SECURITY COMPASS WHITEPAPER

Shifting Towards Scalable Threat Modeling

Security Compass

Threat modeling is a core activity in the process of building technology that can be trusted, allowing for the critical analysis of applications, using security and engineering resources to identify weaknesses that adversaries will attempt to exploit.

In many organizations, at the beginning of a software project, a functional specification is generated, architects provide a design to development, and code is produced. Once coding begins, various security testing tools are used to identify errors that could result in vulnerabilities. Late in the development lifecycle, penetration testing and dynamic analysis can augment other testing methodologies.

But this process is reactive; it focuses on identifying security issues after they are produced. More mature organizations also focus on prevention. Proactive threat modeling is embedded at the beginning of the development life cycle and seeks to identify and address design flaws before their implementation into code.

This can include threats related to the software platform and deployment environment. Security may research MITRE’s ATT&CK Framework to identify tactics, techniques, and procedures (TTP) used by likely adversaries. Software architects, engineers, and security teams may meet to diagram data flows in the application and identify trust boundaries.

If this sounds like a lot of work, it’s because it is. Traditional threat modeling is typically manual, and therefore takes time and resources. Even large and well-staffed organizations face challenges with manual threat modeling.

Challenges with Traditional, Manual Threat Modeling

Resources

Manual threat models require senior security and development resources to discuss architecture, complete questionnaires, produce data flow diagrams, and select controls. These resources are scarce and in high demand. Allocating for days or weeks of threat modeling exercises is not practical in most organizations.

Consistency

Ideally, organizations will identify threats and apply consistent controls. However, the output from manual threat models reflects the knowledge and biases of those participating in the exercise. As team members change, identified threats and controls will also change.

Scalability

Manual threat models are not short exercises. Cataloging threats and identifying appropriate controls can take weeks. Diagramming architecture and generating attack trees and data flow diagrams (DFDs) requires days of discussion. This investment limits in-depth threat models to an organization's most critical projects.

Auditability

When a manual threat model is completed, the threats and controls are often maintained in a spreadsheet or shared document and updated via email messages. This provides poor evidence of compliance with corporate policies and regulatory standards.

A Look At Core Threat Modeling Processes

Threat modeling is a valuable exercise, helping security and engineering teams identify threats to their projects and prescribe controls to mitigate risk. These exercises typically begin with discussions about project architecture and the technical stack required with a goal of modeling the software system. Though methodologies differ between organizations, threat modeling has four common processes:

Identify & Classify Applications

Detail assets and architecture determine the criticality of the application and required controls by analyzing business goals, security policies, compliance and privacy requirements for the project. This could include secure coding standards, prescriptive activities required by standards such as the PCI-DSS, or more general standards of "reasonable security" controls as used in Section 5 of the Federal Trade Commission (FTC) Act.

Identify Foundational Threats

The majority of application threats are universal: based on the programming language, frameworks, and other inherent aspects of the application. These foundational threats are identified using questionnaires, surveys, and interviews to identify the application's technical details, and known threats based on the technical stack.

Identify Targeted Threats

Targeted threats are unique to an application and specific threat actors. These are identified through in-depth threat modeling; a manual process wherein senior security resources and application architects create data flow diagrams to identify user interfaces, data interfaces and data processes, data storages, data entry points and data exit points and trust boundaries.

Assign & Validate Security Controls

Security controls are activities or tasks that mitigate the risk for each identified threat. These may be assigned to software developers for coding requirements or operational security for server configurations or web application firewall rules. For each control assigned, it is critical to define and assign a corresponding validation task to ensure each control is properly implemented.

Different Threats Need Different Approaches

Foundational Threats

Foundational threats are based on important factors, including: the technology used in the application including the programming language and frameworks, the deployment environment for the application, and internal policies or regulatory standards to which the application is subject.

For example, if a team is modeling a web application for which a user needs to authenticate to the system, several threats and controls can be identified, irrespective of the purpose of the system:

- An attacker may attempt to learn user credentials by logging into the system, so on failed logins, don't provide more information than necessary about what was incorrect.
- An attacker may attempt brute force attacks to guess passwords, therefore ensure that the system only allows a fixed number of failed logins for a fixed period.
- A man-in-the-middle attack could capture login credentials, therefore ensure that login pages use HTTPS.
- An attacker may trick a user into revealing their credentials, though a requirement for two-factor authentication would mitigate this risk.
- An attacker could "shoulder surf" a legitimate user to steal a password, but masking passwords by default would increase the difficulty of this tactic.

In this example, these threats and controls require knowledge of the project's functionality, information managed, technical stack, applicable regulatory standards, and criticality to the business - information identified using generic threat modeling - but these generic threats require no understanding of data flow, trust boundaries, or attack trees.

With foundational threats, threat modelers do not have to consider specific threat actors and their tactics, techniques, and procedures.

Targeted Threats

Targeted threats are unique to an application and threat actors, specific to the business case. These are identified through secondary threat modeling; a manual process wherein senior security resources and application architects create data flow diagrams to identify user interfaces, data interfaces and data processes, data storages, data entry points and data exit points and trust boundaries.

Software architects, engineers, and security teams may meet to diagram data flows in the application and identify trust boundaries and security may research MITRE's ATT&CK Framework to identify tactics, techniques, and procedures (TTP) used by likely adversaries.

Because of the high investment in resources required by this type of threat modeling, it is typically restricted to an organization's most critical assets.

Targeted threat modeling requires extensive time and resources to map data flow and build attack trees, and therefore is conducted on few applications. Foundational threats can be automated and used across an organization's entire application portfolio.

Automated Threat Modeling: 10% of the Effort, 90% of the Benefits

Organizations traditionally face a dilemma when building software. They can build fast to beat competitors to market with new features, or slow down development processes to test for security vulnerabilities. Traditional threat modeling exercises have contributed to this choice.

There is a third way, however, to build fast and stay safe. Automation allows organizations to identify the majority of threats quickly and assign consistent, actionable controls directly to developers, testers, and security. The 80/20 rule, or Pareto Principle, posits that 80% of the benefits from an activity results from 20% of the effort. The same principle holds true in threat modeling; 90% of the benefits can result from 10% of the effort.

This is because the majority of the threats to a project are linked to the technical stack and identified using only generic threat modeling, without diagramming and data flow analysis. Secondary, targeted threat models look at data flow and specific threat actors to identify the smaller number of the specialized threats facing a project, but this targeted threat modeling accounts for most of the overall effort.

Foundational threat modeling – focused on identifying universal threats that are controllable through secure design, development and deployment standards that can be addressed as part of the secure development lifecycle – identifies up to 92% of the threats to a project (Security Compass, 2019). Focusing on this, without labor-intensive diagramming and data flow analysis, allows organizations to build more secure software and scale threat modeling across their entire application inventory.

Secondary, targeted, in-depth threat models that look at architecture, data flow, and specific threat actors subsequently identify the final 10% of the threats facing a project, meaning security and engineering resources can then be reserved for an organization's most critical projects.

"We recommend a simple security requirements gathering and threat-modeling tool to make it as easy as possible for the developer. The goal should be self-service, whenever possible. For the highest-risk applications, we recommend engagement directly with the information security team via the liaison for full threat modeling and security requirements gathering."

Gartner, "12 Things to Get Right for Successful DevSecOps." Neil MacDonald, Dale Gardner, 19 December 2019

Security Compass

Security Compass believes in a world where people can trust technology, and it is our mission to help customers proactively manage cybersecurity risk, without slowing down business. Our holistic program offerings are tailored to your organization's needs. Through advisory services, training, and SD Elements, our award-winning policy-to-execution platform, we set you up with all of the resources and tools your organization needs to develop secure software. Security Compass serves some of the world's largest enterprises, including 16 of the largest financial institutions in North America, South America, and Asia, as well as 4 of the largest tech companies in the world.

1.888.777.2211
info@securitycompass.com
www.securitycompass.com

@SECURITYCOMPASS
SECURITY COMPASS

OFFICES

GLOBAL HEADQUARTERS
1 Yonge Street
Suite 1801
Toronto, Ontario
Canada M5E 1W7

TORONTO
390 Queens Quay W
2nd Floor
Toronto, Ontario
Canada M5V 3A6

NEW JERSEY
621 Shrewsbury Avenue
Suite 215
Shrewsbury, New Jersey
USA 07702

CALIFORNIA
1001 Bayhill Drive
2nd Floor
San Bruno, California
USA 94066

INDIA
#4.07
4th Floor, Statesman House
Barakhamba Road, New Delhi
India 110001