A Database-driven Model for Risk Assessment

Research-in-Progress

J. Harold Pardue
University of South Alabama
hpardue@southalabama.edu

Shweta Purawat
University of South Alabama
spurawat@gmail.com

Jeffrey P. Landry
University of South Alabama
jlandry@southalabama.edu

Abstract

Securing valuable information assets is a problem that cuts across multiple industries and organizational missions. Using a database-driven approach to risk assessment, we developed a database model based on prior conceptual work in information security. Our approach is novel in that, as a risk assessment model, it emphasizes harnessing the advantages of a relational database approach to improve the efficiency and scope of risk assessment. Our method was proof of concept. We illustrated the database model's use, implementing it within a single domain—healthcare—and applied it to a single hypothetical scenario. The result was that the implemented database produced a resultant list of threats and countermeasures for the given health security scenario. Future work includes testing the approach with clients across multiple domains. The work has the potential to provide valuable risk-related, cross-domain benefits.

Keywords

Database, healthcare, risk assessment, information security.

Introduction

Information systems are widely used to support organizational missions and goals across industries, and are increasingly understood to be valuable assets requiring protection. With information systems (IS) impacting wide ranging industry domains, including banking, insurance, finance, manufacturing, automotive, telecommunications, government, and healthcare, information security should be seen as a common problem with industry and organization-specific aspects. Organizational information security management includes putting security controls in place to protect IS assets from financial loss, resource overutilization, legal liability, loss of reputation, and penalties (Guttman and Roback 1995). Effective selection and management of security controls to counter current and future threats is a critical to long-term organizational success.

Risk assessment plays an important role in the proper selection of security controls. Risk assessment as defined here involves identification of threats, vulnerabilities and assets and estimation of relative riskiness. Risk assessments also "delineate both the strategy to reduce the likelihood of a risk occurring (preventative measures) as well as the measures to respond effectively if a risk becomes a direct threat (reactive measures)" (Schou and Shoemaker 2007). Risk can be defined as the overall possible undesirable outcome due to deliberate or accidental exploitation of a vulnerability considering both the likelihood and the impact of the event. A vulnerability can be defined as a "flaw or weakness in system security procedures, design, implementation, or internal controls that could be exploited to accomplish a security breach or a violation of the system's security policy" (Stoneburner et al. 2002). A threat is a potential exploitation of a vulnerability by a threat-source against a threat asset. The combination of threat-asset and vulnerability constitute a unique threat action (Stoneburner et al. 2002). One approach to conducting risk assessment is to rank-order all the threats from most risky to least risky. The threat list facilitates risk assessment and risk management (Pardue 2009).

Our risk assessment model is based on relational database technology. Our approach posits it is possible to design a generic threat-database schema *a priori* and use it to derive a threat list specific to a given risk environment, and to perform risk assessment based on the derived threat list. A major advantage of this approach is that the database will evolve with each subsequent risk assessment. The database will refine and accumulate the collective knowledge and wisdom of the risk analysts. Furthermore, we submit that the model is applicable to a wide range of security domains, such as healthcare, voting, banking, insurance, etc.).

The purpose of this paper is to describe a database-driven model for risk assessment and to demonstrate the feasibility of this model. The next sections will provide a complete description of the model's database structure. The model will then be described as having three attributes: robustness to risk-related differences across industry domains, completeness from *a priori* entry of threat data, and adaptability for analyzing an evolving risk landscape. A proof of concept of its feasibility, using a healthcare threat domain scenario, follows. The proof of concept will demonstrate that it is possible to implement our conceptual data model, populate it with data, and produce results. A more complete description of our risk assessment process and its validation is beyond the scope of this work-in-progress paper. We conclude with a discussion of the future directions of this ongoing work.

Generic Threat Database

The threat database presented in this paper is based on relational technology. In the relational model, data are organized into tables (also known as relations), with rows and columns. A relation is an abstraction of a person, place, thing, event, or concept. It is this power of abstraction that enables the model to be applied to a wide range of domains. The structure of the database design abstracts and categorizes essential risk features of the risk assessment process. We used an entity relationship diagram (ERD) to logically represent these abstractions, or entities, and the relationships between and among entities. The database design reduces cognitive complexity while leveraging query technologies to reduce the complexity of the risk assessment search space. Namely, the risk analyst need only assess the threats produced by a query against the database. See section later in this paper for a description of query capabilities.

The beginning point for construction of the database design was identification of essential elements for information security risk assessment. These elements, which comprise the entities or relations, of the database model are THREAT, ASSET, VULNERABILITY, and CONTROL. These essential elements were derived from prior work (Hoffman et al. 1978, Whitman 2003). Hoffman et al.'s SECURATE paper was an early theoretical description of essential security entities. They described security threats, objects (assets), and features (controls), defining a security system as a set of objects, each with a loss value; a set of threats, each with a likelihood, and a set of features, each with a resistance. Whitman advocated performing risk assessment by prioritizing threats and assets along axis of a grid. Starting with the upper left corner cell that matched the highest order threat- asset pairing at an intersecting cell, the risk analyst would consider whether that pairing represented an exposure of a particular asset to a specific threat—in other words, a vulnerability. The analyst would then document one or more controls selected in each cell for each threat-asset pair with an identified vulnerability. Thus, T-V-A analysis became a simplistic form of vulnerability assessment. This triad, perhaps better named for our purposes as a T-V-A-C quadruple, forms the core structure for the entire database design. See Figure 1.

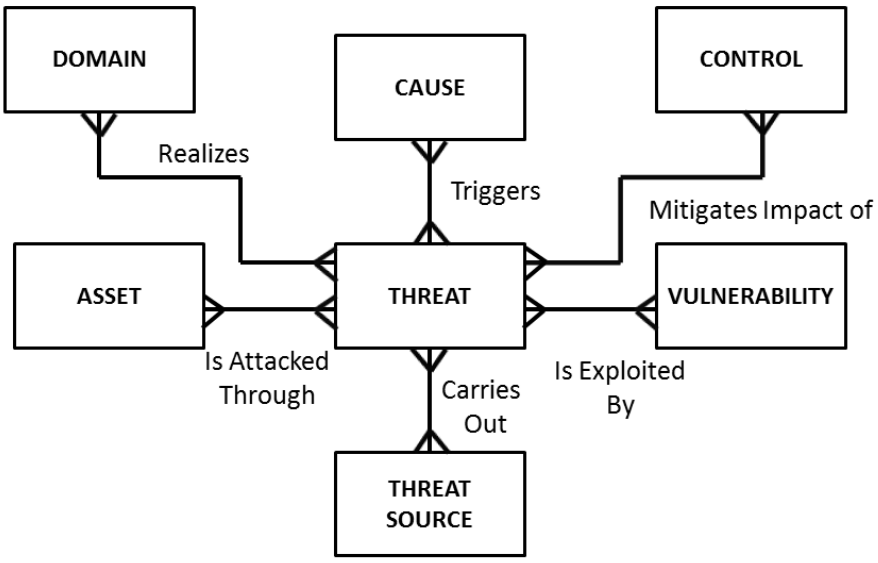

Figure 1. Entity-Relationship Diagram of Threat Model

In addition to T, V, A, and C, our model includes two other entities, THREAT SOURCE, CAUSE, and DOMAIN. These abstractions are derived from our definition of a threat as the potential exploitation of a vulnerability by a threat source against an asset through either deliberate or situational causation. Threats can be countered by security controls. Threats are realized in a specific domain such as healthcare, banking, or financial services.

The following section describes each abstraction, shown in the ERD in Figure 1, separately.

Threat

A threat is either an exploitation or accidental triggering of a vulnerability by a threat source against an asset. In the ERD, we model the event or circumstance that can result in adverse impact on an organization as the entity "THREAT".

Similar to threat sources, threats should be defined in terms that are specific to a domain but generic in form. Ideally, a threat should be written as a verb expression. For example, in the healthcare domain, threats to data can be generically defined in terms of verbs such as disclose, delete, manipulate, and gain unauthorized access to patient health information.

A threat source either intentionally exploits or accidentally triggers a vulnerability against an asset through intent/situation and method that brings about the threat action. A threat can be executed in many ways. Each high-level threat, for example, "disclose critical data or information" or "manipulate critical data or information", can be executed through a number of different methods. Each method or action that brings about adverse impact is termed a "threat action." To the degree possible, the instances of threat action should be specific to the threat domain but generic in form. Example threat actions would be "inject malware into EHR system" or "inject SQL expression into login text for EHR system." Specific malware or SQL injection attacks need not be enumerated. Both threat actions can lead to the threat "disclose critical data or information."

Vulnerability

Vulnerability is another essential component in the threat-vulnerability-asset-control quadruple. According to NIST 800-30, a vulnerability is a flaw or weakness in system security procedures, design, implementation, or internal controls that could be exercised (accidentally triggered or intentionally exploited) and result in a security breach or a violation of the system's security policy (Stoneburner et al.

2002). In our database model, a weakness in a system security procedure, design, implementation, or control is modeled as the entity "VULNERABILITY." Some examples of vulnerabilities are weak antivirus software, poor input validation technique, and computer screens exposed to others. A single threat may be able to exercise more than one vulnerability in a system, and vice-versa. Thus, our design depicts a many-to-many relationship between the entities "THREAT" and "VULNERABILITY."

Asset

Threats to a system are carried out against assets within an organization that result in adverse impact. In general, assets include data, images, all types of hardware, software, and networks found in the domain of the security environment, and also people and procedures. This collection of components is modeled by the entity "ASSET." Assets should be stated as generic types such as wireless routers, desktop computers, or ftp clients rather than vendor specific assets. In the healthcare domain, examples of assets could be a patient's electronic health record or a patient's MRI.

The exploitation/triggering of a vulnerability will likely target a combination of assets. For example, in the healthcare domain, a threat source may execute a SQL injection attack by exploiting a vulnerability associated with an electronic health record exposed through a web browser CPOE (Computerized Physician Order Entry). Therefore, the entity "ASSET" is linked to the entity "THREAT" through a many-to- many relationship.

Control

No security environment can be risk free. The best an organization can do is implement controls that mitigate the risk of a threat to an acceptable level. Controls are employed as countermeasures designed to restrict, monitor, and protect assets against a threat, thereby minimizing the possibility of a threat exercising vulnerability (Stoneburner et al. 2002). Some examples of controls are strong security software integrated anti-virus, anti-spyware, firewall, anti-spam, anti-phishing, backup technologies, intrusion prevention (IPS), and sheltered computer screen.

The control or countermeasures are modeled as the entity "CONTROL" in the ERD. Controls or counter measures are intended to counter actions taken by a potential threat source. A control can counter one or more threat actions. For example, a firewall can prevent threat actions such as unauthorized access, gateway defense, hiding and protecting internal network addresses. Likewise, a threat action can be countered by one or more controls. For example a SQL injection attack can be countered by both intrusion prevention and a firewall. The many-to-many relationship exists between the entity "CONTROL" and the entity "THREAT."

Threat Source

A threat is a potential exercise of a vulnerability by a threat source against an asset. In the ERD, we model the agent who either deliberately or accidently exercises a vulnerability as the entity "THREAT SOURCE". The entity THREAT SOURCE is a classification of threat sources rather than specific instances of threat sources. Sample threat sources could be "human-unintentional insider", "human-deliberate insider", or "human-deliberate outsider" rather than specific instances such as "bank clerk", "claims adjuster", or "physician". The generic classification of threat sources provides guidance for identifying specific threat sources in a particular risk environment during the risk assessment and planning process. Attempts to encode all possible instances of threat sources negate the power of abstraction gained by the use of a database model. Details for a specific threatsource are given in the scenario attribute as an example.

Cause

The Cause entity models the motivational and situational factors associated with threat sources, recognizing that threat sources are both human and non-human, and both deliberate and unintentional in the case of human sources. A vulnerability, therefore, may be intentionally exploited or accidentally triggered. Deliberate causes assume a rational man model for threat sources, meaning an attacker will act as a threat source against an asset to accomplish a specific goal. All things being equal, a deliberate threat source will strive to maximize the impact of exploiting a vulnerability while minimizing the cost of

carrying out the threat action. Examples of costs are money, time, expertise required to attain needed skills, and the risk of detection (Jones 2005). Human-unintentional threat sources include people making mistakes, and non-human causes include technical, natural and environmental causes. Therefore, in conducting a risk assessment and planning for risk management, it is important to document the cause of a threat. Understanding the cause of a threat aids in deciding which of many possible controls to implement. Causes should be described in terms of high level outcomes. For example, in the healthcare domain, a deliberate threat source could carry out a threat action for a threat for "economic gain" (blackmail, unauthorized benefits) or "non-economic gain" (revenge, curiosity) or both, while human-unintentional sources might bring about threat actions due to stress, fatigue, or poor training. Technical, environmental and natural causes of threats might include software defects, power outages and weather disasters.

Threat sources determine threat actions through causation. A threat source can result in a threat action that may be associated with multiple causes. In like measure, a cause can provide the impetus for multiple threats. Therefore, the relationship between the entity THREAT and the entity CAUSE is modeled as many-to-many.

Domain

Finally, we posit that the database design presented here is sufficiently abstracted to be applicable to a wide range of domains. The entity "DOMAIN" models the context for the security environment. In this paper, the healthcare domain is used to illustrate our database-driven risk assessment model. However, our database driven model is applicable to other domains as well.

After describing different entities in the ERD, the following section explains our database based risk assessment model.

Database-driven Risk Assessment Model

The Entity Relationship Diagram (ERD) depicted in Figure 1 is a logical model of all the entities and the relationships between the entities in the risk assessment process: THREAT, VULNERABILITY, ASSET, CONTROL, THREAT SOURCE, CAUSE, and DOMAIN. Given this data model and the relationships therein,

> *we propose to construct a database of threats and populate all the tables with wide-ranging threat data for each domain,* a priori. *We can then use this generic database such that given a set of assets, vulnerabilities, and controls, a risk analyst can derive a set of threats appropriate to a given security context by querying the database.*

This is our database-driven risk assessment model that states that we can derive a threat list specific to a security context from the generic threat database constructed ahead of the time.

The ERD is our logical model used to drive the implementation of the physical database used for risk assessment. The following section briefly describes how we implemented the database from the ERD.

Database Schema

The ERD in Figure 1 is implemented as a database schema depicted in Figure 2. To construct database schema, the conceptual model (ERD) is transformed into an equivalent set of related normalized tables (also known as relations). A normalized or a well-structured relation is defined as a relation that contains minimum data redundancy and allows users to perform data manipulation operations without data inconsistencies.

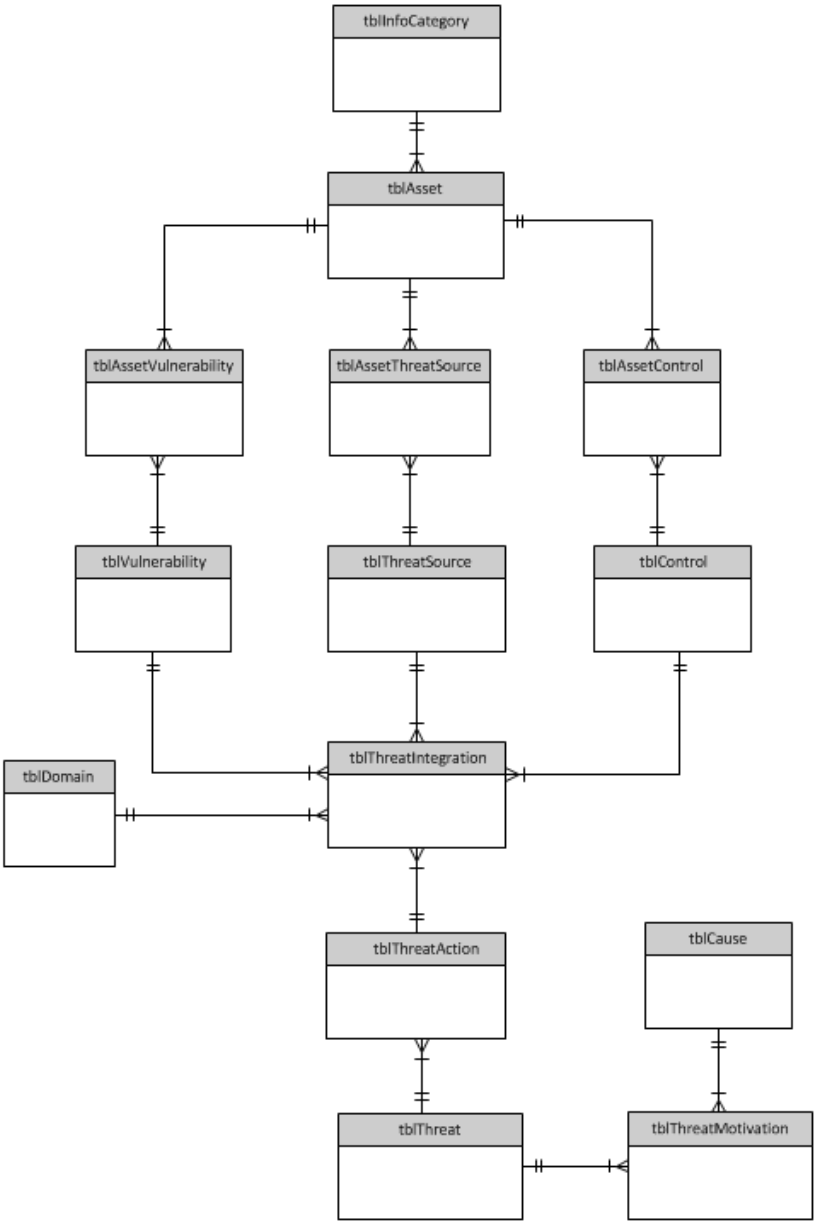

Figure 2. Database Schema

The other important element of the ERD to database schema transformation is to resolve many-to-many (M:N) relationships between the entities. The M:N relationships between two entities are resolved by creating a separate relation. The new relation created is an associative table. The primary key of the new associative table is combination of attributes that take their values from primary keys of the original entities in the M:N relationship.

The rest of the section will describe the tables equivalent to the entities in the ERD, and also will discuss all the associative tables introduced to resolve M:N relationships between the entities in the ERD.

Tables in the Database Schema

The entities THREAT, VULNERABILITY, ASSET, CONTROL, THREAT SOURCE, CAUSE, and DOMAIN are implemented by the tables “tblThreat”, “tblVulnerability”, etc. This many-to-many relationship

between the table "tblThreat" and the table "tblCause" is resolved by the associative table "tblThreatMotivation". The one-to-many relationship that exists between the tables "tblThreat" and "tblThreatAction," is modeled with the foreign key "ThreatID" in the table "tblThreatAction." The many-to-many relationship between the tables "tblAsset" and "tblVulnerability" is resolved through the associative table "tblAssetVulnerability." The many-to-many relationship between control and asset is resolved through the table "tblAssetControl." The many-to-many relationship between the tables "tblAsset" and "tblThreatSource" is resolved with the table "tblAssetThreatSource." Finally, the many-to-many relationships among the tables tblThreatAction, tblVulnerability, tblControl, tblThreatSource, and tblDomain are resolved through the table "tblThreatIntegration." The table "tblThreatIntegration" documents a single combination of threat action, vulnerability, threat source, and control.

The following section briefly describes how we developed the healthcare domain threat database.

Populating the Threat Database

We populated our database with threat data for the healthcare domain in a two-phase process. The first phase in constructing a domain specific threat database is cataloging Threat-Vulnerability-Asset (TVA) triads, using Whitman's model. The TVA cataloging involves threat and asset identification, and vulnerability analysis. In phase two, we extended TVA triads to include related causes, threat sources, threat actions, and controls. The result was essentially a threat matrix, each row of which includes the seven attributes plus a threat scenario that illustrates an actual or hypothetical instance of the threat. For each TVA triad cataloged in the first step, we identified and listed all the associated causes, threat sources, threat actions, and controls.

We identified and listed TVA triads and extended the triads based on a literature review, previous work of the authors and other sources. The specific sources used to create the threat matrix include: literature (Stoneburner et al. 2002, Kahn et al. 2008, Samy et al. 2010, Nematzadeh et al. 201, Kotz et al. 2011), previous work of the authors (Landry et al. 2011, Pardue et al. 2011) and careful understanding the threat domains through field studies, attending conferences on security and privacy, the NIST Computer Security Resource Center (csrc.nist.gov), and expert knowledge.

A row in the threat matrix is transposed and replicated in Table 1. Each row in the threat matrix corresponds to one row in the table "tblThreatIntegration." We populated tables "tblCause," "tblThreatSource," "tblThreat," "tblThreatAction," "tblVulnerability," "tblControl," and "tblAsset" with distinct values of the attributes of the threat matrix Cause, Threat Source, Threat, Threat Action, Vulnerability, Control, and Asset, respectively.

The table "tblThreatIntegration" attributes take their values from primary keys of the "tblThreatSource," "tblThreatAction," "tblVulnerability," "tblControl," and "tblDomain."

Healthcare TVA matrix and Scenario

The table below describes a hypothetical threat scenario.

Threat Attribute	Value
Cause	Non-economic gain – revenge – blackmail
Threat Source	Human-deliberate insider
Threat Action	Disclose health information by extracting data through USB drive/CD.
Asset	Patient health records, patient personal identification information
Vulnerability	Accessible removable media ports
Controls	Disable removable media ports. If ports are enabled the computer system must be in secure area with authorized entry. Strong antivirus and malware protection.
Threat Scenario(s)	Threat scenario: human-deliberate insider - In this scenario, the threat-source is a nurse seeking revenge post-termination. The nurse has insider knowledge that the removable media ports (USB/ CD) on computers are enabled in the hospital facility and systems in which critical information about patients are stored. He visits the hospital under the pretense of retrieving personal items left behind but actually seeks revenge by disclosing patient information in order to damage the hospital's reputation and create legal exposure. While momentarily left alone in a triage area, the former nurse inserts a USB flash drive into an accessible computer and extracts patient health records.

Table 1. Health Care Threat Profile

Proof of Concept

The fundamental premise of the database-driven methodology presented here is that it is possible to construct a generic relational database for the risk assessment process and populate the database with T-V-A-C risk data for different threat domains. To conduct risk assessment for a client with this database model, we would identify assets, vulnerabilities and controls in the client's current security context. A query against the threat database would result in a list of threats specific to the client.

Although many approaches to assessing risk exist, the approach proposed here is to produce a rank-ordered list of threats (Pardue et al. 2009). Rank-ordering allows the analyst to estimate the relative riskiness of a threat, conduct what-if analysis, estimate residual risk, and prioritize resource allocations for implementing controls. The ability to rank risks in our database is a work-in-progress. When this capability becomes available and a suitable client is identified, further validation of our risk assessment method will be possible. For now, we will demonstrate the concept of our risk assessment model by illustrating with a hypothetical scenario. See the healthcare scenario profiled in Table 1.

Asset, Vulnerability and Control Identification

An analysis of the existing security state of the hypothetical health care facility depicted in Table 1 was conducted to identify assets, vulnerabilities and controls. The following list represents a subset of the assets identified.

- Personal computer unit (laptop, desktop)
- Application servers, database server, files, hard-drives, hubs, servers
- Patient health records
- Patient identity information
- Operating system
- Routers
- Smart phones/ devices

The following vulnerabilities were identified.

- Accessible removable media ports
- Computer screens exposed to other patients, insider and outsiders
- Unattended computers
- Unlocked screens

The following controls were in place.

- Anti-Sniffing Tools, Antivirus protection, Security patches
- Authentication by swiping thumb
- Regular file back-up
- Employee education to be careful when engaging in peer-to-peer (P2P) file-sharing (Avoid downloading files with the extensions .exe, .scr, .lnk, .bat, .vbs, .dll, .bin, and .cmd)
- Employee education to be certain a web site is legitimate before you go there
- Monitoring database traffic and compares it to expectations
- Doctors uniform
- Download the latest version of browsers
- Educating medical staff about severity, consequences, fines and penalties related to data breach
- Employee photoID badges and wrist bands
- Enable automatic Windows® updates
- Encrpyted data, Firewall enabled, IAR (Internet Alert Registry)
- Implementing the principle of least privilege
- Intrusion prevention systems (IPS) by inspecting database traffic
- Managed by suitable security architectures, Multilevel authentication - Strong Password
- No cameras allowed within premises
- No open LCD display of patient information, Packet filtering
- PatientID is used wherever necessary, Hijack alert system, Query-level access control
- Network protocols and services that do not rely on the IP source address for authentication
- Routing Intelligence, Sheltered or private registration desk and triage room
- Software updates, Strong authentication mechanisms, Strong encryption key
- Strong mutual authentication
- Strong security software integrated anti-virus, anti-spyware, firewall, anti-spam, anti-phishing, and backup technologies
- Switched networks, Use encrypted Wi-Fi wireless access point
- Use extreme caution when opening attachments
- Use security precautions for your PDA, cell phone, and Wi-Fi devices
- Use Internet service provider (ISP) that implements strong anti-spam and anti-phishing procedures

- Written warnings

With the assets, threat sources, vulnerabilities and controls identified for the threat scenario, we constructed an SQL query and derived a threat list relevant to the hypothetical security context under examination.

Query Structure and Result

The query technology used with our methodology is Structure Query Language (SQL). SQL is used to access and manipulate relational data. The major data operations are Create, Read, Update and Delete (CRUD). CRUD operations are executed against the database to maintain and modify threat data and to produce threat lists for risk assessment based on inputs from risk analysts. The actual query used to produce a list of threats was very complex. A stylized, generic version is provided below to illustrate the basic query structure.

```
SELECT tblThreat.Threat, tblThreatAction.ThreatAction --other attributes…
FROM   tblThreatIntegration, tblThreat, tblThreatAction, tblAsset,
       tblVulnerability, tblThreatSource --other tables
WHERE  tblThreatIntegration.ThreatID=tblThreat.ThreatID AND
       tblThreatIntegration.ThreatActionID=tblThreatAction.ThreatActionID AND
       --other PK=FK comparisons AND
       AssetID IN(list of FKs) AND
       VulnerabilityID IN(list of FKs) AND
       ThreatActionID NOT IN (SELECT ThreatActionID
                              FROM tblControl
                              WHERE ControlID IN (list of FKs))
```

The list of threats and threat actions returned by the query for the hypothetical scenario are reproduced below in Figure 3. The column "Threat" contains duplicate data values because Threat has a 1:m relationship with ThreatAction, that is, there are multiple possible threat actions to execute a given threat.

	Threat	ThreatAction
1	Disclose Patient health information	Disclose health information by extracting data through USB drive/CD
2	Disclose Patient health information	Disclose health information by Malware Injection through enabled USB/CD port
3	Disclose Patient health information	Disclose Patient Information by collecting information from unattended computers/ unlocked computer screens
4	Disclose Patient health information	Disclose Patient Information by looking over computer screen in unauthorized way(looking over the shoulder)
5	Manipulate Patient health information	Manipulate health information by extracting data through USB drive/CD
6	Manipulate Patient health information	Manipulate health information by Malware Injection through enabled USB/CD port

Table 3. Query Result – a Threat List

A second query was run to identify controls or counter measures the client should consider to mitigate their exposure to these threats. The controls identified were:

- Disable removable media ports
- If ports are enabled the computer system must be in secure area
- Automatic screen lock
- Burglar alarm and cameras in storage area
- Sheltered computer screen

The hypothetical illustration served as a proof of concept for our database-driven risk assessment model. It illustrated the conceptual data model's implementation as a database schema, and the population of that database with (healthcare) domain-specific T-V-A-C risk data. Further evidence demonstrated that the database could be queried to produce relevant risk assessment results. Two types of results were reported: a threat list, and possible controls to counter the unmitigated threats. As a proof of concept, this illustration demonstrated, therefore, that our database, in prototype form, is capable of being used. With the addition of the capabilities for estimating risk, and when tested with clients, our work will be more suitably validated, and a contribution to the practice of risk assessment is possible.

Future Work

In this paper we purposed a database driven risk assessment model. We demonstrated our risk assessment model by generating a threat list based on a hypothetical hospital scenario. Future research will involve an actual case study of a health care facility and generation of a threat list and risk assessment report.

The research can be extended to development of a Web-based risk assessment application. The software application would enable a client to enter data on their assets, vulnerabilities, and controls. The resulting list of threats and additional controls would provide the basis for a risk assessment plan and reports.

We plan to expand our database with threat data in other domains, such as banking, e-commerce, insurance, manufacturing, and voting. The threat database will evolve with each subsequent risk assessment study. In this way, the database will refine and accumulate the collective knowledge and expertise of risk analysts. Intelligence can be added to the database functionality through the development and implementation of data mining and machine learning algorithms. The knowledge discovered in one domain could be applied to other domains and also to perform predictive analysis of the threats for clients.

REFERENCES

Guttman, B. and Roback, E. A. 1995. *An Introduction to Computer Security: The NIST Handbook*, Darby, PA: DIANE Publishing.

Hoffman, L. J., Michelman, E., and Clements, D. P. 1978. "SECURATE—Security Evaluation and Analysis using Fuzzy Metrics," in *AFIPS National Computer Conference Proceedings* 47, Arlington, Va., pp. 531-540.

Kahn, S. and Sheshadri, V. 2008. "Medical Record Privacy and Security in a Digital Environment", *IT Pro*, IEEE Computer Society, March/April 2008, pp. 46-52.

Kotz, D. 2011. "A threat taxonomy for mHealth privacy", in *Workshop on Networked Healthcare Technology (NetHealth)*, Bangalore, India.

Jones, D. W. 2005. "Threats to voting systems," Position paper for the NIST workshop on Threats to Voting Systems, Gaithersburg, MD, October 2005.

Landry, J., Pardue, H., Johnsten, T., Campbell, M., and Patidar, P. 2011. "A Threat Tree for Health Information Security and Privacy," *Americas Conference on Information Systems (AMCIS 2011)*, Detroit, MI.

Nematzadeh, A. and Camp, L. J. 2010. "Threat analysis of online health information system," in *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '10)*, Makedon, F., Maglogiannis, I., and Kapidakis, S. (eds.), Samos, Greece, 31, pp. 1-7.

Pardue, H., Landry, J., and Yasinsac, A. 2011. "E-Voting Risk Assessment: A Threat Tree for Direct Recording Electronic Systems," *International Journal of Information Security and Privacy (*5:3), pp. 19-35.

Pardue, H., Landry, J., and Yasinsac, A. 2009. "A Risk Assessment Model for Voting Systems Using Threat Trees and Monte Carlo Simulation." In *First International Workshop on Requirements Engineering for e-Voting Systems (RE-VOTE)*, Atlanta, GA, pp. 55 –60.

Samy, G. N., Ahmad, R., and Ismail, Z. 2010. "Security Threats Categories In Healthcare Information Systems", *Health Informatics Journal* (16:3), pp. 201-209.

Schou, C. and Shoemaker, D. 2007. *Information Assurance for the Enterprise: A Roadmap to Information Security,* New York, NY: McGraw-Hill Irwin.

Stoneburner, G., Goguen, A., and Feringa, A. 2002. *Risk management guide for information technology systems: Recommendations of the National Institute of Standards and Technology*, Gaithersburg, Md: U.S. Dept. of Commerce, National Institute of Standards and Technology, http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf.

Whitman, M. 2003. "Enemy at The Gate: Threats to Information Security," *Communications of the ACM* (46:8), pp. 91-95.