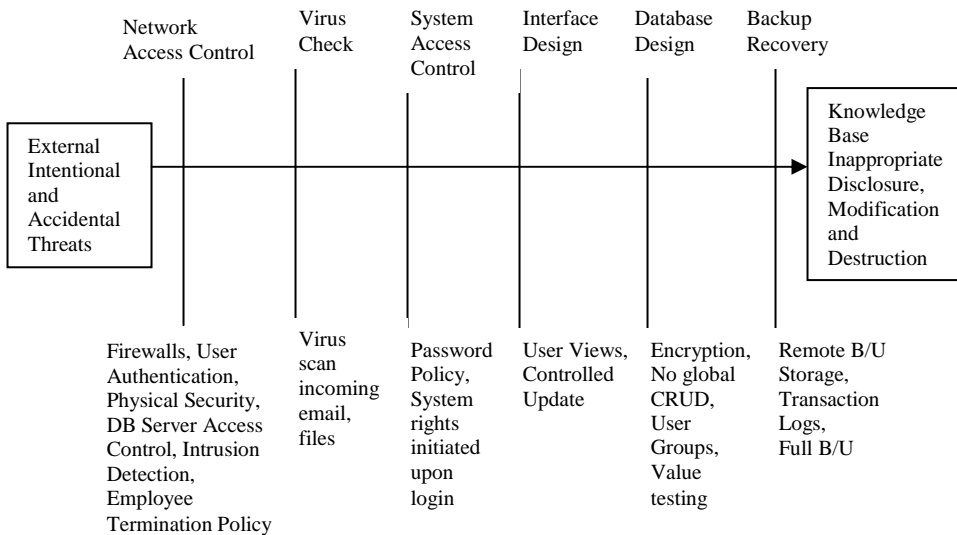
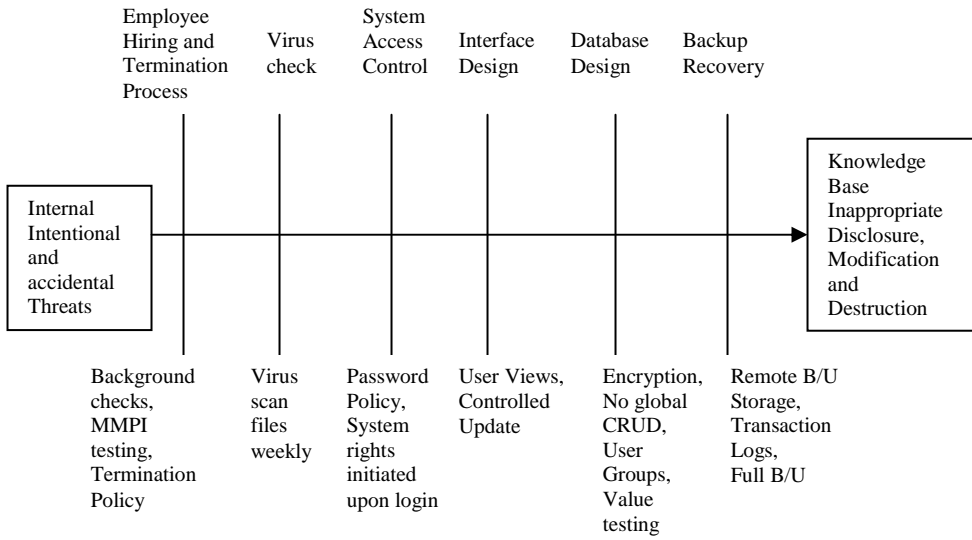
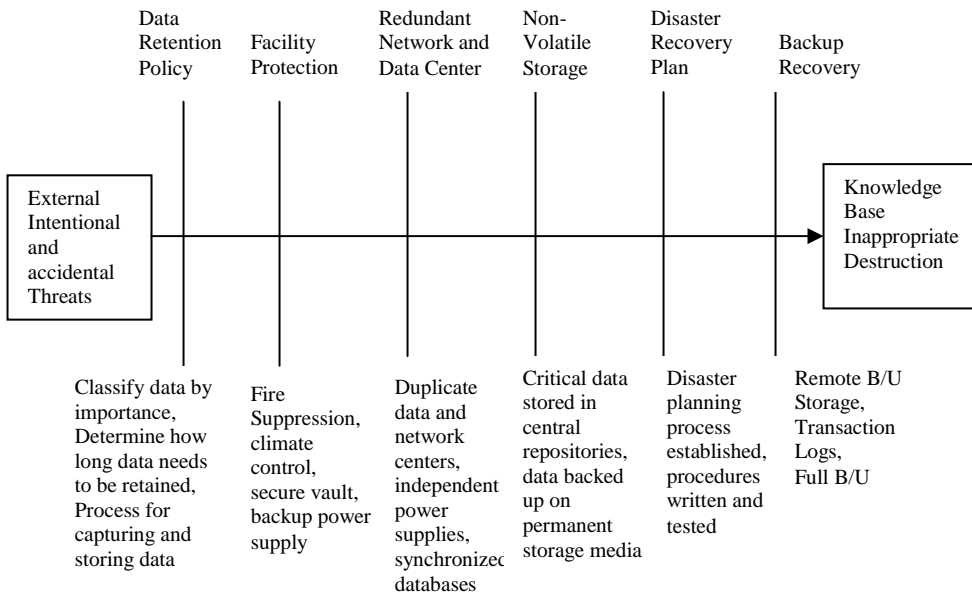


Figure 2, External Intentional and Accidental Threats Requirements phase barrier diagram



**Figure 3, Internal Intentional and Accidental Threats Requirements phase barrier diagram**



**Figure 4, Acts of God Threats Requirements phase barrier diagram**

Boxes represent the threat entity (left end) (this can also be specific vulnerabilities) and the asset to be protected (right end). The line connecting the boxes represents the path the threat takes to get to the protected asset. The lines perpendicular to the threat path are the barriers erected to prevent the threat from reaching the protected asset. Top text lists the barrier while lower text lists the requirements. Barriers and requirements vary based on

the threat group with some overlap expected since security strategies may be applied to multiple threats.

These barrier diagrams illustrate defense in depth by showing that there are six barriers that integrate multiple, security technologies and approaches with existing security policies and infrastructure for each set of threats. Figure 2 illustrates this approach. Firewalls are used to

screen for authorized users, passwords are used to authenticate user identities, user rights groups and user views are used to limit user access to what is needed, data entry testing is used to validate data before it is stored to prevent potentially incorrect data from being stored, encryption is used to prevent unauthorized data disclosure, and backup and recovery is used in case all the barriers fail and the threat entity destroys or modifies the data.

Analysts can use Figures 2, 3, and 4 as is or can combine the diagrams into a single master diagram. Either way, the security requirements from each of the barrier diagrams are combined to generate the final requirements specification. Additionally, the barrier diagrams are useful for communicating these requirements to the stakeholders and security specialists and in gaining their concurrence and approval.

Once the security requirements are approved, the analyst uses them to generate security design specifications. The analyst can continue to develop the meta-notation on the barrier diagram to include more layers of detail, specifically design specifications. Alternately, this detail can be added by supplementing the barrier diagram with tables of design specifications tied to requirements. Table 1 illustrates the table approach to documenting design specifications for the database design barrier in Figures 2 and 3. The table approach is recommended to keep the barrier diagrams readable. This detail is generated during the design phase of system development and could be extracted or generated using the logical data model and physical table design chart. Design detail is expressed as design specifications for each of the functional requirement specifications.

**Table 1, Sample Barrier Diagram Requirements**

Functional Requirement	Design Requirement
Encrypt Critical Knowledge	Encrypt Project_Report.Lessons_Learned attribute
No Global CRUD rights	Instantiate database rights upon login to system
Establish CRUD rights via user groups	Establish Admin user group with administrative rights
	Establish Mgmt user group with partial RU rights
	Establish Update user group with full CRUD rights
	Establish Project Manager user group with partial CRUD rights
Check data input before writing to DB	Establish Knowledge user group with partial CRU rights
	Use choice lists for attributes with finite selections
	Range check numeric and currency attributes
	Format attribute input for those with set formats

During the coding and testing phase, unit and functional test scripts are generated using the design specifications. The barrier diagrams are used to generate integrated system test plans that establish initial conditions, expected system responses, and allow for a series of monitored attacks by a variety of attacker profiles. Attacker profiles are designed to fit the expected attacker profiles of the analyzed vulnerabilities. The barrier diagrams show how the security system is supposed to work and provides a basis for analyzing failures in individual barriers. Testers need to develop scenarios that test the ability of total security system, i.e. of all the barriers working together, to protect the assets. Testing is completed when scenarios cannot be generated that successfully penetrate the asset. If scenarios are found that result in penetration, then designers need to revise the security system to counter it. Scenarios that result in some barriers being defeated need to be reviewed to determine if there are vulnerabilities in the barrier design. Testing can be performed by using “white-hat” hackers to attempt penetration. These testers would use their skills to attempt penetration and to identify vulnerabilities in the overall security plan should they penetrate the outer network security. Additionally, automated network security scans can be used to identify vulnerabilities in network security. Ultimately, it should be assumed that network security will be penetrated and the other barriers need to be tested to determine their effectiveness in protecting or minimizing damage to the assets.

The barrier diagrams and vulnerability assessment are used during the maintenance phase to assess system changes for impact to the security design. As new knowledge bases are added, they are assessed for criticality and the need for encryption. As the organization expands to new locations the diagrams provide a blueprint for designing local facility and network security. As employees change jobs, are hired, or quit, the diagrams provide guidance to what user groups need to be modified. Finally, as new threats are identified, the diagrams are used to assess any needed changes to the security system.

The final use of the diagrams is in assessing the impact of attempted penetrations, events, or internal acts that defeat some or all of the barriers designed to prevent the penetration, event, or internal act. Security specialists can use the diagrams to identify security implementations that failed and what caused the failure. The goal is to identify user behaviors, policy issues, or weak technologies that need changing or improving.

## 6. Experience Using Barrier Diagrams And Defense In Depth

Barrier diagrams and defense in depth was pilot tested by a team of graduate students designing a web site for the International Student Center. The web application is for potential students to contact the university as well as for existing international students to participate in the International Students Association. Functions included databases, web forms, a schedule of events and meetings, and supports the use of online chat for members and potential students. Security requirements were determined through a threat analysis and the generation of barrier diagrams. The diagrams were generated based on discussions with the chair of the university's IS security committee and were validated as correct. Design specifications were generated, documented in tables, and discussed with the chair of the university's IS security committee for approval. The specifications were approved and the final design documented in a system design specification. Interviews with the project team found that the diagrams were very effective and useful in identifying the full set of security requirements needed for the system and in generating the system security design specifications. The diagrams were also found to be effective in conveying security requirements. The team also stated that the diagrams helped them discover weaknesses that they would have otherwise missed.

## 7. Conclusions

Barrier diagrams provide a graphical tool for identifying and determining security requirements. Graphical tools enhance understanding and communications between stakeholders. This tool is expected to enhance understanding of, and compliance with security requirements.

The Defense in Depth paradigm enhances security by providing multiple barriers to prevent threats from causing damaging events. When coupled with barrier diagrams it provides a tool for all stakeholders to integrate security needs and efforts and provides a process for ensuring security measures work together and not against each other.

Combining these tools provides a means for integrating security design and implementation across the SDLC. While the traditional lifecycle was discussed, these tools can be applied to any lifecycle approach. Integrating security design and implementation into the lifecycle should improve overall system quality.

Finally, barrier diagrams and analysis can be used much as it is used in the United States nuclear industry, as a tool for determining root cause. This tool provides an analysis

tool for determining what failed and how preventative actions can be taken to prevent future security breaches.

## 8. References

- [1] Allen, J.H., Mikoski Jr., E.F., Nixon, K.M., and Skillman, D.L., 2002, Common sense guide for senior managers: top ten recommended information security practices, *Internet Security Alliance*, 1<sup>st</sup> Edition.
- [2] Baskerville, R., 1993, Information systems security design methods: implications for information systems development, *ACM Computing Surveys*, 25(4), pp. 375-414.
- [3] Bass, T. and Robichaux, R., 2002, Defense in depth revisited: qualitative risk analysis methodology for complex network-centric operations, <http://www.silkroad.com/papers/pdf/archives/defense-in-depth-revisited-original.pdf>.
- [4] Clemens, P.L., 2002, Energy Flow/Barrier Analysis, 3<sup>rd</sup> Edition", <http://www.sverdrup.com/safety/energy.pdf>.
- [5] Computer Security Institute, 2005 CSII/FBI computer crime and security survey, *Computer Security Issues and Trends*.
- [6] Courtney, R., 1997, Security Risk Assessment in Electronic Data Processing, *AFIPS Proceedings of the National Computer Conference 46*, pp. 97-104.
- [7] Crowe, D., 1990, Root Cause Training Course for Catawba Nuclear Station, General Physics Corporation.
- [8] Fisher, R., 1984, Information Systems Security, Prentice-Hall, Englewood Cliffs, NJ.
- [9] Haddon Jr., W., 1973, Energy damage and the ten countermeasure strategies, *Human Factors Journal*, 15.
- [10] Hartman, S., 2001, Securing E-Commerce: an overview of defense in-depth, [http://www.sans.org/rr/start/sec\\_ecom.php](http://www.sans.org/rr/start/sec_ecom.php).
- [11] Hollnagel, E., 1999, Accident analysis and barrier functions, <http://www.hai.uu.se/projects/train/papers/accidentanalysis.pdf>.
- [12] Hutter, D., 2002, Security Engineering, <http://www.dfki.de/~hutter/lehre/sicherheit/securityengineering.ppt>.
- [13] Jennex, M.E., 2003, Security Design, System Design Lecture, IDS 697, San Diego State University, 4/21/03.
- [14] Jennex, M.E. and Walters, A., 2003, A comparison of knowledge requirements for operating hacker and security tools, *the Security Conference, Information Institute*.
- [15] Lee, Y., Lee, Z., and Lee, C.K., 2002, A study of integrating the security engineering process into the software lifecycle process standard (IEEE/EIA 12207), *6th Americas Conference on Information Systems, AMCIS*, pp. 451-457.
- [16] Pfleeger, C.P. and Pfleeger, S.L., 2003, Security in Computing, 3<sup>rd</sup> Edition, Prentice-Hall, Upper Saddle River, NJ.
- [17] Siponen, M. and Baskerville, R., 2001, A new paradigm for adding security into IS development methods, *8<sup>th</sup> Annual Working Conference on Information Security Management and Small Systems Security*.
- [18] Trost, W.A. and Nertney, R.J., 1995, Barrier Analysis, <http://ryker.eh.doe.gov/analysis/trac/29/trac29.html>.