Teaching Case

IS Security Requirements Identification from Conceptual Models in Systems Analysis and Design: The Fun & Fitness, Inc. Case[1]

Janine L. Spears
School of Computing
DePaul University
Chicago, IL 60604, USA
jspears@cdm.depaul.edu

James L. Parrish Jr.
Nova Southeastern University
Fort Lauderdale, FL 33314, USA
jlparrish@nova.edu

ABSTRACT

This teaching case introduces students to a relatively simple approach to identifying and documenting security requirements within conceptual models that are commonly taught in systems analysis and design courses. An introduction to information security is provided, followed by a classroom example of a fictitious company, Fun & Fitness, in the process of updating its e-Commerce site for class registrations. The case illustrates how UML class diagrams can be used for information classification, data input validation, and regulatory compliance considerations; how a UML use case diagram can be transformed into a "misuse case" diagram to identify threats and countermeasures to functional use cases; and how a data flow diagram may be used to analyze and document threats and countermeasures to data stores, data flows, processes, and external entities using the STRIDE approach developed by Microsoft. The case is geared toward a systems analyst who does not have former training in IS security, and is suitable for upper-division undergraduate and graduate courses.

Keywords: Information assurance and security, Requirements analysis & specification, Business modeling, Modeling Systems analysis and design, Unified modeling language (UML)

1. INTRODUCTION

Reducing information security vulnerabilities in software is a daunting task that organizations face today. Industry reports on information security breaches indicate that web applications remain a persistent target for attacks (Hewlett-Packard Development Company 2011). In a study by the Ponemon Institute (2012b) on application security, developers cited mobile applications as the most likely to disrupt their organizations' business operations, yet 65% of those surveyed indicated that mobile applications in their organizations are not tested. Of the 256 developers surveyed in the study, 79% indicated their organizations had no or an inefficient, ad-hoc process for building security into their applications, and 71% believed that security was not adequately covered in the systems development lifecycle (SDLC) within their organizations. Thus, information systems (IS) security at the application-level is needed, yet under-developed, ultimately contributing to expensive data breaches.

According to the Ponemon Institute (2012a), data breaches cost the average firm $194 USD for each customer record compromised and $5.5 million USD per incident. In the widely-referenced annual Verizon Data Breach Investigations Report[2] (2013), financial motives accounted for 75% of their investigated data breaches, with financial payments and user credentials being the most targeted data types. In spite of the high costs of data breach notifications when personal information has been compromised, less than 35% of respondents in a study by PricewaterhouseCoopers

(2013) believed their organizations had an accurate inventory of where personal data for employees and customers are collected, transmitted, and stored – making it less likely that these data will be adequately protected.

These findings suggest that the time is ripe for IS professionals to begin incorporating security into the analysis and design of an IS as a means to reduce security vulnerabilities and data breaches. While leading security frameworks for integrating security into the SDLC advocate planning for security during software initiation (NIST 2008) and integrating security into software design (Microsoft 2010), guidance is limited on how security may be integrated into the "analysis" phase of the SDLC as systems analysts gather system requirements. Therefore, this teaching case is focused on integrating security into conceptual (i.e., logical) models developed in the analysis and early design phases of the SDLC (see Figure 1) while systems analysts gather functional requirements. In doing so, security will be more tightly integrated into system design. Moreover, conceptual models that incorporate security can lay a foundation from which systems developers, systems engineers, and security professionals can build upon in subsequent design, development, implementation, and maintenance phases of the SDLC.

System modeling has traditionally focused on functional requirements (i.e., what the system should do), typically excluding security requirements (i.e., what and how the system should protect). Though computer scientists have proposed adding security to UML use case diagrams (Sindre and Opdahl 2000; 2008) and modeling security requirements for specific types of applications such as web applications (Salini and Kanmani 2013) or specific system types such as distributed systems (Uzunov et al. 2012), these methods have not been widely adapted in IS systems analysis and design courses. Indeed, scant coverage of security has been found in systems analysis and design textbooks (Biros et al. 2007; Parrish et al. 2009). In absence of including security earlier in the SDLC, security as part of systems development is reactionary and typically "bolted on" during system implementation or maintenance (Parrish et al. 2009; Ponemon Institute 2012b; van Wyk and McGraw 2005), at which point it may be cost prohibitive to make significant security design changes.

Even if practitioners wanted to address security earlier in the SDLC, it is likely that many practitioners do not know how to incorporate security into modeling techniques (Mead and McGraw 2005; Uzunov et al. 2012) as part of requirements gathering. To that end, this teaching case introduces students to a relatively simple approach to identifying and documenting security requirements within conceptual models that are commonly taught in systems analysis and design courses. In particular, this teaching case illustrates how to elicit security requirements from object-oriented UML use case and class diagrams, and from structured data flow diagrams.

This teaching case is geared toward information systems analysts. For the purpose of this article, systems analysts refer to students and practitioners who work on the analysis and design of business information systems, including identifying system functionality, and analyzing or designing business process workflows, user/system interfaces, and data attributes. System architecture and network security are outside the scope of this teaching case.

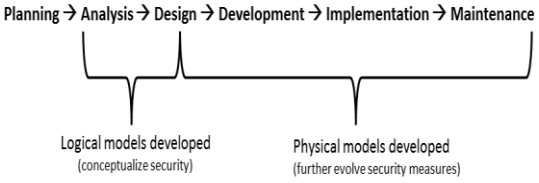

Figure 1: Modeling within the SDLC

2. SYSTEMS ANALYSTS AND IS SECURITY

On a systems development project, systems analysts work with system users to model business processes, identify functional system requirements, and design both logical and physical solutions; thus, systems analysts possess valuable analytical skills that are often applied in a business context. We suggest that systems analysts' knowledge of business processes, coupled with their skill in modeling those processes, places them in a unique position to incorporate security as part of the conceptualization of IS requirements. As systems analysts collect a vast amount of system data from forms, interviews, observations, and other sources during requirements gathering, they can also identify and conceptually model information classifications, access control requirements, and risk tolerance levels (Biros et al. 2007), among other security and privacy considerations.

While the roles of systems developers, systems engineers, security analysts, and security architects are important in the creation of secure software, their impact is more in the physical design of the code and architecture, when physical models are developed (see Figure 1). In contrast, the role that is among the most likely to work with users on conceptual (logical) modeling of a business process during system analysis is the systems analyst. While a systems analyst may not typically be the best role to define firewall specifications, it is among the best roles in the IS/IT organization to work with users and other stakeholders to conceptualize security requirements within a business process. Just as user participation is essential to eliciting functional business requirements for an IS, user participation is also essential to eliciting requirements for protecting information within business processes given their knowledge of information usage (Spears and Barki 2010).

3. SECURITY REQUIREMENTS IDENTIFICATION PROCESS

In this section, we present a 4-step process for identifying security requirements that will subsequently be illustrated in a classroom example. This approach was described by Tondel, Jaatum and Meland (2008) based on a literature review they conducted on various approaches to security requirements engineering. The approach is intended to be "palatable to regular software developers" (as opposed to security experts), because "all software development projects need a well-balanced amount of security awareness right from the beginning" (p. 20). The 4-step process includes:

1. Identify security objectives

2. Identify information assets
3. Identify threats to information assets
4. Document security requirements

We begin our process description by introducing the systems analyst to basic security concepts. ***Information security*** is defined as the preservation of confidentiality, integrity and availability of information; in addition, other properties, such as authenticity, accountability, non-repudiation, and reliability can also be involved (ISO/IEC 27002 2005). ***Confidentiality*** refers to restricting access to information to only those who are authorized to have access. ***Integrity*** refers to protecting information and information systems from corruption, damage, or destruction. ***Availability*** refers to ensuring a system is available for access by authorized personnel when needed.

Security requirements identification begins with identifying the security objectives that you wish to achieve. A ***security objective*** is defined as a goal that you wish to achieve with respect to securing an information system. Examples of security objectives might be regulatory compliance or adherence to industry security standards.

The second step is to identify information assets as a means to prioritize security efforts. An asset is something of value to an organization. An ***information asset*** may be data, hardware, software, process, or people. Consider an asset's value from the perspective of the customer, system owner, and potential attacker (Tondel et al. 2008).

Third, threats to information assets are identified. An IS security ***threat*** is a potential cause of an unwanted incident, which may result in harm to a system or organization. Threats may be intentional or unintentional, and may originate from external or internal sources. For example, an employee can accidentally or intentionally implement programming code that crashes a production server. Another example of a threat is an attack by a hacker. A ***hacker*** is an external agent who intentionally circumvents security measures. A ***vulnerability*** is an existing weakness within the computing environment that may be exploited.

Various approaches to gathering security requirements differ on the extent to which requirements include concrete security measures (Tondel et al. 2008). However, as part of threat analysis in this teaching case, we also identify security measures to counter the identified threats. A ***countermeasure*** (also referred to as a ***safeguard*** or ***control***) is any policy, technology, or procedure designed to detect, prevent, or reduce a security threat. A common countermeasure is a security policy. A ***security policy*** is a stated set of security objectives along with some set of mechanisms for ensuring those objectives are met. For example if we wish to prevent unauthorized persons from accessing an IS (objective), we may have a policy that requires all system users to authenticate (i.e., verify their identity) to the system using a user ID and password (mechanism). Having a person authenticate to the system also helps to enforce the principle of ***non-repudiation***, which is a way to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message (webopedia.com).

Finally, the fourth step in the risk identification process is to document the security requirements identified in earlier steps. Security requirements may be documented alongside functional system requirements. In doing so, security requirements will gain visibility among stakeholders viewing requirements documentation. Similar to functional requirements, security requirements can be prioritized and included in traceability matrices and test scripts. In other words, security requirements become part of system requirements.

4. APPLICABILITY OF THIS TEACHING CASE

This teaching case is particularly suited to students majoring in information systems and is applicable to both upper-division undergraduate and graduate courses. We have taught conceptual modeling of IS security risk in both a systems analysis course that is typically taken as the first course in a Masters-level IS degree program, and in an elective organizational modeling course. Students are initially taught how to construct common types of diagrams used in systems analysis and design, such as various types of UML diagrams, data flow diagrams (DFD), entity-relationship diagrams (ERD), etc. After students have gained an understanding of conceptual modeling, they are then taught how to extend their diagrams by adding security and privacy risk considerations. For this teaching case, a basic understanding of conceptual modeling techniques is assumed, such as how to create a DFD, use case, or class diagram.

In teaching conceptual modeling of IS security threats and countermeasures in use case diagrams, referred to as misuse case diagrams, we have found that IS students not only grasp how to conceptualize risk within a business process, misuse case diagrams are commonly named the diagram students most enjoy creating. Students have expressed enjoying the mental exercise of de-constructing a business process in an effort to consider where the process may break down or otherwise result in loss. Several students working in industry as systems or risk analysts stated they like the visual communication that misuse case diagrams enable with their system users, since information security is challenging for analysts to explain and difficult for their users to grasp. Thus, diagrams are an effective communications medium.

5. CLASSROOM EXAMPLE

Imagine that you are a systems analyst at *Fun & Fitness, Inc.* working on systems analysis and design for an e-Commerce system revision. After conducting a series of meetings with key users and reviewing existing system artifacts, you've learned the Company Background, provided in a narrative below. Next, a UML use case diagram is constructed to model an understanding of the core business objectives and functionality of the system. Conceptual models provide a means for an analyst to gain a greater understanding of the problem domain and to communicate that understanding with system stakeholders. The *Fun & Fitness* case then continues with an illustration of the 4-step process for security requirements identification.

5.1 Learning Objectives

1. Identify security requirements as part of systems requirements gathering.

2. Raise student awareness of security requirements in systems development projects.

5.2 Company Background

Fun & Fitness offers a variety of instructor-led, exercise classes, such as yoga, Zumba, pilates, etc. On the *Fun & Fitness* e-commerce web site, a customer can view the exercise class schedule. Class registration is accessible from the schedule, though it is possible that a customer may view the schedule without registering for a class. Both members and non-members (collectively referred to as "customers") may view the class schedule and register for an exercise class. Approximately 60 percent of *Fun & Fitness* customers are members. Incentives for membership include reduced prices on exercise classes.

The scheduling manager establishes an attendance limit for each exercise class based on class popularity, room availability, and input from the marketing and accounting managers. Attendance limits are typically set at 20 or 30 people. Class registrations may be made online in advance up to 1 hour before the scheduled class time, or at the door. Several popular exercise classes fill up quickly, so customers are encouraged to register for classes online in advance. The class registration page shows the class attendance limit and the number of people currently registered. Unfortunately, customers often register for a class, but ultimately do not attend the class. Though the current web site allows customers to register for a class after registrations have exceeded the attendance limit, prospective customers visiting the online fitness schedule may assume the class is full, and thus no longer available. Consequently, *Fun & Fitness* may be losing potential income if customers choose not to register or attend a class because they perceive the class to be full to capacity and unable to accept additional registrations. In this way, scheduling, forecasting, and potential revenue are hampered. In an effort to improve scheduling, forecasting, and potential revenue, *Fun & Fitness* management has decided to require payment at the time of registration. In other words, a customer cannot register for a class unless payment is made.

Online payments are made by customer credit card. Verification and processing of credit card transactions are handled by an external credit card processing vendor's system. Upon successful completion of a credit card payment, the customer is sent an email confirmation, and a financial log is updated with accounting information to be exported to *Fun & Fitness*' accounting system.

When a customer pays for class registration, he or she is not required to have an online account with *Fun & Fitness*. However, only members are able to access membership account information online. Members are able to manage account information online, such as updating email and street addresses, etc. Members must log into their account to view or update account information.

On average, members take 2.5 exercise classes per week. As a convenience to members, credit card payment information may be stored as part of the member's account information in order to save time by eliminating the need to re-enter payment information each time a member registers for a class. The number of classes taken by non-members varies widely, from only one to many. Currently, only members are allowed to store payment information. In recent months, several non-members who reportedly take several classes a year have requested the ability to store credit card payment information so that they do not have to reenter this information every time they register for a *Fun & Fitness* exercise class. The *Fun & Fitness* management team is considering the implications of this request, and has not yet decided if non-members will be allowed to store payment information. From a data security and regulatory compliance perspective, what are the implications, if any, of accommodating this request? What is gained or lost by *Fun & Fitness* from a benefit-risk perspective?

Finally, the marketing analysts email customers discount promotions for exercise classes. A customer can "opt out" of receiving promotional emails. The marketing manager approves all promotional emails before they are sent to customers. The marketing manager also generates various reports.

6. USER INTERACTION WITH SYSTEM FUNCTIONALITY IN A USE CASE DIAGRAM

The use case diagram in Figure 2 summarizes system functionality requirements from a user's perspective. A use case description may be used to define more detail on the actors, triggers, outcomes, and sequential tasks associated with a given use case in the diagram. For this teaching case, we are not including use case descriptions. Next, we illustrate an informal, relatively simple 4-step process for including security in system requirements identification.

7. STEP ONE: IDENTIFY SECURITY OBJECTIVES

Security objectives are high-level requirements or goals that are most important to internal and external customers, including requirements for compliance with relevant legislation, policies, and standards (Tondel et al. 2008). Security objectives may be identified by interviewing relevant stakeholders and consulting organizational policies and industry standards. In the remainder of this section, we provide an overview of security-related regulations applicable to *Fun & Fitness*.

7.1. Payment Card Industry's Data Security Standard (PCI DSS)

Fun & Fitness must comply with PCI DSS because it accepts credit card payments. PCI DSS is an industry standard that was developed by a council on behalf of 5 credit card companies: Visa, MasterCard, American Express, JCB, and Discover. Any organization that accepts, transmits, or processes credit card payments with these card brands must comply with PCI DSS; otherwise, the organization faces significant fees from its bank(s). The purpose of PCI DSS is to protect "cardholder" data. To that end, PCI DSS contains 12 security requirements related to cardholder data storage and transmission (PCI Security Standards Council 2010).

For the purposes of this teaching case, we focus on PCI DSS Requirements 3 and 4 that are associated with the goal to Protect Cardholder Data, while stored and in transit. These

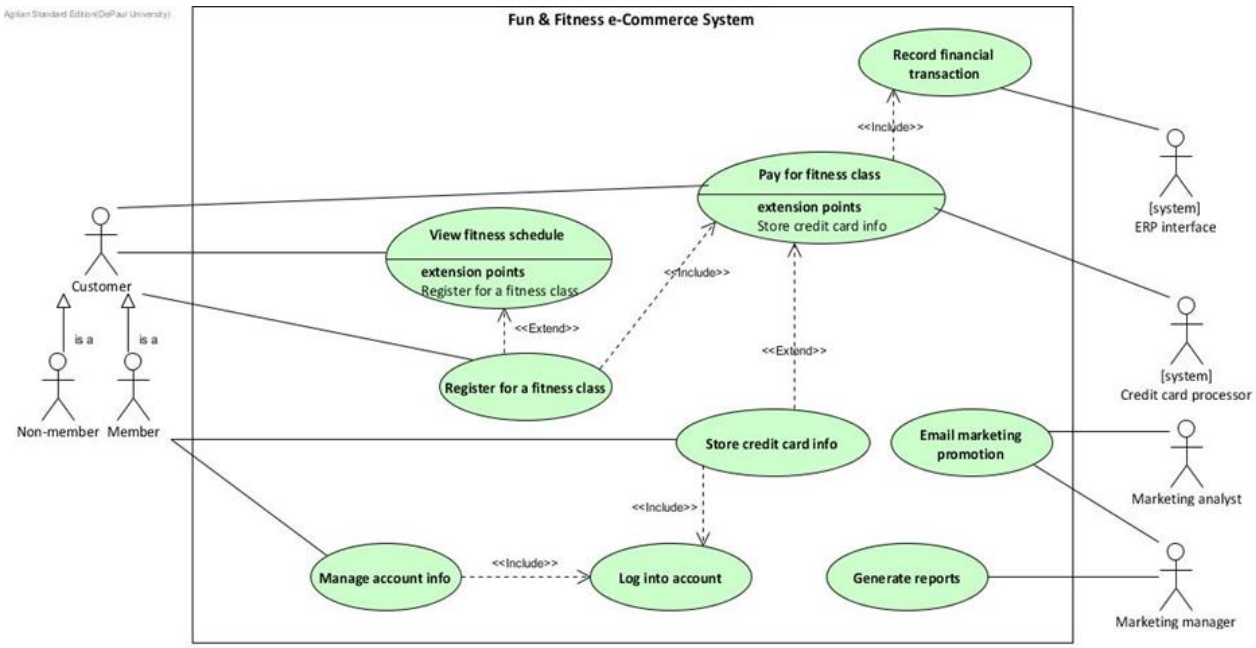

Figure 2. Use Case Diagram for *Fun & Fitness* E-Commerce System

requirements state that cardholder authentication data (e.g., CVV2 code) are not to be stored; minimal, if any, cardholder account data (e.g., number and expiration date) should be stored; cardholder data that is stored must be encrypted; and cardholder data must be encrypted during transmission. PCI DSS also requires strong access control measures for cardholder data, and an organization must have an information security policy. In complying with PCI DSS, an organization needs to determine which business processes handle cardholder data; where is cardholder data located; should the data be retained; how to manage encryption; and who should have access. Policies should then be documented.

7.2. Breach Notification Laws

Over 40 U.S. states require an organization to notify individuals when the organization has experienced a data breach involving personally identifiable information (PII) of individuals residing in a given state. Notification is not required if data are encrypted (e.g., on a stolen laptop), and thus, unreadable. Several international governments are considering similar legislation. Though the definition of PII varies by governing entity, it generally refers to a person's name when combined with any one of the following: email address, postal address, financial account number, driver's license number, or social security number.

7.3. Fair Information Practices

Several governments worldwide recognize a set of "fair information practices" designed to aid privacy protection (e.g., U.S. FTC 2007). These practices are mandated by some governments (e.g., the European Union), and voluntary by others (e.g., the U.S.). For the purposes of this teaching case, we bring to your attention the fair information practices of *notice* (i.e., an individual should know what personal information is collected on him/her, by whom, and how it is used); and *choice* (i.e., an individual should be able to opt out of personal information collection). In e-commerce, notice is typically provided by privacy notices posted on an organization's web site, preferably on the home page and pages where personal information is collected (Culnan and Carlin 2009).

8. STEP TWO: IDENTIFY INFORMATION ASSETS

Review the case narrative (or interview notes and other sources of input on system requirements for a real project). Review any use case diagrams, use case descriptions, and class diagram. From this input, identify the key assets of this system. In other words, what are the objects or attributes that are particularly valuable to *Fun & Fitness*? Value may be tangible, such as financial gain or loss, or intangible, such as reputation gain or loss. If assets are identified that were not originally captured in your class diagram, it may be necessary to add them, as appropriate. We have identified the following assets:

- Credit card information
- Customer email list
- Exercise fitness class schedule
- Financial transaction
- Marketing promotion

If customer names, along with their credit card ("cardholder") information or email addresses are breached, regulatory requirements mandate that customers be notified. Such notification would result in a financial costs incurred, and may possibly have a negative impact on the company's reputation, including potentially losing customers. An accurate and available class schedule has a direct impact on

company revenue in terms of customers' ability to register for classes. The reliability of financial data is critical to investors and the government, and marketing promotions can provide a competitive advantage. For these reasons, these data have been classified as information assets.

8.1 Information Classification

Information classification is closely related to asset identification and is a critical first step to identifying information that an organization wants to protect. Information should be protected according to its classification. Otherwise, resources may be wasted in implementing unnecessary controls or sensitive information may go unprotected. The classification of data attributes is generally based on an organization's industry, internal operations, risk tolerance, and regulatory requirements. Common classifications are shown in Table 1, though an organization may use different names or have additional classifications.

Classification	Description	Example
Public	Information is publicly available and would not cause harm to the organization if disclosed	Public web site content; publicly-reported financial statements; current and past marketing promotions
For internal use only	Information is typically required to perform day-to-day internal operations, and is not to be made publicly available	Internal employee directory; employee or customer ID (other than a government issued identifier)
Confidential	Sensitive information that if disclosed, compromised or destroyed in an unauthorized manner would directly or indirectly adversely impact the organization, its customers or employees	Customer or employee PII; unreleased financial data; unreleased marketing promotions

Table 1. Sample Information Classification

Consensus is needed among organizational stakeholders on what classification to assign information, as well as the lifecycle of the classification. For example, should marketing promotions be classified? At some point, the promotion becomes public knowledge; however, an organization may want to keep their promotions secret until ready for release. In such case, a marketing promotion may be classified as Confidential from inception until release date, and then re-classified as Public on the release date. Certain aspects of the promotion may be Confidential, while other details are For Internal Use Only. The same logic applies to the classification lifecycle of sales revenue reported in financial statements.

In addition to gaining consensus, information classification must be communicated to all internal members who have access to information classified beyond Public. Otherwise, one employee may consider a particular type of data as confidential, while another does not. For example, is a birthdate or phone number considered to be personal information? Is employee salary officially considered to be confidential? The answer to both questions depends on how an organization chooses to classify its data. If any of these data attributes are deemed confidential, measures should be implemented to protect confidentiality, such as more controlled access, encryption, etc. In some cases, an organizational member may not want to classify certain data as confidential in order to alleviate cumbersome security controls. Again, consensus and communication are needed within the organization on information classification. Once information has been classified, security resources can be more effectively allocated by only focusing on protecting those information assets that process or store confidential information.

We propose using a UML class diagram as a means to define and communicate information classifications. In addition to classifying information as public, for internal use, or confidential, a UML class diagram can also be used to identify data attributes that fall within the scope of regulatory requirements. By classifying information, appropriate controls may be considered.

Create a class diagram for the *Fun & Fitness* e-Commerce class registration system:

1. Identify classes (objects) within the e-Commerce system.

2. For each class, identify its attributes (e.g., based on the case narrative above and attributes you believe are necessary within the context of this IS).

3. For each class, identify those attributes that are confidential and denote with "CONF" in the attribute name. Attributes with particular regulatory constraints may be denoted with "REG" in the attribute name.

4. For each class, identify operations (called methods in physical models). Do not include operations that are available to all classes, such as create(), edit() and delete() an instance.

5. For each attribute that was identified as confidential, add an operation to that class for encryption().

6. For each attribute of each class, determine if a validation check is needed. If so, include the validation as an operation with a parameter list of attributes, or the "all" parameter (described next).

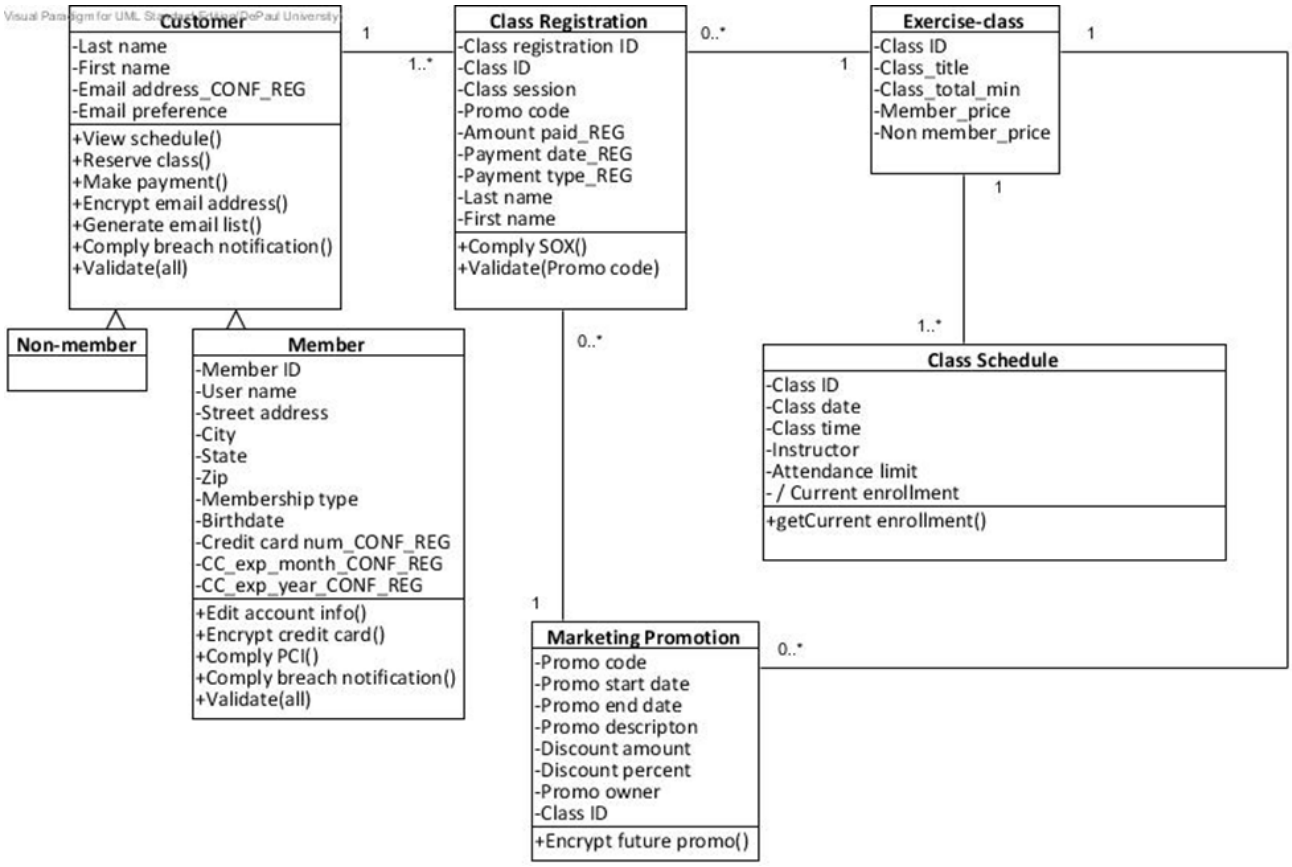

Figure 3. Class Diagram with Data Classification, Validation, and Compliance Requirements

8.2 Additional Uses of Class Diagrams in Security Requirements Identification

Identifying requirements for data input validation may also be facilitated in a UML class diagram (or an entity-relationship diagram). Many of the web program code vulnerabilities are related to un-validated input. For example, a web page may allow a system user to key in his or her login ID and password. If this input is not validated, a security vulnerability (i.e., weakness) is created that could allow someone to input and send remote commands to the database server. This example illustrates a SQL command injection vulnerability, which is a technique used to take advantage of non-validated input vulnerabilities to pass SQL commands through a web application for execution by a backend database.

Therefore, a data attribute should be validated if it will be entered by a system user or transmitted from a system interface. Validation requirements can be captured in a class diagram as an operation at the attribute or the class level. If the entire class is created from user input (e.g., a customer account information), a validation() operation can include "all" as the parameter to validate all attributes. Otherwise, if individual attributes within a class require validation, include relevant attribute names as parameters in a validation operation; the Promo_code attribute in the Class Registration class is an example.

9. STEP THREE: IDENTIFY THREATS WITH A MISUSE CASE DIAGRAM

A use case diagram depicts how stakeholders (called "actors") interact with a system, and the desired system functionality from the stakeholder's perspective. Conversely, a misuse case diagram provides a means to model undesirable system events that threaten successful completion of the system functions that were modeled as use cases (Sindre and Opdahl 2008). If a business function carried out within a system is viewed as an organizational objective and represented as a use case, then a misuse case depicts a threat to an organizational objective. That is, while use cases illustrate desirable system functionality, misuse cases illustrate undesirable events that could occur and disrupt the desirable system functionality. In a misuse case diagram, threats are modeled as misuse cases, threat agents as mis-actors, mitigating controls that counter the specified threats as use cases, and the associations between these components (Sindre and Opdahl 2000; 2008). Just as use case diagrams are particularly useful when analyzing a business process where stakeholder interaction is of primary interest, misuse case diagrams are particularly useful when analyzing data security threats posed by stakeholders.

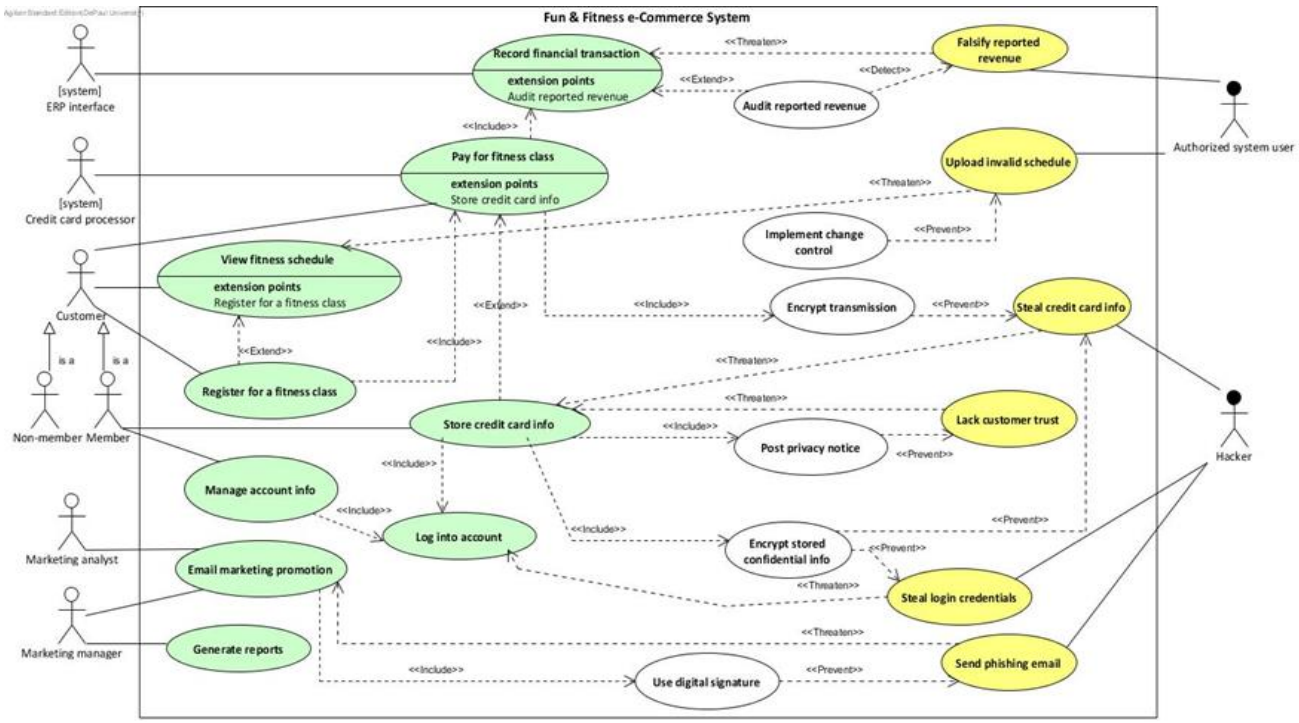

Figure 4. Misuse Case Diagram
Green = business objectives; Yellow = threats to business objectives; White = countermeasures

Construct a misuse case diagram for the *Fun & Fitness* e-Commerce system:

1. Review the use cases and actors in your original use case diagram.

2. For each use case, consider "what could go wrong" to prevent this use case from successful completion. These undesirable events are threats and are notated as misuse cases in your diagram. Given space constraints, include the more important misuse cases: those that have a reasonable likelihood of occurrence and would have a potentially significant impact. Color-code misuse cases in the model to distinguish them from use cases. Be sure to name each misuse case such that it explicitly and clearly indicates the threat. Start the name with a verb.

3. Identify and include in your diagram the mis-actor(s) that would interact with (e.g., initiate or perform) each misuse case. Draw associations between mis-actors and misuse cases. Color-code the mis-actors in the model to distinguish them from actors. One example is to apply a green background for use cases, a yellow background for misuse cases, and a white background for countermeasures (Spears 2012). Note that an actor may also be a mis-actor, so would be modeled twice in the diagram and color-coded accordingly. It may also improve readability to list actors separately from mis-actors. For example, in Figure 4, the actors are located on the left side of the diagram, while mis-actors are on the right. Secondly, the mis-actor "authorized system user" does not distinguish any particular internal role, but instead implies that anyone with legitimate system access could falsify reported revenue, intentionally or unintentionally.

4. Identify and include in your diagram "countermeasures" that would "prevent" or "detect" the misuse from occurring. These countermeasures are modeled as use cases and are to be included in the diagram in an organized manner. For example, the misuse case diagram in Figure 4 lists use cases on the left side, misuse cases on right, and countermeasures in the middle of the diagram. You may decide to organize your diagram differently; what is important is to clearly distinguish between functional use cases, countermeasure use cases, and misuse cases, as well as aid diagram readability by placing diagram elements in an organized manner.

5. Draw associations as applicable between functional use cases, countermeasure use cases, and misuse cases. Label each association line as appropriate to indicate "include", "extend", "threaten", "prevent", or "detect". For example, the countermeasure use case "audit reported revenue" is used to detect falsified reported revenue, and the "post privacy notice" use case is intended to prevent a lack of customer trust in relation to storing credit card data.

6. Evaluate the logical soundness of your diagram and descriptions. For example, does your diagram fit the context of the business case? Have major threats related to the case narrative been identified and modeled as misuse cases? Do

the threats relate to the use cases? Would the countermeasure logically prevent or detect the threat?

7. (optional) Document each misuse case by adapting a use case description to fit its misuse case. For example, in place of documenting a normal flow of procedures as one would for a use case, in a misuse case description, document a threat scenario and a description of how the countermeasure addresses the threat.

In introducing misuse case diagrams to students or system stakeholders, it may be helpful to construct a table to facilitate the discussion, and then transform the table contents into a diagram. As shown in Table 2, the first column of the table contains each use case in the initial use case diagram. For each use case, ask "What could go wrong to prevent this use case from executing successfully?" The answers become potential misuse cases. For each misuse case, ask what mis-actor might implement or perform the misuse case. Finally, for the fourth column, discuss prospective countermeasures for each misuse case. Constructing this table generates interesting discussion, including a variety of ideas. Students begin to get comfortable with their ability to identify security threats and countermeasures.

Use Case (Functional objectives)	**Misuse Case** (What could go wrong)	**Mis-actor** (By who)	**Counter-measure** (How to prevent or detect)
View fitness schedule			
Pay for fitness class			
Etc.			

Table 2. MAC (Misuse-actor-countermeasure) Table

In the misuse case diagram in Figure 4, six misuse cases were identified. As you can see, misuse case diagrams can get very busy. However, they reveal and communicate valuable information by highlighting IS security and privacy threats and countermeasures. Importantly, they prompt discussion among the systems development team and system users on system risk. Given space constraints, not all threats and countermeasures can be captured in a misuse case. Capture the most important elements and document remaining items from discussions with stakeholders.

10. STEP THREE: IDENTIFY THREATS WITH DATA FLOW DIAGRAMS

In this section, we illustrate an approach to threat and countermeasure identification that was developed by Microsoft using data flow diagrams (DFD). STRIDE is an acronym of information security threat categories (see Table 3): spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. STRIDE dictates that each element in a DFD is susceptible to some or all of these threat categories, as shown in Table 4. For example, an external entity (which STRIDE refers to as an external interactor) is susceptible to spoofing and non-repudiation (Hernan et al. 2006; Torr 2005).

STRIDE Threat Category	**Definition**
Spoofing	Pretending to be someone or something that one is not
Tampering	Making unauthorized changes to data at rest or in transit
Repudiation	Taking actions that cannot be traced back to the person that took them
Information Disclosure	Gaining access to data in transit or at rest that one is not meant to have access to
Denial of Service	Interrupting the normal, legitimate operations of a system
Elevation of Privilege	Gaining more system access privileges than intended, resulting in the ability to perform unauthorized actions

Table 3. Elements of STRIDE

In conducting a threat analysis using the STRIDE approach, the analyst creates a DFD of the system. Each DFD element can then be analyzed to identify potential threats and countermeasures. We used Microsoft's Security Development Lifecycle (SDL) Threat Modeling Tool[3] that is a free add-on to Microsoft Visio. The SDL Threat Modeling Tool is designed for threat analysis using the STRIDE technique and helps to facilitate the analysis. The SDL Threat Modeling Tool is designed for threat analysis using the STRIDE technique and will help facilitate the analysis. The tool provides guiding questions to help the analyst think about threats associated with a DFD element. If you are not using the SDL Threat Modeling Tool, use the information in Table 4 below to guide your analysis.

DFD Element	**S**	**T**	**R**	**I**	**D**	**E**
External Entity	x		x			
Process	x	x	x	x	x	x
Data Flow		x		x	x	
Data Store		x	x	x	x	

Table 4. DFD Elements and their STRIDE Susceptibilities (adapted from Hernan et al. 2006)

Perform a threat analysis with STRIDE in a DFD:

1. Re-read the *Fun & Fitness* case and use the information to create a DFD of the system.

2. Brainstorm the major types of threats that are associated with each element in the DFD (external entities, data flows, processes, data stores) using STRIDE.

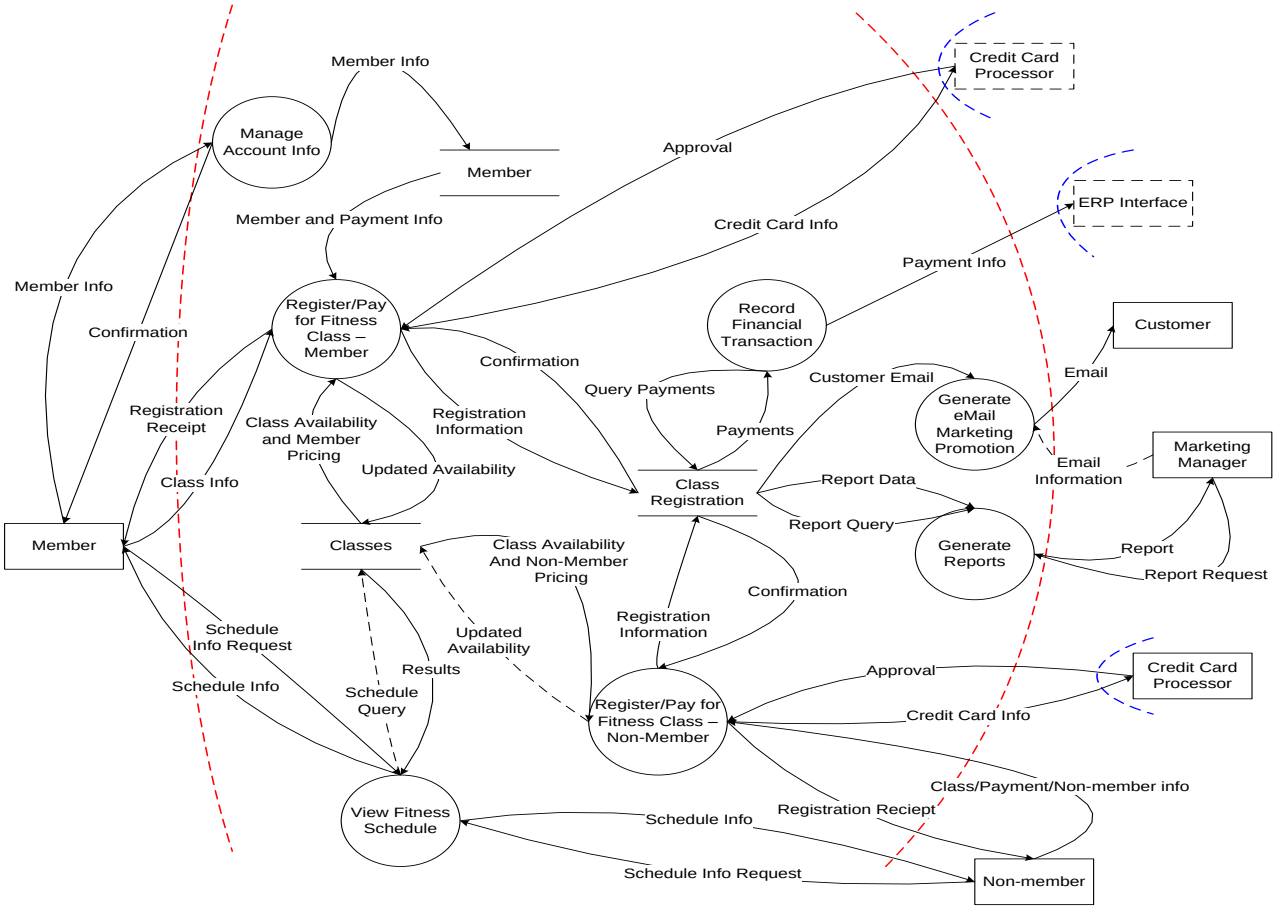

Figure 5: Level-0 DFD for Fun & Fitness E-Commerce System

3. Document threats and countermeasures for each DFD element using the SDL Threat Analysis add-on tool for Microsoft Visio. Alternatively, document in word processing software.

The DFD in Figure 5 depicts how data flows through the *Fun & Fitness* e-Commerce System. It shows us data input into the system from external entities; how data flows between system processes; data storage within the system; and the system's outputs.

Next, we illustrate a STRIDE analysis for a DFD process. For example, let's imagine that you are brainstorming threats for the "Register/Pay for Fitness Class" process. You might ask the following questions:

Spoofing

- Can someone register for a class without providing identification (name, email address, and payment) information?
- Can someone use a credit card without providing sufficient information (CVV2 code, billing address)?

Tampering

- Can someone falsify class availability information?
- Can the class pricing information be changed or overridden in the process?

Repudiation

- Are the registrations time-stamped?
- Are the payments time-stamped and verified?
- Are the details of the registration process logged in a log file?

Information Disclosure

- Is the credit card data encrypted in accordance with PCI DSS standards when it is being transmitted to the system?
- If someone starts to register for a class, but does not finish, does the process timeout, automatically log out the user, and redirect to a home screen after a period of inactivity?

Denial of Service

- Can someone key a SQL statement (SQL injection) into the registration form that could cause the system to crash?
- Can one person accidentally register for a class several times, thus denying access to others because the class seems to be full?

Elevation of Privilege

- Is it possible for users to perform administrative functions or access other users' records from this process?

Once a threat is identified, discuss an appropriate countermeasure with stakeholders for mitigating the threat. For example, ensuring that sensitive information (such as cardholder data) is encrypted will help to ensure that it is not disclosed to unauthorized parties; validating all inputs will help to defend against SQL injection attacks. These can either be logged in a table or in the SDL Threat Modeling Tool as shown in Figure 6.

11. STEP FOUR: DOCUMENT SECURITY REQUIREMENTS

Security requirements elicited from conceptual modeling should be documented as part of a general requirements document for the system that contains all of the system's requirements (Tondel et al. 2008). In doing so, security requirements will relate more clearly to functional requirements. Moreover, security requirements will gain visibility and be part of traceability and system testing efforts.

Table 5 illustrates an abbreviated requirements document. In general, the document template used for functional requirements will also be suitable for documenting security requirements. Notice that each requirement is ranked by priority, and an owner is assigned.

Requirement	Category	Priority	Owner
Post privacy notice where customer enters cardholder data	Privacy	High	Jane Doe
Use digital signature for email	Security	Medium	Joe Blog
Encrypt cardholder data	Security, Regulatory	High	Nancy Drew

Table 5. Systems Requirements Documentation with Security Measures Included

12. SUMMARY AND DISCUSSION

This teaching case has illustrated how conceptual models commonly constructed during systems analysis and design may be extended to include security considerations. In doing so, information classification, regulatory requirements, and security threats can be analyzed early in the SDLC so that more effective security and privacy countermeasures may be built into software and process workflows. Modeling techniques illustrated in this case also provide a graphical means to communicate IS security and privacy risk within a business process to stakeholders.

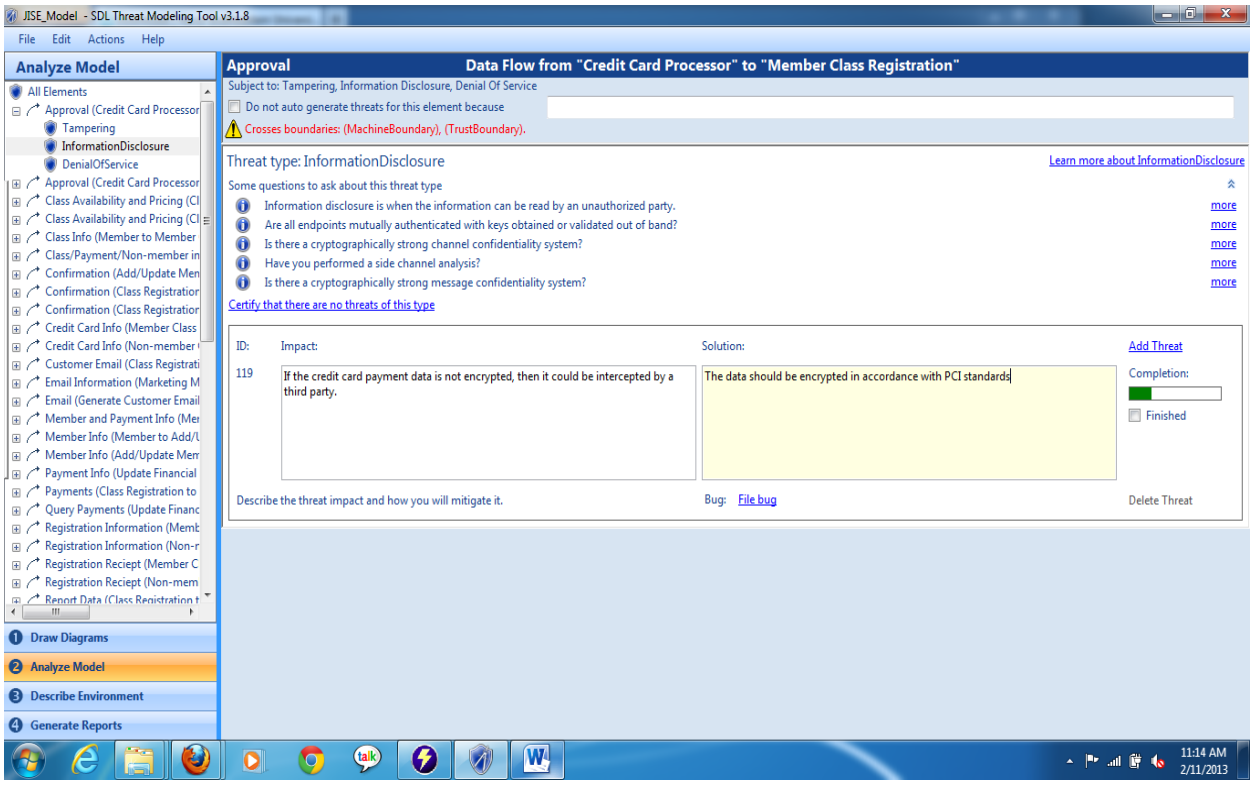

Figure 6. Using the SDL Threat Modeling Tool to Identify Threats and Countermeasures.

For additional discussion, consider the following questions:

1. How may security and privacy be integrated into other commonly used models, such as UML sequence, UML activity, or BPMN?

2. Can regulatory requirements and industry standards be relied upon for identifying confidential information assets? Why or why not?

3. Which stakeholders within an organization would you consult to gather information security and/or privacy requirements for an IS? Explain.

4. What security and privacy related requirements or design elements have you worked with, even if you didn't initially identify them as being security or privacy related?

ENDNOTES

[1] A portion and earlier version of this paper was presented at the 11th Security Conference in Las Vegas, NV in April 2012.

[2] Breaches included in the 2013 annual Verizon report are from actual external forensics investigations, with data contributed from 18 sources, including the U.S. Secret Service and other international law enforcement agencies.

[3] The SDL Threat Modeling Tool is available at http://www.microsoft.com/security/sdl/adopt/threatmodeling.aspx.

REFERENCES

Biros, D.P., Weiser, M., Barnes, S.D., and Romano Jr., N.C. 2007. "Incorporating Information Assurance in Systems and Analysis Design Curricula," The Americas Conference on Information Systems (AMCIS).

Culnan, M.J., and Carlin, T.J. 2009. "Online Privacy Practices in Higher Education: Making the Grade?" Communications of the ACM (52:2), Mar, pp 126-130.

Hernan, S., Lambert, S., Ostwald, T., and Shostack, A. 2006. "Uncover Security Design Flaws Using the STRIDE Approach." MSDN Magazine Retrieved May 31, 2013, from http://msdn.microsoft.com/en-us/magazine/cc163519.aspx

Hewlett-Packard Development Company, L.P. 2011. "2011 Top Cyber Security Risks Report." Retrieved May 31, 2013, from http://www.hpenterprisesecurity.com/collateral/report/2011FullYearCyberSecurityRisksReport.pdf

ISO/IEC 27002. 2005. "Information Technology - Security Techniques - Code of Practice for Information Security Management," ISO/IEC 27002:2005.

Mead, N.R., and McGraw, G. 2005. "A Portal for Software Security," IEEE Security & Privacy (3:4), pp 75-79.

Microsoft. 2010. "Simplified Implementation of the Microsoft Sdl." Retrieved May 31, 2013, from http://www.microsoft.com/security/sdl/resources/publications.aspx

NIST. 2008. "Special Publication 800-64 Security Considerations in the System Development Life Cycle." Retrieved May 31, 2013, from http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf

Parrish, J.L., Bailey, J.L., and Jensen, B.K. 2009. "Designing Security Information Systems: An Examination of the Treatment of Security in Systems Analysis and Design Textbooks," 8th Annual Security Conference, Las Vegas, NV.

PCI Security Standards Council. 2010. "PCI Data Security Standard." 2.0. Retrieved May 31, 2013, from https://www.pcisecuritystandards.org/security_standards/documents.php

Ponemon Institute. 2012a. "2011 Cost of Data Breach Study: United States." Retrieved May 31, 2013, from http://www.ponemon.org/local/upload/file/2011_US_CODB_FINAL_5.pdf

Ponemon Institute. 2012b. "2012 Application Security Gap Study: A Survey of IT Security & Developers." Retrieved May 31, 2013, from https://www.securityinnovation.com/uploads/Application%20Security%20Gap%20Report.pdf

PricewaterhouseCoopers LLP. 2013. "Global State of Information Security® Survey 2013." from www.pwc.com/giss2013

Salini, P., and Kanmani, S. 2013. "Model Oriented Security Requirements Engineering (Mosre) Framework for Web Applications," in: Advances in Computing and Information Technology, N. Meghanathan, D. Nagamalai and N. Chaki (eds.). Springer, pp. 341-353.

Sindre, G., and Opdahl, A.L. 2000. "Eliciting Security Requirements by Misuse Cases," Technology of Object-Oriented Languages and Systems, 2000. TOOLS-Pacific 2000. Proceedings. 37th International Conference on: IEEE, pp. 120-131.

Sindre, G., and Opdahl, A.L. 2008. "Misuse Cases for Identifying System Dependability Threats," Journal of Information Privacy & Security (4:2), pp 3-22.

Spears, J.L. 2012. "Conceptualizing Data Security Threats and Countermeasures in the E-Discovery Process with Misuse Cases," Americas Conference on Information Systems (AMCIS), Seattle, WA.

Spears, J.L., and Barki, H. 2010. "User Participation in IS Security," MIS Quarterly (34:3), Sep, pp 503-522.

Tondel, I.A., Jaatun, M.G., and Meland, P.H. 2008. "Security Requirements for the Rest of Us: A Survey," IEEE Software (25:1), Jan/Feb, pp 20-27.

Torr, P. 2005. "Demystifying the Threat-Modeling Process," IEEE Security & Privacy (3:5), Sep/Oct, pp 66-70.

U.S. FTC. 2007. "Fair Information Practice Principles," http://www.ftc.gov/reports/privacy3/fairinfo.shtm, U.S. Federal Trade Commission.

Uzunov, A.V., Fernandez, E.B., and Falkner, K. 2012. "Engineering Security into Distributed Systems: A Survey of Methodologies," Journal of Universal Computer Science (18:20), pp 2920-3006.

van Wyk, K.R., and McGraw, G. 2005. "Bridging the Gap between Software Development and Information Security," IEEE Security & Privacy (3:5), pp 75-79.

Verizon. 2013. "2013 Data Breach Investigations Report." Retrieved May 31, 2013, from www.verizonenterprise.com/DBIR/2013

AUTHOR BIOGRAPHIES

Janine L. Spears is an Assistant Professor at DePaul University's School of Computing where she teaches graduate and undergraduate courses in information security management, legal issues in information assurance, system analysis, and organizational modeling. Professor Spears' research focuses on regulatory and organizational issues in information security and consumer privacy, and on methodologies for integrating information security into information systems analysis and design.

James L. Parrish, Jr. is an Assistant Professor at Nova Southeastern University's Graduate School of Computer Science and Information Systems where he teaches graduate courses in systems analysis and design, enterprise information systems, cloud computing, and information systems management. Dr. Parrish's research focuses on information systems education, cloud computing, and social networking.

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096