THE ART OF THREAT MODELING FOR IT RISK MANAGEMENT

Solving the application risk riddle

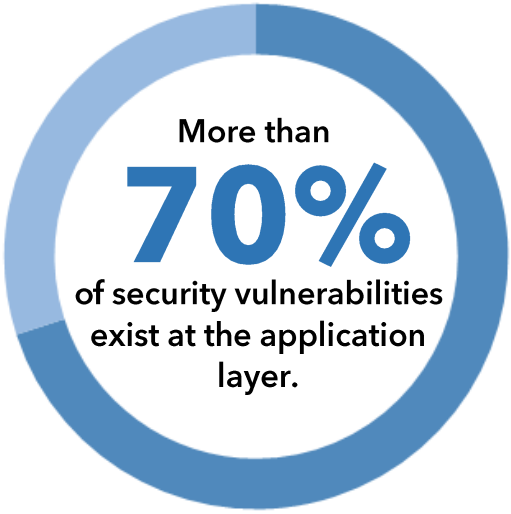
Ed Adams
CEO, Security Innovation

SECURITY INNOVATION

Table of Contents

Introduction

As recently as a few of years ago, many people viewed information security as a network problem. The assumption being that an organization's data can be protected by securing the perimeter. The most common solution was to protect the boundary of the information systems via firewalls and antivirus software. To mitigate IT risk many corporations instinctively turned to security technologies such as firewalls, IDS/IPS, two factor authentication utilizing smart cards and biometrics, and even cryptographic solutions to protect data at rest.



The truth is security is a software problem - more so than at any other layer in the information system infrastructure. Several years ago, Gartner Group, as well as several other reputable industry sources, estimated that greater than 70% of security vulnerabilities existed at the 'Application Layer' and not at the system or network layer.

Unfortunately, security is still an afterthought for most companies and security of software applications is often addressed after implementation or deployment. The downside to this is twofold:

1. Fixing an issue is much more expensive than preventing it, and
2. Every vulnerability that exists in your applications is a risk waiting to be exploited

These challenges dictate two fundamental philosophies to be adopted as "best practices" in organizations concerned with software security risks:

1. Building security into your software development lifecycle (SDLC) to reduce the number of vulnerabilities in your software applications and detect them earlier
2. Introducing an activity that facilitates the identification of risks in deployed or in-construction software applications quickly and effectively

The first best practice (building security into your SDLC) is not covered in this whitepaper; however, it is discussed in other whitepapers from Security Innovation. The second best practice is precisely what this whitepaper is all about – and the activity for finding risks in software rapidly is Threat Modeling. The concepts of threat modeling discussed in this paper are for the purpose of making better risk management decisions.

A Note on Scope

This paper is not a "how-to" for threat modeling; there are many good references for that. This paper is specifically designed for IT Risk Management, Information Security, and Management personnel that seek a more effective way to identify and prioritize risk. It is a description of the activities involved in application threat modeling and the goal of threat modeling in the context of IT risk management.

Risk Management Practices

As stated previously, many organizations use network or perimeter security technologies to mitigate risk. These technology controls are where most organizations begin their security efforts. Although these controls are tangible, the effectiveness of such tactical and reactive measures is lower than that of strategic and proactive measures such as assessment, auditing, and secure application design, development, and testing.

In order to understand how to assess risk in the application layer, the concept of threat modeling has tremendous utility. Threat modeling is a powerful exercise that can help in risk determination.

We will discuss two approaches to threat modeling:

Threat modeling of an existing application

Threat modeling during the various phases of the software development lifecycle

In each case, threat modeling needs to follow a proven methodology for effectiveness.

Because applications access data unencrypted, even if it is encrypted at rest in the database, vulnerabilities in software afford an attacker the same privileges granted to the application itself and access to that same unencrypted data. Network defenses must let software perform its functions — they don't know if it has been compromised. The need to understand risks from application vulnerabilities is great and you must understand this risk holistically - in the context of your entire information management infrastructure.

The Need for Threat Modeling

Threat modeling is a powerful technique that helps to characterize the higher level threat and separate it into more manageable sub-threats that can be addressed.

From a business perspective, IT security exists only in the context of risk management. Large corporations are driven by risk management concerns including the risk of non-compliance, the risk of data loss, and the risk of financial loss through IT theft, legal sanctions, and infrastructure downtime.

Although vulnerabilities, hackers and exploits are compelling reasons to focus on application security, they have been overshadowed by compliance and risk issues in the minds of corporate decision makers. The myriad of regulations and standards are imposing stricter IT security requirements and application security has reached a pinnacle of importance in the context of regulatory compliance.

Risk management is a key requirement of many of these regulations and is one of the most difficult processes to conduct and complete. The difficulty lies in the fact that although the high-level threat is generally well understood (breach of customer data, denial of service etc.), the underlying causes and sub-threats that can lead to it remain obscure.

The perimeter defense concept is fairly easy to grasp: keep the "bad guys" out. However, perimeter defenses also facilitate remote user access to corporate resources; thus, the distinction between trusted and distrusted users is increasingly unclear. IT divisions must remember that once a user passes perimeter defenses the burden of security lies with the applications that process data.

Both developers and users of software applications need to understand the risks those applications impose, but understanding this risk is not a trivial task. The higher-level threats are generally well-understood because they are frequently mentioned in regulatory and educational texts, e.g., inappropriate disclosure of sensitive data; however, it is difficult to address these threats without considering the underlying sub-threats that can (and often do) lead to catastrophe.

Risk management is a balancing act:

Weight of threat realization and

Weight of countermeasures

IT risk has now become an integral part of the operational risk and a headache for risk managers.

Factors that make IT risk assessment difficult:

- Software internals (especially 3rd-party) remain a bit mysterious
- Attacker techniques seem like black magic
- The impact of software vulnerabilities isn't clear

Common Pitfalls and Problems that Threat Models Helps You Avoid

Risk Management is not a simple or easy activity and when application security is involved it can be complicated and filled with uncertainty… and as anyone involved with risk knows, uncertainty is something you want to be able to avoid, or at least measure.

Penetration Testing vs. Automated Tools

The number of applications used by any given company makes the cost of penetration testing ALL applications prohibitive.

Automated tools are expensive, inaccurate and not designed for risk management teams:

- Results need to be interpreted by security professionals
- Not easily deployable for large scale applications
- Fail to understand the business logic of applications

In the end IT risk is misevaluated:

- Spend money on unneeded countermeasures
- Fail to recognize a real threat

Common Problems of Risk Management

Threat modeling assists the risk management process by helping you through common problems, such as:

TOO MANY APPLICATIONS; TOO LITTLE TIME

Many risk management and IT audit teams have hundreds or even thousands of applications to assess for risk and vulnerabilities. This is an almost impossible task without something like threat modeling.

INCORRECT RANKING OF YOUR APPLICATIONS

There may be a highly insecure application that is currently risk-ranked low because you don't realize how vulnerable it is to attack. Conversely, you may have an application that is risk-ranked high when it is actually very secure because of compensating controls you already have in place.

YOUR COMPANY IN THE NEWS BECAUSE OF A DATA BREACH

Keeping an insecure application in operation introduces risk to your organization. Often, companies have no idea which applications bring the most risk because they have been assessed in a haphazard or inaccurate manner. All too often, the vulnerabilities in these applications are exploited (by both insiders and outsiders) to steal sensitive data or money. Threat modeling provides a map of high-risk areas in software applications quickly; and the maps created are re-usable assets!

IMPROPER ALLOCATION OF LIMITED SECURITY BUDGET

Organizations often make investments in a reactive manner, e.g., an incident has occurred that causes fear and security investments are driven by this fear. The risk here is two-fold: First, you may make an investment that delivers little protection in return (emotional decisions are dangerous ones.) Secondly, you are likely to ignore an area of higher risk; after all, there is only so much money to be allocated. You need to understand where you greatest risks are first, so you can make informed security spend decisions.

Threat Modeling for Better Risk Management

Ironically, the most effective risk management measures are also the most difficult to employ and often the least correctly implemented. Many organizations are rebuffed by how difficult it is to correctly implement these measures:

- Prioritize the assessment of existing applications
- Understand which applications introduce the greatest risk
- Understand how good your developers are when it comes to security
- Know what to do security assessment results

Performing security assessments improperly are costly: you may not be able to perform assessments on all your applications and you're trying to avoid spending time and dollars on one application while another application remains open and vulnerable and costs your company a lot of money and poor publicity. It is also difficult to determine whether or not your developers are improving or repeating their mistakes. Finally, it is not always clear what your repercussion steps are. For example, if an application were developed by an outsourced partner or purchased "off the shelf", what recourse do you have to ask the software vendor or partner to fix the issues? Must you accept the risk? How to do know how to assess the extent of those risks?

Threat modeling can help in two important areas of risk management.

1. Identify important application criteria that can become enterprise risks

- Technologies
- Bad/Best practices
- Entry points
- Users
- Countermeasures
- Assets

2. Prioritize countermeasures

Understanding what threat modeling can't do is just as important and realizing its benefits. It cannot identify all vulnerabilities that can become threats in an application. It is not a replacement for a detailed security analysis, which often takes the form of a code review of penetration test.

In an ideal world, risk management budgets would allow all applications to be assessed and all developers and architects would receive professional training on secure design, coding, and testing principles. Threat modeling is a powerful exercise that can help in risk analysis and can be used either on existing applications or at each stage of the SDLC.

Threat modeling of existing applications can be performed by security professionals who perform exploratory security testing to understand the design and implementation of the application as well as identify the most obvious security failures. This results in intelligence that can be used to modify an application's risk rating and implement additional risk controls (i.e. penetration tests, replace/upgrade application, etc.).

Threat modeling throughout the Software Development Lifecycle will help define the proper security requirements at the design phase, facilitate the correct technology decisions at the implementation stage and lay the foundation for thorough security test plans by defining the negative test cases that will ensure that the application correctly handles abuse. This is where the biggest return is observed because threat modeling throughout the SDLC will help minimize the cost of developing secure software.

Threat Modeling for Existing Applications

For many software vendors, security is still an afterthought. In this model, the priority is software functionality and security is bolted on in the later stages of the software development lifecycle (SDLC). This is also the relevant model for software users who wonder about potential security failures in the products they have purchased.

The key areas threat modeling assists with are in Risk Assessment and Risk Mitigation:

RISK ASSESSMENT	RISK MITIGATION
• System characterization – define system boundaries, hardware, software, interfaces, data etc. • Threat identification – hackers, terrorists, industrial espionage, insiders • Vulnerability identification – design flaws, coding mistakes, improper configurations	• Risk assumption • Risk avoidance • Risk limitation • Risk planning • Research and Acknowledgment • Risk transference

Threat modeling facilitates these components of Risk Management with respect to application security assessment because (a) is was created specifically for software applications, and (b) it relates directly to risk management in approach and business-related risk identification. A common approach for threat modeling existing applications is a three step process consisting of:

The first step, analyzing the application, consists of identifying the application's features and user/attacker entry points. During this process it is important to note feature characteristics such as its relevancy to security and access level required to perform related tasks.

The second step, determining threats, is the most challenging aspect of threat modeling. This step consists of breaking down the higher level threat into sub-threats that can be more easily addressed. Various models can be used to categorize threats. The STRIDE[1] model can be used as a starting point for the characterization and identification of high-level threats. An example of a high-level threat would be when a malicious user escalates privileges. We must identify failure conditions (the sub-threats) that would lead to the realization of this threat. One way to gain elevated privileges is for an unauthorized user to bypass the authentication mechanism; another way is for an attacker to steal credentials from a privileged user. There can be many more ways this attack could be conducted, but for the sake of simplicity we'll focus on these two.

Bypassing authentication may seem like an unrealistic attack vector, but we often find alternate entry points that can be used to circumvent authentication mechanisms. Intercepting another user's credentials during transport or because of improper storage is another all too common way of elevating privilege.

Once the threats and sub-threats have been identified they can be prioritized using various ranking techniques. One such technique is feature ranking. Here we score application features according their characteristics (is it a new feature with the current release? is it a security feature? is it installed by default? etc.). We then assign each feature a score that is a combined measure of the likelihood it will contain a security flaw and its potential for damage. This feature score allows us rank each identified threat by averaging the scores for features that are relevant to the threat. The broken down threats listed above are much more easily addressed when it comes to risk mitigation than the higher level threat. Bypassing authentication can be addressed by ensuring that all access routes are locked down. Stealth of credentials can be prevented by encrypting credentials during transport and storage. The threat ranking can be used to prioritize risk mitigation, and the completed threat model can drive application security testing. A

[1] STRIDE: Spoofing identity, Tampering of data, Repudiation, Information Disclosure, Denial of Service, Elevation of privilege. Howard, Michael and David LeBlanc, Writing Secure Code, Second Edition, Redmond, WA: Microsoft Press, 2002

completed threat model should provide the supporting material to prioritize risk mitigation and the framework for security testing

Threat Modeling in the SDLC

Proactive software vendors integrate security at all stages of the SDLC. For vendors or internal development teams that do so or realize they need to start doing so, threat modeling can be used to its full potential. In this context the threat model should evolve during the SDLC and guide decisions at all stages.

Threat modeling is a design process used to identify potential threats to your applications.

Threat modeling is a design process used to identify potential threats to your application. Leveraged as such, it will help guide the programming process and provide a framework for developing secure code and security testing. It also identifies both threats and vulnerabilities and provides ways to perform risk analysis. Threat modeling helps with visualizing risk. It is at the different phases of the SDLC. Threat modeling drives design and testing and is often still addressed during the testing phase.

The process by which threats are characterized and ranked can be similar to the one described previously, however the threat model will grow and be refined as the product progresses through its lifecycle.

At the requirements/design phase, for instance, a proposal for the addition of a feature may be rejected because of the additional attack vectors it creates. At the implementation phase the use of recognized encryption libraries may be elected in order to mitigate stealth of sensitive information. Just as it was the case in the previously described context, the threat model can help prioritize risk mitigation and drive the security testing effort.

Whether threat modeling is performed on an existing application or throughout the software development lifecycle it is an essential component in the risk management arsenal because it can help quantify and visualize the otherwise intangible threats an application carries. Threat modeling is not a trivial exercise and should be done with effort and precision so as not to miss any aspect of the attack surface applications expose.

Getting Started with Threat Modeling

The most important prerequisite to threat modeling an application is to understand what the application does and how it does it. This will help identify the assets that it uses and needs to protect (application decomposition may be necessary to understand complex applications). You need to understand the business purpose of the application, its architecture and components (a system or component diagram is most useful here), and the assets that the application touches or manages. This will help you identify users (both intended and unintended) as well as the high level threats facing the application. When identifying potential attackers, don't dismiss the most damaging class of unintended users: the insider attacker. Once the application assets and potential attackers are identified we can derive the high level threats to the application. These may include denials of service to system components, data theft, data corruption, and so on.

Identifying entry points

From a network perspective, entry points are identified as those external or internal routes that ultimately lead to an application running on a server or workstation. Identifying entry points from this perspective is fairly straightforward. Assuming the proper network protections are in place and that the application hosts are fully patched, there is an important residual risk that comes from the applications themselves. Identifying the entry points for an application is not always simple because we have to consider the environment in which the application operates. Applications may handle input from the operating system, user interfaces, local or external databases or other applications. The first step in threat modeling an application is therefore to identify all possible input to the application. One common mistake is to assume that all attack entry points reside in the user interface.

Identifying Realization Conditions

One of the most challenging and critical aspects of threat modeling is identifying the conditions in which the high level threats can be realized. Just knowing the high level threats will not necessarily help us prioritize our risk mitigation efforts – many applications will have the same high level threats (e.g. denial of service, database corruption, data theft and so on). The added value in threat modeling applications is that the identification of realization conditions will help determine the likelihood of a threat being realized. This is done by examining the many aspects of an application including the development language, the network protocols used, the use of cryptography and so on. Other factors to consider when gauging the likelihood of an attacker reaching the next step include existing controls (internal or external), auditing and logging, the use of coding "best practices," attacker techniques and attacker profile.

One of the most challenging and critical aspects of threat modeling is identifying the conditions in which the high level threats can be realized.

An application that resides behind a corporate firewall, for example, that is available only through the intranet can be considered relatively safe from external attacks. This would mean that we can assume the attacker profile is 95% internal and 5% external because some other outward facing application could be misconfigured or vulnerable and could give an external attacker access to the corporate intranet.

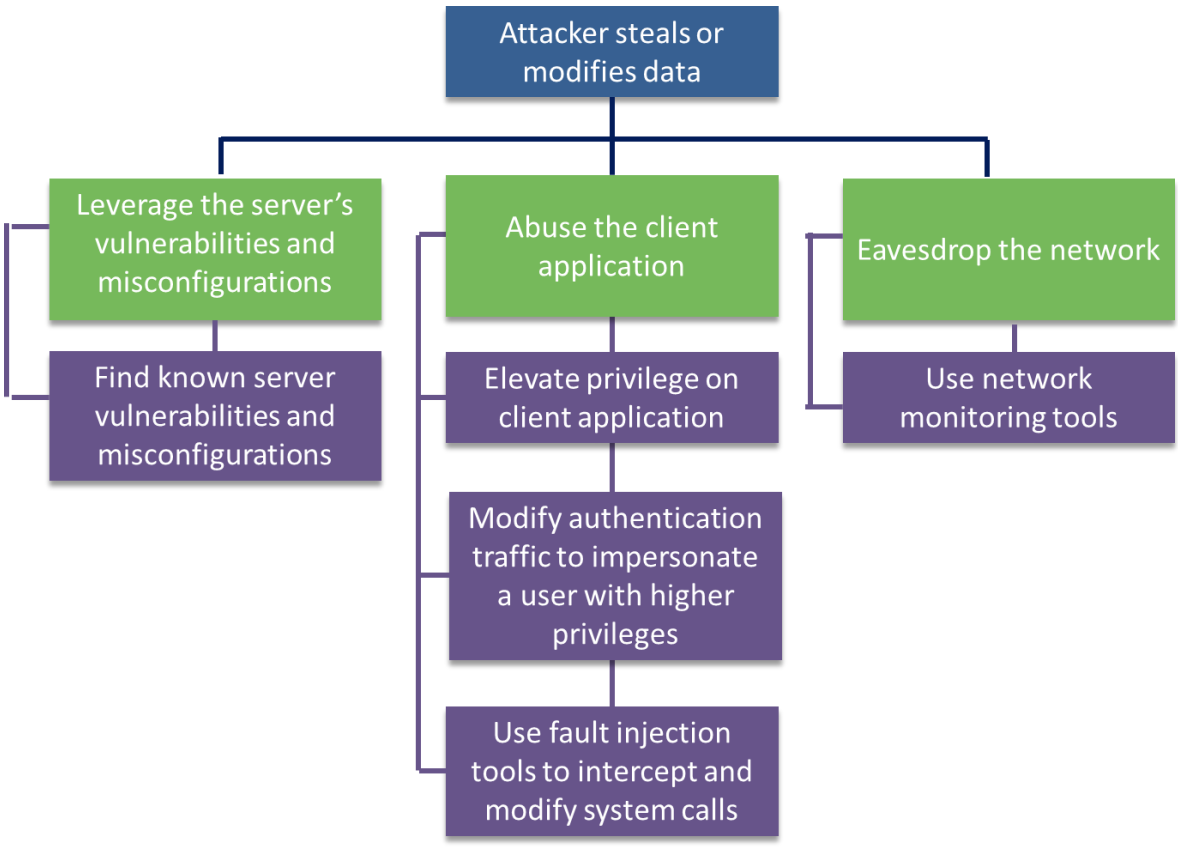
Other things come into play as well like our knowledge of the application's history. Was the authentication feature built-in to the application from the start or was it added at a later point? Every interface provides another attack surface; therefore, multi-layered applications with legacy code on the back end and newer code in subsequent layers, (i.e. transaction server, application server, web/EJB server, user interface layers, etc.) are particularly difficult to assess because there are so many attack points to consider.

Once you have an understanding of attack surfaces, you need to determine realization conditions, (the damage potential of each attack.) Damage potential can be assessed by combining users' access level (anonymous external attacker, internal user, admin, etc.) with the attack path and likelihood and the criticality of the data and system at risk.

Following is an example of a risk realization using a common scenario that we can all understand: data theft or modification from an external threat. Assume that there is a remotely accessible client which authenticates the user with a username-password combination and provides access to a back-end database server. The user in this scenario provides a username and password hash (via SSL) and receives certain permissions as a result. There are three possible ways data can be compromised:

1. By abusing the database server vulnerabilities/configuration
2. Over the network
3. By abusing the client application

For these attack scenarios consider the diagram on the following page to plot how an attacker could obtain administrator username and permissions with the attacker's password hash by modifying intercepted traffic and impersonating a user with higher privilege:



Mapping Your Findings into Your Risk Management Framework

Risk Management is a discipline but it is also a balancing act of constantly weighting threat realization and impact, countermeasures and controls. Risk Management frameworks are constantly refined to identify, assess, and predict known attack scenarios, uncover new attack scenarios, and help security teams and organizations decide where to best invest its funds to protect its assets. Threat modeling fits into existing risk management frameworks by providing more efficient risk management in the realm of information systems and technology.

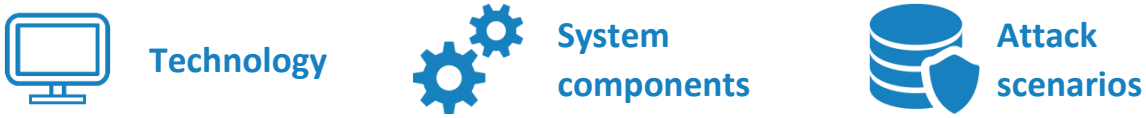
The biggest risk management challenge is determining risks in applications and information systems. When you are able to identify system components and attack surfaces of applications, you are more easily able to discuss threats in an informed manner and predict possible attack scenarios.

More importantly, threat modeling provides a critical risk management element for applications which has historically been a challenge for many organizations – deterministic probability. Threat modeling helps you determine attack possibility and success probability, damage potential, and the effect of various mitigating controls. Further, it allows you to emulate different compensating controls before deployment. Finally, threat modeling yields a persistent asset which you can at later dates as new threats are realized. A simple change to the model will tell you if you already have a mitigation control for the new threat.

Communication Critical Issues to the Application Development Team (In Their Language)

A common challenge for risk management teams is communicating with application development teams in their language. The language of risk, assessment, mitigation, and compensating controls is a foreign one to most application development teams; however, threat modeling provides a convenient bridge.

Threat modeling results can be easily communicated to development teams, outsource partners, and/or vendors because they are expressed in terms that those teams understand, specifically:



Therefore, mitigating controls are also easily identified and discussed between IT Risk and Application Development teams. A risk management team can express and verify existing controls in terms of attacks on specific system components or technology implementations; conversely, application development teams can implement additional or new controls in an informed manner because they understand the risk being communicated.

Threat modeling is a powerful visualization and communication tool for risk management and application development teams that can assist in many aspects of risk management and software construction. For risk management teams, threat modeling yields rapid risk assessments of software applications with system characterization in risk management terms. Furthermore, it provides threat and vulnerability identification in a language that is instantly understandable to development organizations. Identification of technology controls that are suited to combat the threats identified also assist risk management teams with risk mitigation and provides a common ground for both teams to discuss and agree upon best practices for secure coding, deployment configurations, and application assessment.

About the Author

Ed Adams is CEO of Security Innovation, a cybersecurity firm that specializes in application security testing and training. He is a software executive with over 20 years of experience in the industry working in senior management positions at Rational Software, Lionbridge, Ipswitch, and MathSoft. He was also an engineer for the US Army and Foster-Miller earlier in his career. Adams is a Ponemon Institute Fellow, sits on the board of the National Association of Information Security Groups (NAISG) and the International Secure Software Engineering Council (ISSECO), and is a Privacy by Design Ambassador. No stranger to the podium, he has presented to thousands at numerous industry events, including Solutions for 2020 and Beyond: Spotlight on Cybersecurity and the Tokyo 2020 Olympics, RSA Conference, C3 Cybersecurity Summit and Connected Security Expo @ISC West. Ed has also contributed commentary for media outlets such as, CSO Magazine, SC Magazine, and CFO Magazine. He has also been a regular contributor to New England Cable News and CIO Update, to name a few.

About Security Innovation

Since 2002, Security Innovation has been the trusted partner for cybersecurity risk analysis and mitigation for the world's leading companies, including Microsoft, Sony, GM, Disney, Google and Dell. Recognized as a Leader in the Gartner Magic Quadrant for Security Computer-Based Training for the third year in a row, Security Innovation is dedicated to securing and protecting sensitive data in the most challenging environments - automobiles, desktops, web applications, mobile devices and in the cloud. Security Innovation is privately held and headquartered in Wilmington, MA USA. For more information, visit www.securityinnovation.com or contact us at info@securityinnovation.com.