SECURITY INNOVATION

Securing the Modern Enterprise:

Software Total Risk Management (SToRM) Framework

Table of Contents

> ***Haphazard or random approaches to securing application software usually leads to wandering in the wilderness, seeking to find answers that can only come about from deliberate planning, analysis, and willingness to use your resources in the right ways, at the right time, using the right resources. Throwing technology into a situation of software defects without a prepared workforce or sufficient analysis takes you further away from actually solving software security issues.***
>
> Mark Merkow
> Application Security Architecture and Design, WageWorks

Software Powers the World

It's no secret, software forms the basis for how consumers and businesses operate. Phones, homes, cars, infrastructure, commerce, communication and yes, even hardware, all depend on software. Unfortunately, the marvelous conveniences of our on-demand economy have also created near-infinite attack surfaces for our adversaries. An always-on, connected-everywhere world doesn't just put digital data at risk. Cyber and physical boundaries are quickly disappearing. If you subscribe to the theory that there is no safety without security (as we do) you'll understand why we have a dire need to get serious about software security.

Consider the wide variety of security disciplines: physical, people, data, infrastructure, networks, crisis management and more. How dependent are each of these on software today? How much do these seemingly disparate areas interact?

Home appliances, alarm systems and cars have millions of lines of software code required to function. Offices have RFID badges for door access, digital identity management systems and connectivity for communication devices. Most organizational infrastructure consists of enormous amounts of digital assets and when something goes wrong, we communicate via email, phone, or IMs. All driven by software.

> ***Consider how many things could go wrong if each one of these things had obvious security defects.***

Leaders looking to decode cybersecurity need to understand which products, systems, and teams are putting their business at the most risk so they can deploy appropriate action plans and controls. Trying to do so without considering software risks is akin to driving a car blindfolded.

Multi-faceted Considerations

Business requirements and concerns come in many forms, all of which need to be considered and have an impact on evaluating prioritization of risk:

- Regulatory and compliance mandates
- Customer and legal obligations
- 3rd-party services
- Supply chain dependencies
- Intellectual property protection
- Critical systems where safety may depend upon cyber/software security.

The lack of systematic and adaptive risk management approaches to understand and measure software security risk and protect software systems poses major challenges to business worldwide. In this guide, we will offer practical advice with a focus on software-based system assessment and action plans to enable better software security and make smart decisions on where to prioritize cybersecurity efforts.

Conventional approaches to software security are not risk-based.

Organizations have become overly reliant on vulnerability scanning. Scanners are akin to motion sensors - great for low risk, but not great for higher risk areas. Scanners are notorious for generating false positives, indicating problems that either are non-existent or cause additional time spent on needless investigation. Utilization of scanners frequently fails to address each application's unique code-, system-, business logic- and workflow-level vulnerabilities. More importantly, it provides little practical guidance on prioritizing threat remediation or creating a roadmap to guide enterprise software security posture improvements.

Risk Assessment should be Relative

High-risk software systems require more security controls and more rigor – and organizations often struggle with this calibration. We have developed a new framework that can be utilized to identify and prioritize the highest risk software applications through rapid, relative risk analysis and threat modeling to enable an accurate assessment of software risks and lead organizations to the development of a customized remediation roadmap that will help manage the business more effectively – from security engineering activities and policies and procedures to tools, training, and related efforts. We call the approach **SToRM**, for Software Total Risk Management.

Taken by a SToRM

The value of understanding security risks is only realized when you can decide where (and whether) to dive deeper into specific problems and take appropriate risk mitigation actions.

Below is a summary of the SToRM approach. Each component will be discussed in depth in subsequent sections.

FIGURE 1. HOLISTIC DIAGRAM

- **Asset Rating:** Document critical software assets, prioritize business applications and rate them on a scale that makes sense for the situation/company
- **Software Development Lifecycle (SDLC) Gap Analysis:** Augment internal and customer requirements with industry best practices and see potential gaps
- **Risk Discovery and Assessment:** Perform a threat-based review of applications and/or systems-level technical specifications and attack vectors
- **Application Re-ordering:** After threat modeling, re-assess the risk ranking of your applications and put them into tiers based on standardized risk criteria
- **Security Test Calibration:** Construct a security testing framework that ensures breadth and depth of testing is commensurate to application criticality

> ***The SToRM Methodology provided here represents the best thinking by highly-experienced and keenly-aware software security professionals to help you to avoid the pitfalls and landmines ever-present in the software development world. SToRM helps you to take the bull by the horns and bring about an effective and provably-secure software development lifecycle.***
>
> Mark Merkow
> Application Security Architecture and Design, WageWorks

SToRM in Action:
Application Asset Rating

Applications rely heavily on their environment in order to work properly. They depend on the OS to provide resources like memory and disk space; they rely on the file system to read and write data; they use structures such as the Windows Registry to store and retrieve information; the list goes on and on. These resources all provide input to the software— not as overtly as the human user does—but input nonetheless. Like any input, if the software receives a value outside of its expected range, it can fail. Inducing failure scenarios can allow us to watch an application in its unintended environment and expose critical vulnerabilities.

Asset Rating

Step 1: Portfolio Analysis

Begin with a discovery phase to document critical assets. This effort should represent your full portfolio.

Step 2: Prioritize business applications

Rate them on a scale that makes sense to you, for example 1 to 10, A to F. Ensure the scale is relative and consistently used.

> ***Example:*** *You can base your asset rankings on Business Impact Analysis (BIAs) that may have already been performed, or base it on the sensitivity of data processed by these assets, the value to the business, or impact on operations - ie. what might be lost if the asset is compromised or if access is denied.*

> ***Tip:*** *A useful tool for this analysis is called Factor Analysis of Information Risk (FAIR). FAIR underlines that risk is an uncertain event and one should not focus on what is possible, but on how probable an event is likely to occur. This probabilistic approach is applied to every factor that is analyzed. The risk is the probability of a loss tied to an asset. Learn more about the FAIR Methodology to see if this tool makes sense for your analysis.*

STоRM in Action:
SDLC Gap Analysis

For systems built internally, a SDLC gap analysis is critical. This can be as simple as a questionnaire, with results compared against an industry standard, for example, PCI SSF and the associated Secure Software Lifecycle (Secure SLC) Requirements.

FIGURE 2. SDLC GAP ANALYSIS

This will help quickly gather a sampling of your development team practices and determine where your biggest gaps are, benchmarking discrepancies in the development process based on product line, technology, locality, and other factors. The result should be both the gap report and creation of a secure development policy. OWASP's Open Software Assurance Maturity Model (OpenSAMM) is a useful model to help with this activity. The Software Assurance Maturity Model (SAMM) is an open framework to help organizations formulate and implement a strategy for software security that is tailored to the specific risks facing the organization. OpenSAMM may be useful for:

- Evaluating an organization's existing software security practices
- Building a balanced software security program in well-defined iterations
- Demonstrating concrete improvements to security assurance programs
- Defining and measuring security-related activities within the organization

The SAMM project also provides a worksheet and questionnaire you can use to assess your development teams across platforms and products. OpenSAMM is free-for-use.

> ***Tip:*** *Merely documenting a policy is not a guarantee that the policy is being followed. The questionnaire should be re-issued at regular intervals, for example semi-annually or upon any significant process change. It should then be validated to ensure that progress is being made toward adherence to the policy. Keep your change requests simply organized into those affecting people (training needs), technology (tool changes), and process (activity adjustments.) You can often impact multiple people and groups with just one change, for example providing courses on web application security training fundamentals.*

A similar questionnaire should be developed for your 3rd-party IT system providers. Use it for acceptance testing before you put the outsourced application into production (or pay for it fully). This analysis, combined with your risk-ranked asset documentation, is the baseline you need to get started with STоRM.

STORM in Action:
Risk Discovery and Assessment

Risk Discovery involves the review of application- or systems-level technical specifications in order to understand exactly how the technical system in question has been designed and deployed. To ensure that the risk implications of each application are correctly assessed, start by constructing a unified view of threats and risks at both business workflow and technical system levels. Different threat modeling techniques can be used to ensure a 360-degree view. One of the best is STRIDE, created and adopted by Microsoft in the early 2000s; however, you may find others like PASTA more to your liking. There are a myriad of great models covered here.

> ***Tip:*** *Carnegie Mellon University's Software Engineering Institute lists 12 different methods for modeling.*

Software risk assessment activities should include the review and/or development of:

- Asset and data classification schemas
- Security classification schema for applications which specifies classification criterion, e.g. type of data processed, internal vs. external, etc.
- Tabletop exercises to discuss additional threat vectors
- Current secure software development/SDLC security practices and controls
- The software application portfolio risk rating framework that considers:
  - Business impacts
  - Security threats
  - Compliance and regulation mandates
  - Customer requirements
  - Operational safety and Cybersecurity risk

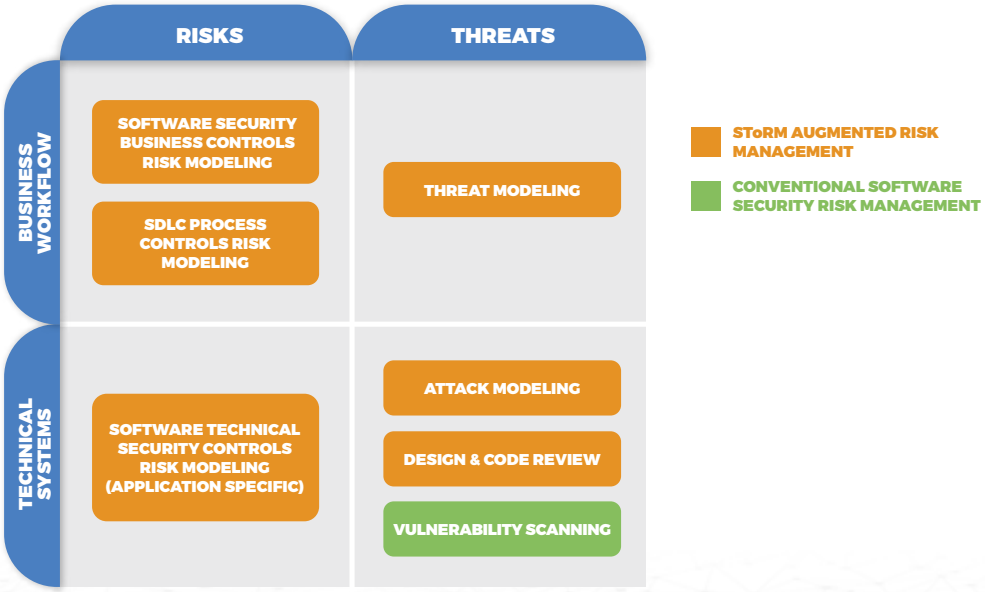

Figure 3. STORM Risk Activity Matrix

Look at each business application *and its environment* to generate a business-level threat model and as many deep threat vectors as possible:

- **Identify and prioritize high-risk applications** based on business impact, security threats, compliance mandates and operational risk.

- **Recommend security assessment activities** based on application and data security risk. Take a first pass at test recommendations. They may range from a rapid fully-automated scan for lower risk applications to a deep manual inspection for applications that pose very high business risk.

- **Create a risk rating framework** that will enable you to allocate security budget and resources to match the level of effort to the criticality of each application.

Risk modeling, incorporating Secure Software Development Life Cycle (SSDLC) and other best practices drawn from such internationally-accepted standards as the NIST Cybersecurity Framework (CSF), ISO 2700x series, and PCI-SSF, helps ensure that application vulnerabilities are viewed in the broader risk context of business asset valuation, regulatory compliance, and operational efficiency.

Threat modeling of the application workflow, coupled with attack modeling and design/code review of the application, traces all the ways that application end-users and administrators might accidentally or intentionally exploit faulty application control logic or coding errors. The insight gained from threat modeling may be used to determine the next steps for your application's lifecycle. This may include:

- Deeper or broader testing
- Rebuilding
- Replacing it with another application
- Accepting the risks as-is
- Deploying different/additional mitigation controls, or
- Taking it out of service completely

Regardless of the outcome, you will have better insight into:

- Digital assets that are most at risk and require protection
- The most likely threats to those at-risk assets
- Specific malicious attacks that could be used to realize those threats
- Design and deployment conditions under which attacks would be successful
- Mitigations or additional testing that must be conducted to reduce the identified threats or prove/disprove their existence

SToRM in Action:
Application Re-ordering

When threat modeling is completed, it's time to re-assess the risk ranking of your applications. It may be simplest to create 3 tiers of application risk: Critical, High, and Medium-to-Low risk. Place each application in its appropriate tier according to the criteria. The diagram below illustrates this approach, but keep in mind this exercise is highly contextual to each organization. Most likely, you will end up with many Medium-to-Low risk applications and only a handful of mission critical ones. The objective here is to use your limited staff and budget appropriately and strategically based on your critical analysis of risk.

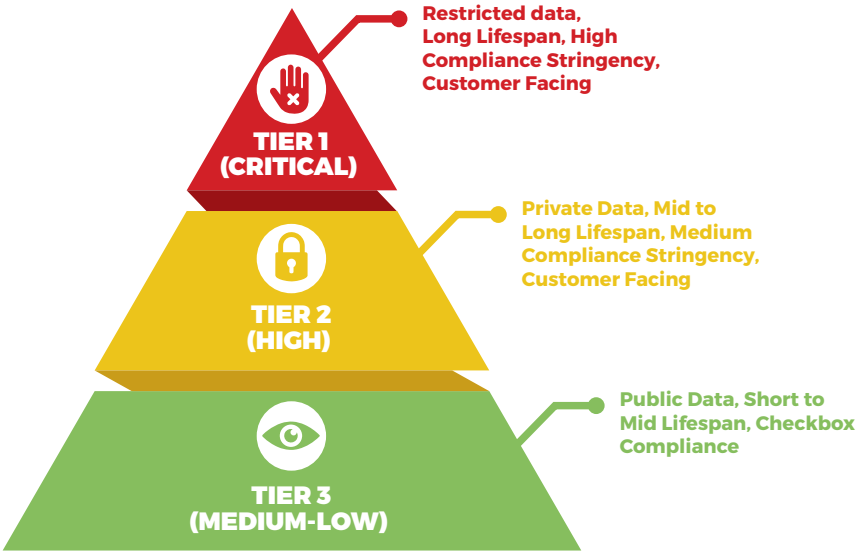

Figure 4. TIERS OF APPLICATION RISK

Data classification reflects the level of impact to your organization if the data is compromised and includes factors such as compliance mandates, federal laws and internal standards. Application attack exposure is an important factor in the relative attack risk each application carries. Some applications have very little exposure, while others are exposed to a large number of users over the internet. Some are connected to other enterprise systems, databases or web services, while others are more isolated and harder to access. **The combination of data criticality and attack exposure allows you to produce the risk-ranked grouping of your applications.**

To help group applications, even simple scoring systems can be effective. Again, choose whatever scale makes most sense to you and your organization. In this oversimplified example, we use a scale of 0 - 3, with 0 representing the lowest degree of risk and 3 representing the highest. We illustrate 5 basic categories and assign a 0, 1, 2, or 3 based on risk. The categories in this example are:

- **Data sensitivity.** Does the application store, process, or transmit sensitive information such as intellectual property, financial records, privacy information, etc. The scale should reflect the degree to which the data handled is sensitive.

- **Lifespan.** How long is the application expected to live? Statistically, the longer the life of an application, the more likely it is to be attacked and compromised.

- **Compliance.** What legal, regulatory, or other compliance mandates is this business application expected to meet? More importantly, what is the risk implication should it be out of compliance?

- **Impact of Application Loss or Compromise.** This is a measure of hurt, i.e., how painful it would be to the business or mission should this application be made unavailable for an extended period of time.

- **Customer/Internet Facing.** This is a binary 0 or 1. Many view internet-facing applications as more risky than internal. In our view, because threats come from both inside and outside an organization, both scenarios carry risk.

For each application, simply add the 5 numbers together to get a total ranging from 0 to 12. The resulting table looks something like this:

APPLICATION NAME	Data Sensitivity	Lifespan	Compliance	Lost Impact	Internet Facing	RISK SCORE	RISK TIER
Application 1	1	3	0	0	1	5	2
Application 2	3	2	3	0	0	7	1
Application 3	2	3	1	0	0	6	2
Application 4	1	1	1	1	0	4	3
Application 5	1	1	1	1	1	5	2
Application 6	2	2	1	0	0	5	2
Application 7	3	3	2	3	1	12	1
Application 8	2	3	1	0	0	6	2
Application 9	1	1	0	1	1	4	3

Figure 5. APPLICATION RISK

Let's walk through examples of 3 different applications:

Application 1:

This application helps the company to collect names and e-mail addresses for the company's newsletters. Data is stored in a shared database within a public cloud. This application was built by a third party, however the company owns all source code, maintenance, and rights to the application.

Parameter	Score	Notes
Data Sensitivity	1	Full names, email addresses
Lifespan	3	No EOL set
Compliance	0	No compliance requirements
Impact of Application Loss	0	Marketing/customer communication. Impact of loss is minimal
Customer or Internet Facing	1 (True)	Hosted on a shared Virtual Server with a shared database in public cloud

Application 4:

Application is an internal support help-desk application for support and ticketing on a specific product. Customer records are stored in a central database in an internal data center; however, it is only customer name and email address. This application was built in-house and the product it supports will be end-of-life within the next 9-12 months.

Parameter	Score	Notes
Data Sensitivity	1	Full names, email addresses
Lifespan	1	EOL <12 months
Compliance	1	Minimal internal compliance requirements
Impact of Application Loss	1	Although EOL is in less than a year, it would still be somewhat disruptive to lose access to it before then.
Customer or Internet Facing	0 (False)	Hosted on an internal server and is not accessible externally

Application 7:

Application is an operational e-commerce application. It was built by a 3rd party to sell the company's products. Once the data has been collected, it is stored in an encrypted database. Data collected is sensitive and must be treated as such.

Parameter	Score	Notes
Data Sensitivity	3	Full names, addresses, account numbers, credit card information
Lifespan	3	No EOL set
Compliance	2	PCI, PII, GDPR
Impact of Application Loss	3	Mission critical application. If compromised or taken offline it would cost the company hundreds of thousands of dollars each day.
Customer or Internet Facing	1	Hosted on a dedicated cloud infrastructure, is internet-facing, and accesses a database in a private cloud data center

STORM in Action:
Security Test Calibration

The final step in the framework builds from the analysis to culminate in the construction of an application security testing framework. For each application, you've considered the combination of criticality of data stored, transmitted or processed, plus the attack exposure in order to logically risk-rank your portfolio.

There is no standard formula for this, as risk tolerance and data mapping is always contextual to each organization; but the objective is for each application risk tier to have a specific recommended testing frequency and depth so you can apply resources intelligently. In the example below, we use 3 tiers for simplicity of illustration.

THREAT RATING	Static Analysis (Source Code)		Dynmaic Analysis (Web App Scanning)		Manual (Penetration Testing)		Threat Modeling	
	Complete	Frequency	Complete	Frequency	Complete	Frequency	Complete	Frequency
Tier 1 (Critical)	Required	Major Code Changes	Required	Major Code Changes	Required	Per-Milestone	Required	Per-Release
Tier 2 (High)	Suggested	Monthly	Required	Quarterly	Required	Per-Release	Suggested	Per-Release
Tier 3 (Low)	Optional	Quarterly	Required	Annually	Optional	As Needed	Optional	As Needed

Figure 6. SECURITY TEST CALIBRATION

SToRM: Benefits and Persistent Assets

The by-product(s) of utilizing the SToRM analysis above will ensure that technology and security teams are equipped with the necessary framework to implement repeatable software security and development best practices. The application portfolio risk assessment and classification can be leveraged to provide visibility into the state of development practices and application security across an organization's business lines. Combined with asset and application classification, you can now identify and prioritize high-risk applications based on business impact, security threats, compliance mandates, and operational risk. The result is a risk rating framework that will allow for better resource allocation and identification of high-risk areas.

SDLC recommendations:

The outcome of this gap analysis should contain a specific remediation roadmap, as mentioned above. However, the sequencing of new activities and tools needs to be considered carefully. Too often, organizations try to do too much too quickly, and efficiencies can grind to a halt.

Consider training as a critical success factor. Introduction of new tools without training on the objective, activity, and benefit is going to be significantly less effective. This is analogous to handing a pneumatic jack hammer to an unskilled laborer who's currently using a manual handpick. Maybe s/he could figure out how to use it; maybe they'll use it wrong and do more damage than good; but most likely, they simply won't bother to even try and will stick with what they know.

When rolling out training, consider each person's job function and the technology stack for which they need training. One size does not fit all when it comes to software security education. A Java developer needs very different training than a .NET developer, even though they both may be building web applications. An IT engineer tasked with protecting software in play will require different training than a software or DevOps engineer that focuses more on building security into the process.

Business level threat model:

The threat models you generate during this activity will be living, persistent assets that can be revisited in the future. Considering a significant new feature? Recently read of a new attack technique? Return to your threat model and make the necessary tweaks to view potential impact.

Software development and procurement policy:

Almost every organization has some combination of home-grown applications, COTS (commercial off-the-shelf) applications, and outsourced/bespoke applications. Each one poses similar types of risks to your organization and each one should have some type of cybersecurity acceptance testing criteria based on your SDLC policy. Documentation of security standards and checklists for product development teams (regardless of where they reside, organizationally) is a byproduct of this exercise as well. Be sure to enforce them and review them regularly. These formal procedures and steps should be used to evaluate new and existing software applications against your customer, regulatory, and security standards.

Utilizing SToRM

There are no standard formulas for Software Risk Assessments, but this strategic framework should help you to find the path. The by-products associated with the SToRM activities will ensure that product and security teams are equipped with the necessary framework to implement repeatable and reliable software security and development best practices.

ABOUT SECURITY INNOVATION

Since 2002, organizations have relied on Security Innovation for our unique software and application security expertise to help secure and protect sensitive data in the most challenging environments – automobiles, desktops, web applications, mobile devices and in the cloud. A best-in-class security training, assessment, and consulting provider, Security Innovation has been recognized as a Leader in the Gartner Magic Quadrant for Security Awareness Training for three years in a row. Security Innovation is privately held and headquartered in Wilmington, MA USA. For more information, visit www.securityinnovation.com or connect with us on LinkedIn or Twitter.

SECURITY INNOVATION