Proceedings of the 53rd Hawaii International Conference on System Sciences | 2020

Creating a Standardized Risk Assessment Framework Library for Healthcare Information Technology

Suzanna Schmeelk
St. John's University
schmeels@stjohns.edu

Abstract

Data breaches are occurring at an unprecedented rate. In February 2019 alone, over a million individuals were reported to the United States government as having been involved in a breach of their medical data by healthcare entities. Although many organizations have some policies, procedures and risk management components in place, few (if any) organizations are centrally connecting legal requirements, penetration tests, policies and procedures into a standardized and consistent methodology for further analysis and auditing. This research produces a new open source risk management standardized library coordinating the aforementioned risk management components. The new library is applied to an open source vulnerable web-application example to emphasize the benefits from the adoption of such a public standardized risk assessment library.

1. Introduction

In the United States of America, medical entities are covered under federal laws to protect patient information[1]. Specifically, *Health Insurance Portability and Accountability Act* (HIPAA) and *Health Information Technology for Economic and Clinical Health Act* (HITECH) [2] are regulations at the federal-level to protect the privacy and security of patient health information such as name, birthdate, social security numbers, medical record numbers, etc. This specific digital patient information is considered electronic patient health information (ePHI). Medical entities may also be under other legal requirements such as non-disclosure or confidentiality requirements of other data (e.g. research, employee, drug, etc.).

Since many covered entities are siloed where different components of the organizational risk (e.g. legal, budget, security, privacy, technology, etc.) are being managed by different department entities without a standardized and well-connected system, organizations deal with frustrations both when needing to produce detailed and accurate audit records and when communicating risks to the business. For example, an exception to a policy may result in unidentified organizational risk if these components are not consistently coordinated and periodically reviewed/updated.

The research that is described contributes a standardized risk assessment library model and, then, provides an example use case where a vulnerable web application risk assessment findings are connect with the developed standardized library. This connection enables the organization to report on its risks and maintain internal statistics as related to technical limitations, administrative limitations, organizational policy exceptions and federal legal requirements to inform the business, auditors and business-associates on the risks involved if the organization adopts the vulnerable web application into its business processes.

2. Risk Assessment Standards

The National Institute of Standards and Technologies (NIST) has produces many Special Publications on Risk Assessments [3]. In fact, many organizations around the world are following the NIST Risk Assessment frameworks. In addition to developing a standardized framework, NIST and MITRE have worked tirelessly to produce a standardized attack/malware (e.g. Common Attack Pattern Enumeration and Classification (CAPEC) [4] and National Vulnerability Database (NVD) [5]) and vulnerability dictionaries (e.g. Common Weakness Enumeration (CWE) [6] and Bug Framework (BF) [7]). These standardized dictionaries are agnostic to industries. They have been developed to encourage a standardize languages for software faults since a standardized language promotes software development and software assurance tool discussions.

To date the industry has worked tirelessly to standardize language on software bugs since software vulnerabilities and malware has been around for decades. Recently, however, there has now become an industry need on the standardization of actual sub-component findings for assessing risk. For example,

URI: https://hdl.handle.net/10125/64216
978-0-9981331-3-3

two risk assessments for identical systems from different organizations may result in entirely different risk discovery. Furthermore, the language applied into the assessment by the different organizations may be unique to each organization. As such, a gap exists into the research and literature on standardizing the risk assessment findings. This gap is mainly due to how recent risk assessments have become pertinent to organizational survival. For example, Facebook announced that in March 2019 it appropriated three-billion dollars to pay fines related to federal privacy regulation breaches [8].

2. Risk Assessment Literature Review

Related risk assessment literature involves the automation of risk assessments and education of risk assessments. However, currently a research gap exists on the development of a standardized risk assessment library, which includes all risk components when developing findings.

2.1. Risk Assessment Automation

Risk assessment automation have been proposed in the form of automated penetration testing frameworks (e.g. [9], [10], [11], [12], [13], [14], [15], [16], [17], and [18]). These automated tools are excellent resources for identifying vulnerabilities. The focus of these automated frameworks are specific; but they are not focused on the importance of other risk assessment components, such as legal medical requirements/regulations. Moreover, these frameworks are not focused on a standardization of language for cross-communication, cross-collaboration, auditing and legal-prosecution. Conforming to a specific language improves overall process reporting and analysis.

2.2. Risk Assessment Education

Research on the education of risk assessments has primarily focused on the learning of penetration testing techniques (e.g. [19]). These education curriculums are great learning resources. They do not, however, consider over-arching organizational risk, nor are they specific to the medical industry legal requirements. This paper fills this literature gap suggesting that penetration test findings should be carefully crafted to direct link with policies, laws, and other risk assessment components.

2.3. Risk Assessment Standards

Risk assessment standards literature is disjoint. Regulatory requirements, such as HIPAA and PCI, require risk assessment components. Risk standardizing bodies and framework supporters such as NIST, Fair, ICS^2, work to improve the overall risk assessment process. Risk Management research is typically focused on introducing a new tool and can be geared towards a specific industry. Table 1 below summarizes the different risk standards.

Table 1 Risk Standards Summary

Risk Standards	***Examples***
Regulatory	HIPAA, PCI, SOC, SOX
Industry Best Practice Models	NIST, SANS Guidance, Fair, ISC^2
Research	Tool and industry specific

3. Risk Assessment Model

There are many risk assessment models at large. Two of the predominate organizations are The Faire Institute and The National Institute of Standards (NIST). In addition, risk assessments are typically quantitative, qualitative or a hybrid of the two. NIST has one of the most popular and widespread standardized risk assessment frameworks in practice. Of all the industry best practice models, nothing at large is yet advocating for a standardize risk assessment library (e.g. a findings library) to enable cross-organizational, cross-system and cross-application analysis.

4. Risk Assessment Library Considerations

Managing the risk in a medical setting is unique to the medical setting due to specific regulations. As such, standard risk assessments such a penetration test from an outside organization may not properly report on the risk from the organizational level. In addition, many medical facilities employ a ticketing system between siloed departments where connection between inter-department components is not identified in a standardized or repeatable format.

The following five sub-sections identify organizational components, which should be connected in a standardized public risk assessment language dictionary to inform on organizational risk in a repeatable format (shown below in Table 2): legal requirements, training requirements, vendor

requirements, web application security requirements and organizational controls. A standardized risk finding library encourages cross-organizational collaboration, communication, auditing and legal consistency if/when cases ultimately end-up in court.

Table 2 Risk Component Examples Requiring Standardized Language

Risk Component	***Example***
Legal	HIPAA, PCI, SOX
Training	Specific requirements in legislation
Vendor	Business Associate Agreements
Web Application	Penetration Test Results
Organizational Controls	Technical, Physical, Budget, Administrative

4.1. Legal Requirements

Medical entities have different federal requirements in the United States of America from other data-driven organizations. Specifically, medical covered entities under HIPAA/HITECH are subject to audits by the United States Health and Human Services Office of Civil Rights (US HHS OCR). In addition, organizational breaches of patient electronic health information of over 500 individuals must be reported to the OCR as ruled in HITCH and subject to federal fines.

HIPAA also has specific mandates for electronic health data requirements, which should be consistently mapped during a risk assessment to appropriately manage organizational risk.

Beyond the requirements of HIPAA/HITECH, medical entities may be under other legal requirements at the state, city or other contractual obligations, which also should be consistently mapped during a risk assessment.

4.2. Training Requirements

Training requirements may be requirements at the vendor-level, federal, state, or city requirements. For example, the protection of credit card data under Payment Card Industry (PCI) compliance, expects software developers to be properly trained. In addition, state labor laws and federal laws such as HIPAA also have specific training requirements. If an organization or their accepted vendors are missing any of these training requirements, the organization may be finically liable.

4.3. Vendor Requirements

Vendors have different potential requirements, which must be in place, if/when a healthcare entity decides to work with the vendor. Specifically, vendors managing a covered entities patient data traditionally needs to have a business associate agreement in place for federal requirements. Other federal or organizational requirements may be for annual vendor system/application penetration tests and malware incident responses plans.

4.4. Application and System Requirements

Application and system security are typically measured through their own risk assessments, tests and potentially source code auditing. (Note there may be legal obligations to assessments, e.g. penetration test, which need to be in place prior to an assessment.) Typically, technical teams identify technical issues with applications and systems; however, they may not have correct use cases, legal and budget information at their disposal. Once they perform a risk assessment, they may upload the paper document into an Integrated Risk Management (IRM) system; and, then, may even forget about the updating/re-assessing the collected risks. In such cases, the risk assessment is more an impression rather than an informed reproducible science informing on the true likelihood and impact. The following sub-sections identify eight standard sub-categories (visualized in the Table 3 below) employed during a risk penetration-assessment to report on the risk.

Table 3: Penetration and System Analysis Findings

Application and System Risk Domains	***Example findings***
Authentication	Missing two-factor
Session Management	No session timeout
Data-in-Motion	Lack of TLS
Data-at-Rest & Media	Missing encryption
Data-in-Use	Datacenter RAM
Access Control	Privilege Escalation
Auditing & Monitoring	Lack of audit trails
Injection/Input Vuln.	SQL Injection

4.4.1. Authentication

Authentication is the process or action of proving or showing something to be true, genuine, or valid. Best industry practices in authentication, such as multifactor authentication, vulnerable password rest requests, and robust error messages, are traditionally considered and tested during a system/web-application assessment.

4.4.2. Session Management

Session management is the rule set that governs interactions between a web-based application and users. Browsers and websites typically use HTTPS/HTTP to communicate, and a web session is a series of requests and response transactions created by the same user after authentication. In most cases, the user and server communicate with a special token so that the user does not have to repetitively re-authenticate with each new server page.

Current best practices in session/token management such as setting cookie flags (e.g. Secure and HTTPOnly), generating a random session token, and session timeout intervals, are examined.

4.4.3. Data-in-Motion

Data-in-motion is the data transfer of data between a client and a server. Continually changing, this category of vulnerabilities include industry best practices in how to transmit the data such as confidentiality (e.g. encryption/decryption) and integrity (e.g. hashing both data or passwords) controls used during data transmission between clients and servers.

4.4.4. Data-at-Rest and External Media

Data-at-rest refers to data that is stored on a system/server. In an ever-changing dynamic landscape, this category of vulnerabilities includes industry best practices in the storage of the data to include controls such as confidentiality (e.g. encryption/decryption best practice algorithms), integrity (e.g. SHA512, bcrypt) and proper rotation/storage of encryption/decryption keys.

4.4.5. Data-in-Use

Data-in-use is typically considered in shared memory system such as datacenters where different client virtual machines or applications are running on a semi-trusted hardware/software infrastructure models (e.g. in Software as a Service (SaaS), Platform as a Service (PaaS), or Infrastructure as a Service (IaaS), etc.). These categories of vulnerabilities can be considered, such as confidentiality, integrity and availability of information, on these shared system resources during an assessment (e.g. questions, ISO certifications, etc.).

4.4.6. Access Control

Access controls are security techniques that regulates who or what can view/use resources stored in a system/application. Current best practices in access controls, such as user-based or host-based, are traditionally coded into the software system/web-application architecture. These controls are typically tested during a risk penetration assessment.

4.4.7. Auditing and Monitoring

Systems and application generate logs before and after critical functions take place. These logs are stored in the system/server backend for regulatory requirements, performance indicators and other analytics. Current best practices in auditing, such as user login/logout activities, user access activities, and user upload/download activities are traditionally tested during an assessment.

4.4.8. Injection and Input Vulnerabilities

Injections and input vulnerabilities enable malicious entities to insert malicious code into running systems to deviate the system from normal functionality. In some cases, these vulnerabilities can result in the unauthorized exposure of sensitive information.

Current best practices in injection and input vulnerability controls such cross-site scripting (XSS), cross-site request forgery (CSRF) and SQL injection are traditionally considered and tested during a system/web-application assessment.

4.5. Organizational Control Requirements

At the organizational-level other non-technical requirements, perhaps managed by completely distinct organizational teams, may be required to manage the risk and meet legal requirements. Three traditional sub-categories at the organizational-level are

policies/procedures, physical security, and budgeting for adverse circumstances.

4.5.1. Policies and Procedures

Organizations should have policies in place, which they consistently follow to avoid all kinds of legal ramifications (e.g. from discrimination to security). In addition, findings discovered during technical reviews need to correctly identify which, if any, policies are effected. (For example, if a system is missing authentication, then both a federal requirement as well as organizational policy is violated.)

Procedures must also be in place, and specifically in writing. Specific procedures, which must be in place at the federal level, include business continuity and potentially disaster recovery plans.

Lastly, this sub-category encompasses the administrative controls to electronic patient data, which are federal requirements. HIPAA requires certain administrative controls of patient information and the lack therefore needs to be correctly identified.

4.5.2. Physical Security

This component describes the physical security aspects of the system, if any, which are requirements in the United States Federal HIPAA laws.

4.5.3. Budget for Adverse Effects

Risk assessment traditionally includes developing a budget for adverse effects. Many organizations are not storing-up financial resources in accordance to the risks being generated. (For example, HITECH requires notifications when over 500 patients are affected.) Digital Guardian [23] has various reports on current costs per record; the costs vary with time. Risk management, in addition to an application penetration test and connection to policies, should have a financial penalty indicator for both correct insurance coverage and potential organizational indirect costs/penalties. Thus, simply indicating that a web-application is vulnerable to CSRF, may really have no budgetary ramification under certain other conditions (e.g. no sensitive information, few people involved, etc.)

5. A Risk Assessment Library

This paper contributes a new open source risk assessment library example to enable researchers, penetration testers, risk assessment managers and institutions to further expand on a consistent risk assessment findings library with their policies, procedures, organizational controls and legal requirements.

Bug libraries/dictionaries are being maintained by large organizations such as NIST and MITRE (described in Section 2). Bug libraries/dictionaries do not include risk assessment (e.g. penetration test) findings. As such, a risk assessment from one organization is traditionally written with completely different language choices from another organization making analysis extremely difficult.

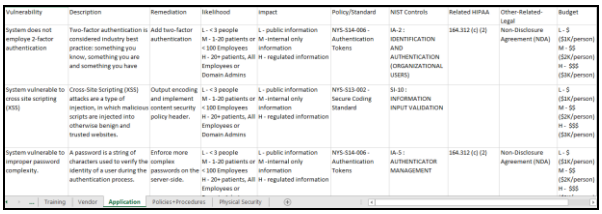
Eventually, our open-source risk assessment library will need to be maintained by either a standardizing body (e.g. NIST, MITRE) or maintained by industry to orchestrate round-tables between different community discussions (e.g. law enforcement, penetration testers and regulators) to continuously update best practice language for such a library.

Many organizations already may have their own framework in place for risk management. Our library does not affect the framework, but rather helps organizations use standardized language when writing up their risk assessment reports. For example, one penetration tester may indicate on a report "a SESSID is not random." Another penetration tester may indicate, "A session id is not random." A third penetration tester may indicate, "A session identifier is not random." As we can see from this example, three different penetration testers wrote up three entirely different findings making meta-analysis on the findings extremely difficult. In addition, external penetration testers may not indicate if findings are requirements under regulations such as HIPAA, PCI or SOX.

This example library was developed from years working in industry with risk management and information security. It was found during internal audits that reports for policy exceptions numbers led to difficult analysis since all risk assessment findings were written differently. A risk assessment might mention that a password configuration led to higher risk, but it would not indicate that this was in fact a policy violation. Depending on the risk assessment language, it was nearly impossible to collect accurate statistics about how many risk assessments findings were also in fact policy violations.

5.1. Example Findings Library

The open-source library example, seen in the Figure 1, applies standardized language. The columns are the following: *Vulnerability*, *Description*, *Remediation*, *Likelihood*, *Impact*, *Policy/Standard*, *NIST Controls*, *Related HIPAA*, *Other-Related-Legal*, and *Budget.*



Vulnerability	Description	Remediation	likelihood	Impact	Policy/Standard	NIST Controls	Related HIPAA	Other-Related-Legal	Budget
System does not employe 2-factor authentication	Two-factor authentication is considered industry best practice: something you know, something you are and something you have	Add two-factor authentication	L - < 3 people M - 1-20 patients or < 100 Employees H - 20+ patients, All Employees or Domain Admins	L - public information M -internal only information H - regulated information	NYS-S14-006 - Authentication Tokens	IA-2 : IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS)	164.312 (c) (2)	Non-Disclosure Agreement (NDA)	L - $ ($1K/person) M - $$ ($2K/person) H - $$$ ($3K/person)
System vulnerable to cross site scripting (XSS)	Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites.	Output encoding and implement content security policy header.	L - < 3 people M - 1-20 patients or < 100 Employees H - 20+ patients, All Employees or Domain Admins	L - public information M -internal only information H - regulated information	NYS-S13-002 - Secure Coding Standard	SI-10 : INFORMATION INPUT VALIDATION			L - $ ($1K/person) M - $$ ($2K/person) H - $$$ ($3K/person)
System vulnerable to improper password complexity.	A password is a string of characters used to verify the identity of a user during the authentication process.	Enforce more complex passwords on the server-side.	L - < 3 people M - 1-20 patients or < 100 Employees H - 20+ patients, All Employees or	L - public information M -internal only information H - regulated information	NYS-S14-006 - Authentication Tokens	IA-5 : AUTHENTICATOR MANAGEMENT	164.312 (c) (2)	Non-Disclosure Agreement (NDA)	L - $ ($1K/person) M - $$ ($2K/person) H - $$$

Training | Vendor | Application | Policies+Procedures | Physical Security

Figure 1: Risk Assessment Library

The column descriptions are presented. *Vulnerability* summarizes the found weakness of the system/application. The *Description* column describes the weakness in high-level terms. The *Remediation* column describes how to remediate the issue, if any. The *Likelihood* column describes the probability of occurrence. The *Impact* column estimates the measure of damage from occurrence of the exploited vulnerability. The *Policy/Standard* column gives the related and/or violated policy/standard. The *NIST Controls* column expresses the related NIST SP 800-53 controls. The *Related HIPAA* column expresses the related US Federal HIPAA sections. The *Other-Related-Legal* column expresses other legal documentation such as Non-Disclosure Agreements, Service Level Agreements, etc. The *Budget* column gives an estimated budget to prepare for damage.

The different sheets of the page represent different risk assessment aspects such as the following: web-application, physical security, training, vendors, policies and procedures, vendor and legal-requirements. Further work on the language employed during risk assessment should be legally verified to so that the business understands legal ramifications during data breach or hacking legal cases.

To date, there are no publically available risk assessment libraries to connect the components of organizational risk discussed in Section 3. Furthermore, a risk assessment at one organization can be entirely distinct from a risk assessment at a different organization. The disconnect between risk assessments can be on what was analyzed as well as the language used to write-up the assessment since everyone is predominately different units of measure. As data breaches, legal requirements, government audits and security requirements expand and develop, a uniform library for assessing the risk will become essential for consistent auditing, cross-communication and cross-collaboration between medial entities.

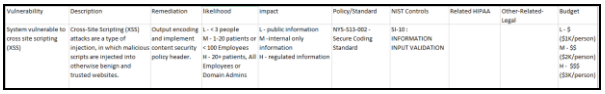
5.2. Findings Library Use Case

One well-known vulnerable application specifically designed to teach penetration concepts is Google's Gruyere [20]. This application runs in its own sandbox and teaches users about common web application vulnerabilities that can be discovered during a penetration test. The application thankfully does not house any electronic patient health information (ePHI).

A penetration test of this application, as many applications used in medical settings, which actually do house ePHI, does not require consistent language and may not unearth violations to organizational policies, legal requirements, breach budget, or software development trainings. As such, a strictly technical penetration test on a medical web application can lead to incorrect organizational risk management as it lacks any connection to the other risk components, such as policies, procedures and legal requirements, to inform on the overall risk and the potential legal breach budget requirements.

The New York State (NYS) Information Technology Security (ITS) Policies [21] are wonderful examples for this proof-of-concept risk assessment when exploring the following risk assessment findings; however, in reality the organization should map their own policies/procedures into their specific risk assessment findings.

5.2.1. Findings 1: Stored Cross-Site Scripting (XSS). Cross-site scripting is considered an injection/input validation software development programming error. HIPAA does not specifically mention cross-site scripting within the law itself, but other interpretations about access control, confidentiality, integrity and availability could potentially affect legal recourse. In addition, considering the NYS policies, accepting an XSS vulnerability may be in violation of the organizational Secure Coding Standard (NYS-S13-002), as it requires systems free of such software bugs. During a risk assessment, not only should the finding be identified, it should be mapped with organizational policies/procedures, etc., to inform on the overall organizational risk, seen overall in Figure 2 and in detail in Tables 4 and 5.



Vulnerability	Description	Remediation	likelihood	Impact	Policy/Standard	NIST Controls	Related HIPAA	Other-Related-Legal	Budget
System vulnerable to cross site scripting (XSS)	Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites.	Output encoding and implement content security policy header.	L - < 3 people M - 1-20 patients or < 100 Employees H - 20+ patients, All Employees or Domain Admins	L - public information M -internal only information H - regulated information	NYS-S13-002 - Secure Coding Standard	SI-10 : INFORMATION INPUT VALIDATION			L - $ ($1K/person) M - $$ ($2K/person) H - $$$ ($3K/person)

Figure 2: Risk Library – XSS Vulnerability

Table 4: XSS Vulnerability (Columns 1-4)

Vulnerability	***Description***	***Remedy***	***Likelihood***
System vulnerable to cross site scripting (XSS)	Cross-Site Scripting (XSS) attacks are a type of injection,	Output encoding and implement content security	**L** - < 3 people **M** - 1-20 patients or < 100 Employees

	in which malicious scripts are injected into otherwise benign and trusted websites.	policy header.	**H** - 20+ patients, All Employees or Domain Admins

Table 5: XSS Vulnerability (Columns 5-7 and 10)

Impact	*Policy/ Standard*	*NIST Controls*	*Budget*
L - public information ***M*** -internal only information **H** - regulated information	NYS-S13-002 - Secure Coding Standard	SI-10 : INFORMATION INPUT VALIDATION	**L**-$ ($1K/person) **M** - $$ ($2K/person) **H** - $$$ ($3K/person)

5.2.2. Findings 2: Denial of Service. The application is susceptible to a denial of service attack based on how the application is constructed. Again, denial of service is not mentioned in HIPAA directly; however, organizations are required maintain the availability of ePHI which is within an application. Connecting this finding to policies, for example the NYS ITS policies, a violation of the Secure Coding Standard (NYS-S13-002) occurs, which should be managed. Figure 3 shows the overall library with details in Tables 6 and 7.

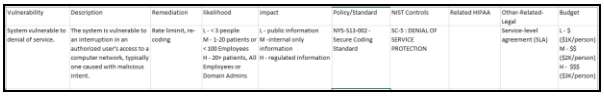


Figure 3: Risk Library – DoS Vulnerability

Table 6: DoS Vulnerability (Columns 1-4)

Vulner.	*Description*	*Remedy*	*Likelihood*
System vulnerable to denial of service (DoS).	The system is vulnerable to an interruption in an authorized user's access to a computer network, typically one caused with malicious intent.	Rate limiting, re-coding	**L** - < 3 people **M** - 1-20 patients or < 100 Employees **H** - 20+ patients, All Employees or Domain Admins

Table 7: DoS Vulnerability (Columns 5-7 and 10)

Impact	*Policy/ Standard*	*NIST Controls*	*Budget*
L - public information ***M*** -internal only information **H** - regulated information	NYS-S13-002 - Secure Coding Standard	SC-5 : DENIAL OF SERVICE PROTECTION	**L**-$ ($1K/person) **M** - $$ ($2K/person) **H** - $$$ ($3K/person)

5.2.3. Findings 3: Cookie Manipulation. The application is susceptible to cookie manipulation meaning that the session management vulnerable. This particular finding is not discussed directly in HIPAA; however, HIPAA discusses access control standards, which may come into question in such a case where a known vulnerability exists. Again, this particular finding violates the NYS Secure Coding Standard (NYS-S13-002). A library example is given in Figure 4 and in detail in Tables 8, 9 and 10.

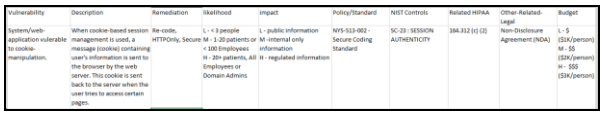


Figure 4: Risk Library – Cookie Vulnerability

Table 8: Session Mgmt. Vuln. (Columns 1-4)

Vulnerability	*Description*	*Remediation*	*Likelihood*
System/web-application vulnerable to cookie-manipulation.	When cookie-based session management is used, a message (cookie) containing user's information is sent to the browser by the web server. This cookie is sent back to the server when the user tries to access certain pages.	Re-code, HTTPOnly, Secure	**L** - < 3 people **M** - 1-20 patients or < 100 Employees **H** - 20+ patients, All Employees or Domain Admins

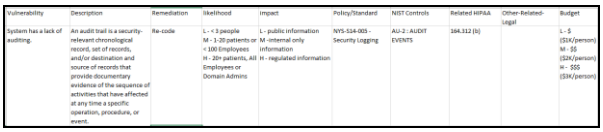
Table 9: Session Mgmt. Vuln. (Columns 5-7 and 10)

Impact	*Policy/ Standard*	*NIST Controls*	*Budget*
L - public information ***M*** -internal only information **H** - regulated information	NYS-S13-002 - Secure Coding Standard	SC-23 : SESSION AUTHENTICITY	**L**-$ ($1K/person) **M** - $$ ($2K/person) **H** - $$$ ($3K/person)

Table 10: Session Mgmt. Vuln (Columns 8 and 9)

Related HIPAA	*Other-Related-Legal*
164.312 (c) (2)	Non-Disclosure Agreement (NDA)

5.2.4. Findings 4: Lack of Application Auditing. This particular application may be found to be improperly auditing associated activities. If the application were to house ePHI, then it would be required to provide auditing records under HIPAA. This would be a direct violation of the federal law. This particular finding would also be in violation of the NYS Security Logging (NYS-S14-005) policy, so a policy exception should be put into place. A library row for this vulnerability is seen overall in Figure 5 and in detail in Tables 11, 12 and 13.



Vulnerability	Description	Remediation	likelihood	Impact	Policy/Standard	NIST Controls	Related HIPAA	Other-Related-Legal	Budget
System has a lack of auditing.	An audit trail is a security-relevant chronological record, set of records, and/or destination and source of records that provide documentary evidence of the sequence of activities that have affected at any time a specific operation, procedure, or event.	Re-code	L - < 3 people M - 1-20 patients or < 100 Employees H - 20+ patients, All Employees or Domain Admins	L - public information M -internal only information H - regulated information	NYS-S14-005 - Security Logging	AU-2 : AUDIT EVENTS	164.312 (b)		L - $ ($1K/person) M - $$ ($2K/person) H - $$$ ($3K/person)

Figure 5: Risk Library – Auditing Vulnerability

Table 11: Auditing Vulnerability (Columns 1-4)

Vulnera.	*Description*	*Remedy*	*Likelihood*
System has a lack of auditing.	An audit trail is a security-relevant chronological record, set of records, and/or destination and source of records that provide documentary evidence of the sequence of activities that have affected at any time a specific operation, procedure, or event.	Re-code	**L** - < 3 people **M** - 1-20 patients or < 100 Employees **H** - 20+ patients, All Employees or Domain Admins

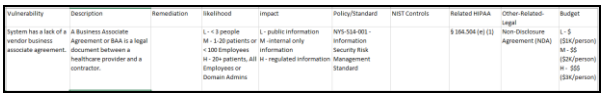
Table 12: Auditing Vuln. (Columns 5-7 and 10)

Impact	*Policy/ Standard*	*NIST Controls*	*Budget*
L - public information ***M*** -internal only information **H** - regulated information	NYS-S14-005 - Security Logging	AU-2 : AUDIT EVENTS	**L**-$ ($1K/person) **M** - $$ ($2K/person) **H** - $$$ ($3K/person)

Table 13: Auditing Vulnerability (Columns 8 and 9)

Related HIPAA	*Other-Related-Legal*
164.312 (b)	-

5.2.5. Findings 5: Lack of Vendor Agreements. This particular application may be from a vendor. In such a case, proper agreements such as a Business Associate Agreement (BAA) or other vendor requirements must be in place. If the application is housing ePHI, then both HIPAA and the organizational polices/standards (e.g. NYS ITS Information Security Risk Management Standard (NYS-S14-001)) may be violated are at stake so the connection to the laws and policies/standards needs to be clear to effectively manage the risks to the organization. Figure 6 overview (and details in Table 14, 15 and 16) shows the finding library describing the vulnerability.



Vulnerability	Description	Remediation	likelihood	Impact	Policy/Standard	NIST Controls	Related HIPAA	Other-Related-Legal	Budget
System has a lack of a vendor business associate agreement.	A Business Associate Agreement or BAA is a legal document between a healthcare provider and a contractor.		L - < 3 people M - 1-20 patients or < 100 Employees H - 20+ patients, All Employees or Domain Admins	L - public information M -internal only information H - regulated information	NYS-S14-001 - Information Security Risk Management Standard		§ 164.504 (e) (1)	Non-Disclosure Agreement (NDA)	L - $ ($1K/person) M - $$ ($2K/person) H - $$$ ($3K/person)

Figure 6: Risk Library – BAA Vulnerability

Table 14: BAA Vulnerability (Columns 1-4)

Vulnerability	*Description*	*Remediation*	*Likelihood*
System has a lack of a vendor business associate agreement.	A Business Associate Agreement or BAA is a legal document between a healthcare provider and a contractor.	Put one in place.	**L** - < 3 people **M** - 1-20 patients or < 100 Employees **H** - 20+ patients, All Employees or Domain Admins

Table 15: BAA Vulnerability (Columns 5-7 and 10)

Impact	*Policy/ Standard*	*NIST*	*Budget*
L - public information ***M*** -internal only information **H** - regulated information	NYS-S14-001 - Information Security Risk Management Standard	-	**L**-$ ($1K/person) **M** - $$ ($2K/person) **H** - $$$ ($3K/person)

Table 16: BAA Vulnerability (Columns 8 and 9)

Related HIPAA	*Other-Related-Legal*
§ 164.504 (e) (1)	Non-Disclosure Agreement (NDA)

5.2. Findings Library Summary

In summary, this use case example shows how the library can be used to standardize language for the risk assessment process therefore improving the entire process. When each person in an organization (e.g. penetration tester, lawyer) uses their own personal language to describe risk components, then managing the risk and comparing risk among organizations is ineffective. To date, no standardized library exists for such risk management requirements. This research introduces such a standardized library and presents a use case.

6. Future Work and Implications

Risk is currently being distributed across many departments in medical institutions across the United States. Most IRM solutions require the institutions to configure and customize the software to meet their own personal needs. As such, organizational risk owners may face frustrations as to what risk they are inheriting and for what exactly they are liable during a breach of regulations by the organization, especially if they are the personnel involved in accepting the organizational risks.

In fact, as people leave/retire and newer staff replace existing medical staff roles, the newer staff legally need to know what responsibilities and risks have already been accepted at their job-level by their predecessor. Perhaps future job postings should reflect the expected level of risk, which is associated with the job position. For example, breaches investigated by the US HHS OCR which result in organizational corrective action plans are inherited and stay with the breached organization for the duration of the organization's legal responsibilities--even if the involved breach staff leave the organization. Newer staff legally need to be appropriately informed of the organizational risks, which they have inherited since these risks fall into the jurisdiction of their job responsibilities. The new employees have inherited the risk from a professional perspective, which may even need to be clear prior to accepting their new position.

In addition, to further the risk management science, assessment findings risk measurements as determined by impact and more importantly likelihood should be further standardized across industries (e.g. healthcare) based on standardized metrics including the presences of well-known organizational controls (e.g., NIST Security Controls in SP 800-53 [22]). This standardization will improve understandings and cross-communication during international, national, state, city, and other legal scrutiny.

7. Conclusions

Risk management is essential for medical organizations to properly appropriate funds, budget human resources, fulfill their international legal requirements and uphold the privacy/security of their proprietary and patient information. For example, Facebook [8] recently announced that it had appropriated three-billion dollars to prepare to pay government privacy fines. In addition, the OCR Breach Notification website is filled with covered entity corrective action plans and their related fines. As privacy, security and data breaches expand, so will court cases and required audits. In such legal cases, a standardized and connected risk assessment library and language will become indispensable for organizational risk communicating and collaborating.

8. References

[1] U.S. Government. 2019. Office of Civil Rights. Retrieved from: https://www.hhs.gov/ocr/index.html

[2] HIPAA Journal. 2019. HIPAA and HITECH. Retrieved from: https://www.hipaajournal.com/hipaa-and-hitech/

[3] NIST. 2019. Cybersecurity Resource Center. Retrieved from: https://csrc.nist.gov/Topics/Security-and-Privacy/risk-management/risk-assessment

[4] MITRE. 2019. Common Attack Pattern Enumeration and Classification. Retrieved from https://capec.mitre.org/

[5] NIST. 2019. National Vulnerability Database. Retrieved from: https://nvd.nist.gov/

[6] MITRE. 2019. Common Weakness Enumeration. Retrieved from: https://cwe.mitre.org

[7] NIST. 2019. Bug Framework. Retrieved from: https://samate.nist.gov/BF/

[8] Jeff Horwitz. 2019. Facebook Sets Aside $3 Billion to Cover Expected FTC Fine. Retrieved from: https://www.wsj.com/articles/facebook-sets-aside-3-billion-to-cover-expected-ftc-fine-11556137113

[9] B. Xing, L. Gao, J. Zhang and D. Sun, "Design and Implementation of an XML-Based Penetration Testing System," 2010 International Symposium on Intelligence Information Processing and Trusted Computing, Huanggang, 2010, pp. 224-229.

[10] K. P. Haubris and J. J. Pauli, "Improving the Efficiency and Effectiveness of Penetration Test Automation," 2013 10th International Conference on Information Technology: New Generations, Las Vegas, NV, 2013, pp. 387-391.

[11] H. Radwan and K. Prole, "Code Pulse: Real-time code coverage for penetration testing activities," 2015 IEEE International Symposium on Technologies for Homeland Security (HST), Waltham, MA, 2015, pp. 1-6.

[12] Lei Liu, Jing Xu, Chenkai Guo, Jiehui Kang, Sihan Xu and Biao Zhang, "Exposing SQL Injection Vulnerability through Penetration Test based on Finite State Machine," 2016 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, 2016, pp. 1171-1175. doi: 10.1109/CompComm.2016.7924889

[13] A. Blome, M. Ochoa, K. Li, M. Peroli and M. T. Dashti, "VERA: A Flexible Model-Based Vulnerability Testing Tool," 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation, Luxembourg, 2013, pp. 471-478.

[14] H. M. Z. A. Shebli and B. D. Beheshti, "A study on penetration testing process and tools," 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, 2018, pp. 1-7. doi: 10.1109/LISAT.2018.8378035

[15] A. Tetskyi, V. Kharchenko and D. Uzun, "Neural networks based choice of tools for penetration testing of web applications," 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), Kiev, 2018, pp. 402-405.

[16] G. Chu and A. Lisitsa, "Poster: Agent-based (BDI) modeling for automation of penetration testing," 2018 16th Annual Conference on Privacy, Security and Trust (PST), Belfast, 2018, pp. 1-2. doi: 10.1109/PST.2018.8514211

[17] N. A. Almubairik and G. Wills, "Automated penetration testing based on a threat model," 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), Barcelona, 2016, pp. 413-414.

[18] Joshua Eckroth, Kim Chen, Heyley Gatewood and Brandon Belna. 2019. Alpaca: Building Dynamic Cyber Ranges with Procedurally-Generated Vulnerability Lattices. In Proceedings of the 2019 ACM Southeast Conference (ACM SE '19). ACM, New York, NY, USA, 78-85.

[19] Lee Epling, Brandon Hinkel and Yi Hu. 2015. Penetration testing in a box. In Proceedings of the 2015 Information Security Curriculum Development Conference (InfoSec '15). ACM, New York, NY, USA, Article 6, 4 pages. DOI: https://doi.org/10.1145/2885990.2885996

[20] Bruce Leban, Mugdha Bendre and Parisa Tabriz (2017) Web Application Exploits and Defenses. Retrieved from: http://google-gruyere.appspot.com/

[21] New York State. 2019. ITS Security Policies. Retrieved from: https://its.ny.gov/eiso/policies/security

[22] NIST. 2019. NIST Special Publication 800-53. Retrieved from: https://nvd.nist.gov/800-53

[23] Digital Guardian (2019) What's the Cost of a Data Breach in 2019? Retrieved from: https://digitalguardian.com/blog/whats-cost-data-breach-2019