WHITEPAPER

Understanding the Developer-centric Threat Modeling Process

Security Compass

Table of Contents

Introduction

This document describes the Security Compass developer-centric threat modeling process and how companies can implement DCTM using the latest version of the SD Elements platform. It outlines a high-level process flow, stakeholders, artifacts, and how it aligns with the four key questions of the Threat Modeling Manifesto. It also contrasts the DCTM approach to the legacy threat modeling approaches.

Because threat modeling implementation can differ substantially between organizations, this document is designed to educate stakeholders on DCTM in general. It is not meant to be a detailed process flow document for a specific organization. Organizations should tailor this process to fit their specific level of application security maturity, resources, and processes.

Process Overview

The threat modeling community has produced a great deal of research, insights and best practices over the years that are valuable and adaptable to modern software development practices. The process in this document represents an evolution as legacy software threat modeling best practices have merged with modern software development team practices.

The DCTM process was designed so software development teams can introduce threat modeling to their development cycle without requiring the expertise of an application security expert. For organizations that have dedicated application security teams to support development, it relieves application security experts from tedious tasks and allows them to focus on sophisticated attacks and threats or supporting and educating teams as needed. In most cases, organizations will use the application security expert-assisted processes for the most critical systems and apply a lighter, developer-led model for all other applications.

DCTM can be divided into two distinct process flows:

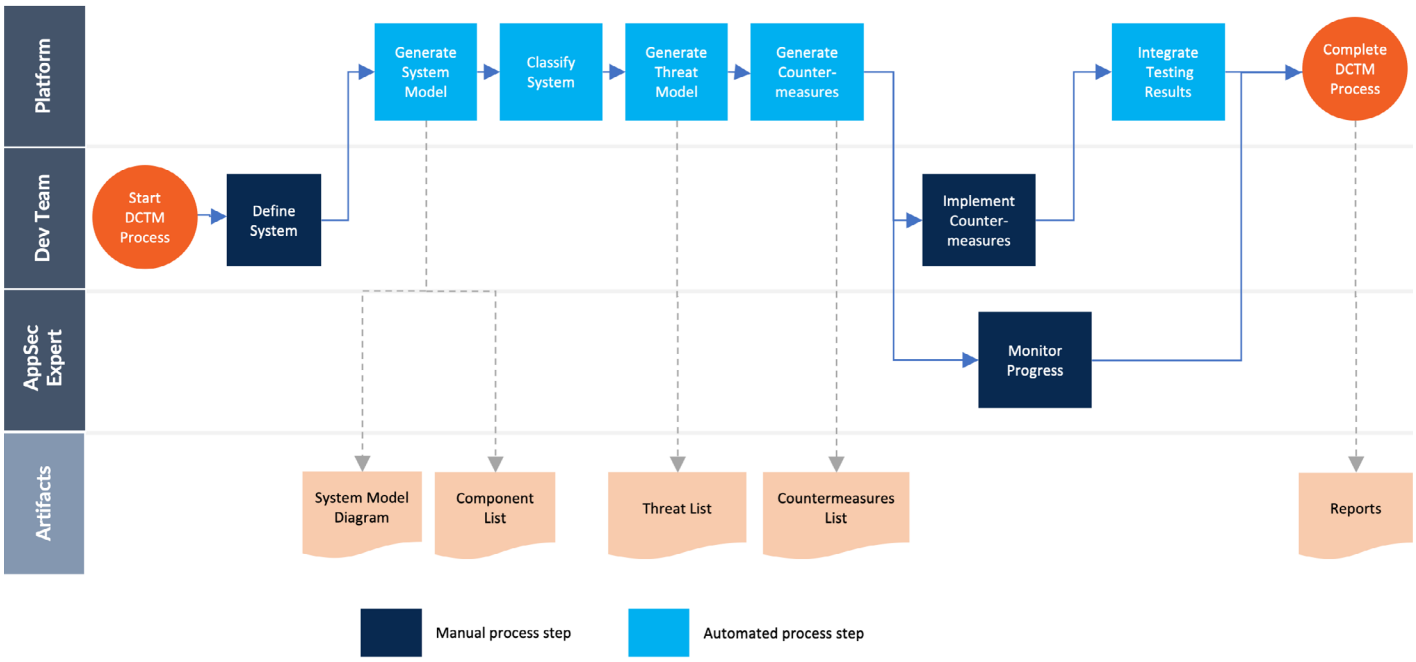
- **Machine assisted:** Software developers use the DCTM process without the direct involvement of an application security expert

- **Application security expert assisted:** Application security experts are involved in the DCTM process, typically to provide insight on areas outside of common threats and countermeasures produced by the DCTM system.

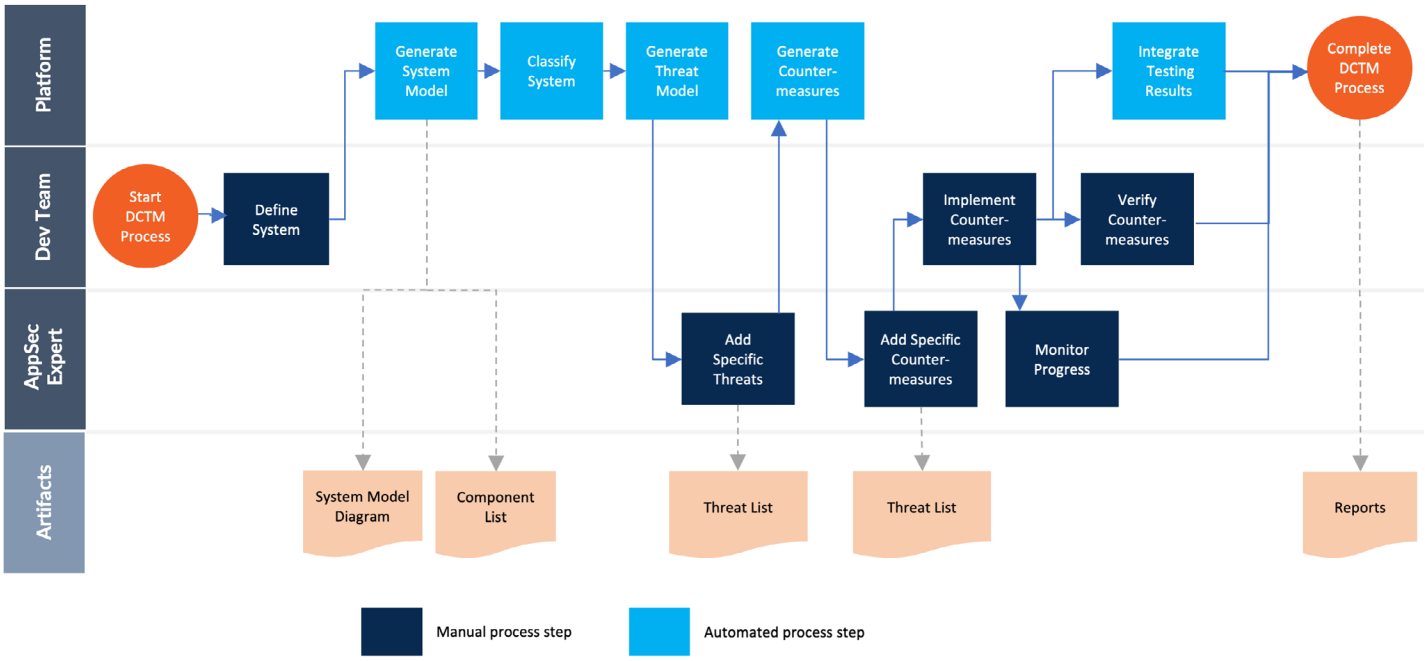
We also compare this process with manual, legacy methods.

Process Diagram

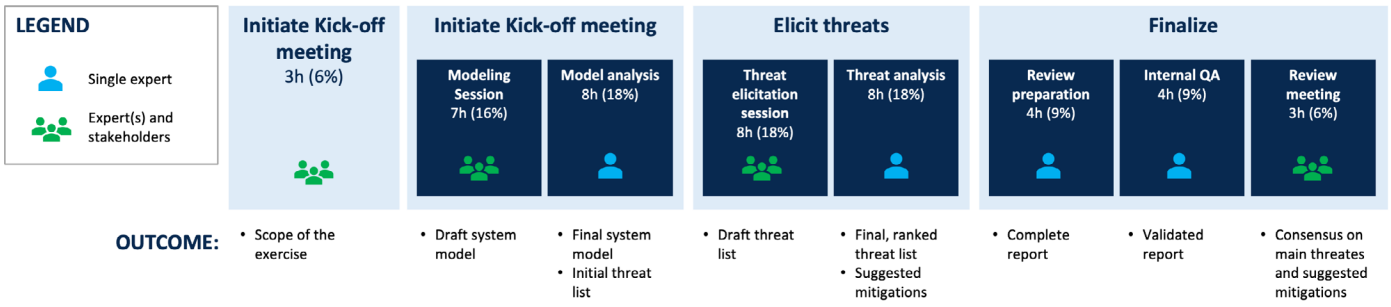
Machine Assisted DCTM Process Flow



AppSec Expert-Assisted DCTM Process Flow



Manual, Legacy Threat Modeling Process Flow



Source: https://lirias.kuleuven.be/retrieve/565533 , "Threat modeling: from infancy to maturity"

Stakeholders

For the sake of simplicity, we will keep the description of this process focused on the key stakeholders in most threat modeling activities we observed. Team names, job titles, roles and responsibilities varied across organizations so we've defined general personas, but they may apply to a number of roles or functions at an organization.

Key Stakeholders:

- **Software Development (Dev) Team:** Refers to a team of cross functional roles that is formed to build and deliver a technical product or service. Teams may be permanent scrum teams or temporary project teams which often include Project and Product Managers, Software Developers, QA, and Testers, UX Designers and Team leads.

- **Application Security (AppSec) Expert:** A technical role as a security practitioner with a primary focus on identifying and mitigating security and compliance risk for products, systems and services that are developed by their organization. Some roles that fit this persona could be Threat Modeler, Application Security Engineer, Security Architect, etc.

- **Threat Modeling Development Platform:** Refers to tools or cloud services used by software development teams that may be configured to automatically perform a task or action as part of the threat modeling process. In some cases to replace the action of another Persona.

Other Stakeholders:

- **Risk Executive:** An executive or key stakeholder of the product or asset that teams are focused on keeping secure and compliant. This may include a Chief Security Officer, Chief Risk Officer, VP of Application Security etc.

- **Technology Executive:** An executive or key stakeholder of the technology teams that build and deliver a key product or asset that needs to be secure and compliant. This may include a Chief Technology Officer, VP of Development or Engineering, etc.

Artifacts

The DCTM process creates artifacts or tangible documents at various stages. These artifacts can be used to satisfy security, compliance and audit evidence requirements.

- **System or Component Model Document:**
  A structured document that is a representation of a system, describing the components and their relationships in the system architecture. A system model should be a machine readable document that can also be presented in a human readable format. For example, the system model may be documented in JSON or XML but visualized as a component list, an architecture or data flow diagram.

- **Project Classification Rules:**
  A set of rules that automatically categorizes an asset, system or application based on the potential risk that it represents to the business.

- **Threat Report:**
  A report of all applicable security threats to the system.

- **Countermeasures Report:**
  A report of all relevant countermeasures to threats and applicable compliance controls to the system.

- **Other Reports:**
  Teams that are threat modeling should understand the objectives of the exercise and have a clearly defined business objective or measure of success. Reports and dashboards are used to help teams measure the impact of their threat modeling efforts and provide stakeholders with visibility into the system risk. For example, a report might show the compliance status of a given system to NIST 800-53, the number of critical vulnerabilities not addressed in an application, or the changes in monthly quantifiable risk scores for an application portfolio.

Process Steps

This section describes each process step in detail. In order to help explain how the process works and highlight the difference with legacy threat modeling, we introduce an example scenario where an application security and development team wants to threat model an application.

1. Generate a Machine Readable System Model Document

DCTM Approach

Estimated time (machine assisted process): 15-60 minutes
Estimated time (application security expert assisted process): 1 hour

The first step in identifying threats to a system is to gain a basic understanding of the system itself and how it works. A system model is a way to describe or represent a system and usually takes the form of a diagram that is easy for technical teams to understand.

Legacy Approach

Estimated time: 18 hours

In a legacy threat odeling process, application security experts collaborate with software development teams to document a system model that describes the major technology components a product is built on and how they interact. This can be a very time-consuming process with lots of back-and-forth between application security and software development teams. The output is usually an architecture or data flow diagram drawn manually using a whiteboard or digital drawing tools. It annotates critical data or assets and trust boundaries to identify who controls what. Supporting artifacts may also include sequence diagrams, process flows, API contracts, and sub-system or component details.

DCTM vs. Legacy

- DCTM uses system properties and knowledge bases to greatly reduce the time spent composing a system model from 18 hours to < 1 hour using the automated process.
- The DCTM approach is consistent and repeatable. The legacy approach yields inconsistent results depending on the knowledge of the practitioners.
- DCTM does not **require** an application security expert to be involved in this step, whereas legacy threat modeling does. This means in addition to elapsed time, legacy threat modeling can introduce project delays when application security experts are unavailable.

2. Classify System

DCTM Approach

Estimated time: Instant

In DCTM, it's important to understand the inherent risk of a system before further threat analysis. In many organizations, an Internet facing web application with personally identifiable information has a much higher inherent risk than an internal site. DCTM allows an organization to specify pre-determined risk classification for a system. That classification can then be used to determine what degree of risk should be accepted in the threat model, and how extensive the countermeasures should be. Risk classification can also identify the most critical systems, where an application security expert could provide greater support to the development team that develops that system.

Legacy Approach

Estimated time : N/A

Legacy threat modeling does not feature a classification step. For this reason, organizations typically choose to only threat model the highest risk systems. The determination of whether or not an application should be threat modeled is itself a potentially time consuming process, often requiring manual analysis of free-form text fields to arrive at a conclusion.

DCTM vs. Legacy

- The legacy approach does not have a classification step. Organizations must determine, outside of the threat modeling process, whether or not threat modeling is required for a given system.

3. Generate Threat Model

DCTM Approach

Estimated time (machine assisted process): Instant
Estimated time (application security expert assisted process): 4 hours

A threat model takes the system model and adds a layer of threats and countermeasures for components that introduce risk. When structured and presented correctly it becomes a key point of communication and collaboration between the application security experts and the development teams. It can also be an important artifact that makes it easier to integrate countermeasures into existing development workflows.

In a developer-centric threat modeling process where you have a machine readable system model as your input, you should be able to generate a baseline threat model based that identifies the common threats associated with the components in your system model. To accomplish this you will need to create or use a threat library that maps known threats to common system components. It is also important to have the risk policy or specific security and compliance requirements defined at or before this step. This will help filter and prioritize the threats and risks that are most applicable for the business context of your organization. (While it is possible to complete this without the business context provided by a risk policy, the resulting output of threats and countermeasures may be very large and difficult to ingest into developer workflows.)

The process will automatically generate a baseline threat model but it also allows for manual updates and adjustments based on further analysis from application security experts or the technology team.

Legacy Approach

Estimated time: 16 hours

In a legacy threat modeling process the application security experts will analyze the system model and identify threats using approaches like STRIDE, community research like OWASP Top 10, and spend time brainstorming different threat actors, threat vectors or novel threats for their situation. In some cases they may use qualitative or quantitative methods to assess and estimate the risk associated with each threat and assign a score, rating or monetary value. The resulting list of threats and their classifications is usually shared with key stakeholders such as risk and technical executives or software development teams.

DCTM vs. Legacy

- DCTM generates a threat model instantly, and experts can add to the model if/when they have capacity to do so. Legacy threat modeling has an involved collaborative approach that requires several meetings.

- DCTM does not require application security experts and software developers to spend time on threats that have already been captured by the system, whereas legacy requires practitioners to determine all the relevant threats themselves.

- DCTM ensures consistent elicitation of threats, whereas the outcomes for legacy systems differ depending on the skills and expertise of the practitioners involved.

- DCTM does not **require** an application security expert to be involved in this step, whereas legacy threat modeling does. This means in addition to elapsed time, legacy threat modeling can introduce project delays when application security experts are unavailable.

4. Recommend Prioritized Countermeasures to Implement

Estimated time (machine assisted process): Instant
Estimated time (application security expert assisted process): 4 hours

In developer-centric threat modeling the objective is to automatically categorize and prioritize the list of security threats and compliance risks and recommend the actions to be taken for each. A risk policy can specify the rules and scope for a threat model that help identify the countermeasures that should be assigned to a development team.

Most legacy threat models produce an overwhelming list of threats that software developers would have difficulty prioritizing without the necessary application security background. It's important that the teams implementing the countermeasures get a very focused list that identifies the most critical threats in a system that a development team can control. Once the priority threats and countermeasures have been identified, the team needs to understand what they should do about them. Developer-centric threat modeling recommends that a countermeasure includes:

- A documented **requirement** or user story written in developer friendly language that clearly identifies the threat and the steps needed to mitigate it.

- Where applicable, **sample solutions** such as source code, configuration templates or recommended tools that can help implement the provided requirements.

- An **acceptance test** that can be run to verify whether a threat has been mitigated based on the implementation of a countermeasure.

- **Training** material that helps a development team understand the threat and demonstrates what is needed to implement a countermeasure and how to avoid making the same mistake again. This may be in the form of video tutorials, written documentation, sample code, or hands-on exercises.

For common vulnerabilities there are existing databases or services that provide a good starting point for countermeasure requirements. You can also leverage existing static application security testing (SAST) and dynamic application security testing (DAST) tools to automatically run tests for vulnerabilities associated with threats discovered. There are plenty of excellent free and paid resources that provide advanced training and education for software development teams who need to adopt secure coding practices. But while this information is readily available, it requires a significant investment of time and resources to:

1. **Match** the security requirements for components with the appropriate application security tests and training material,

2. **Deliver** the requirements, tests and training directly to software developers in the tools and process they already use,

3. Keep everything relevant and **up to date** as both threats and system components continue to change on a weekly basis, and

4. **Track** the implementation status of each countermeasure to determine whether the risk has been mitigated.

Automating these actions is critical to the success in scaling threat modeling across the enterprise. It is heavily dependent on the artifacts being machine readable, and portable across other development and application security tools used at your organization. In most situations it may require a common connecting layer that makes it easy to push and pull data across tools and automatically trigger activities in the process.

Legacy Approach

Estimated time: (Included in previous step)

This is often the step where threat modeling breaks down and it becomes challenging for application security and software development teams to maintain consistency and scale as part of the process.

In legacy threat modeling, the application security experts will categorize the list of threats and identify whether the risk needs to be mitigated, eliminated, transferred (to another component) or accepted (within reason). The result of this exercise is usually a list of categorized threats and may also include the recommended countermeasures to implement, that should address the risk identified. The output format is often a spreadsheet of threats and countermeasures that gets shared with the software development teams and scheduled in upcoming development work.

DCTM vs. Legacy

- DCTM generates countermeasures instantly, and experts can add to the model if/when they have capacity to do so. Legacy threat modeling involves a collaborative approach that requires several meetings.

- DCTM automatically determines the countermeasures that are already implemented within other system components, minimizing the time spent by development teams. Legacy threat modeling does not have a mechanism to do this and therefore development teams must spend time determining which countermeasures are in-scope for them.

- DCTM includes compliance controls. Legacy threat modeling does not. Software development teams seeking regulatory or standards compliance must undergo a separate set of processes to integrate compliance into their design.

- DCTM ensures consistent selection of countermeasures, whereas the outcomes for legacy systems differ depending on the skills and expertise of the practitioners involved.

- DCTM does not require an application security expert to be involved in this step, whereas legacy threat modeling does. This means in addition to elapsed time, legacy threat modeling can introduce project delays when application security experts are unavailable.

5. Implement and Test Countermeasures

DCTM Approach

Estimated time:
Varies based on application security and compliance requirements for each system

Once development teams receive their countermeasures, the work usually enters a prioritized backlog along with features and other non-security related work. It is at this stage that application security professionals play a critical governance role in ensuring that the relevant security tasks are given appropriate priority. DCTM allows application security teams to periodically track progress of development teams and enquire about results when teams progress slower than anticipated. DCTM systems can integrate with development tools like JIRA to seamlessly deliver countermeasures to software developers. The countermeasures could include relevant training and code samples to facilitate implementation. Furthermore, integrations with code scanners can be used to test for the presence or absence of threats.

Legacy Approach

Estimated time:
N/A - not part of legacy threat modeling

Legacy threat modeling does not include an implementation step.

DCTM vs. Legacy

- Legacy threat modeling ends at the generation of a threat model report. There is no logical link to ensure the countermeasures have been implemented, nor is there a mechanism to tie application security testing back to the threat model.

- DCTM links threat modeling with application security testing and tracks implementation progress. DCTM goes beyond a table-top exercise of eliciting threats, and ensures that the controls are implemented within the system.

6. Monitor and Measure Results

DCTM Approach

Estimated time:
Varies based on application security & compliance requirements for each system

One of the big benefits of using machine readable threat models with a standard structure, as specified by DCTM, is that it is easier to manipulate data for status tracking and reporting purposes. DCTM tools also allow for better overall governance of the application security program through reporting.

The most successful organizations define their desired business outcomes right from the start, and configure their threat modeling process in a way that makes it easy to report the current state vs. the desired state. This might be measuring the number of threats addressed each release or the time to deliver new features while maintaining policy compliance.

With developer-centric threat modeling, we recommend that you define your success metrics up front and ensure that you are logging or capturing data from the integrated tools so you can automatically generate reports that reflect the security posture of your system. We stress the importance of having access to the right data in the right format more than the style or content of any specific report, because threat modeling is only one tool teams use to mitigate their risk, and in most cases teams will want to combine data and metrics from a number of application security tools and processes to produce a dashboard of reports that measure the success of individual programs and the combined efforts overall.

Business decisions around application security investments rely on the value of meaningful data. DCTM provides this value by reducing the overhead (time and cost) of collecting vulnerability data so that business leaders can make informed decisions more accurately about which countermeasures to invest in. The countermeasures will be defensibly tied to the appropriate high priority risks. From an investment point of view, that has a direct impact around costs of cyber insurance, for example. Lowering the cyber insurance costs (or preventing overallocation) can help manage resources more prudently to generate greater business value elsewhere.

Legacy Approach

Estimated time:
N/A - not part of legacy threat modeling

Much of the traditional threat modeling material focuses on the modeling and analysis to identify threats and countermeasures but is light on guidance around measuring the success of the program or processes put in place. It recommends that you should test the implemented countermeasures and update the status for further review, but in practice this usually amounts to a long list of issues with a checkbox marking them as Complete or Incomplete. In situations where compliance requirements are included the success may also be measured by successfully passing a regulatory audit. These are typically published as PDF reports and shared across teams.

DCTM vs. Legacy

- Legacy threat modeling correctly focuses on the step that asks “Did we do a good job,”’ but is light on guidance for completing this step. It typically suggests that QA and testing will track whether countermeasures have been implemented, or it relies on other processes like penetration testing, although it’s difficult to attribute that directly back to your threat modeling activities.

- Legacy threat modeling is usually performed on a periodic basis (ie. annually) and often starts from scratch when subsequent threat models are produced.

- DCTM puts more emphasis on producing artifacts in a structured, machine readable manner that makes it easier to monitor changes as software evolves and compare against previous data such as risk level, number of threats identified, and time to complete.

- The integrated nature of DCTM (connections with development platforms and application security tools) makes it easy to consolidate data from multiple sources and provide a more comprehensive view of system risk across the development lifecycle.

- The scalable nature of DCTM makes it easier to expand the scope of coverage, both in classifying asset risk as well as providing broader views of risk across the organization, with the flexibility to produce a variety of reports at the component level, project level and team level.

SampleScenario: DCTM vs. Legacy

This section provides a sample scenario that highlights the differences between application threat modeling using DCTM vs legacy threat modeling .

SampleScenario Description

A software development team at Acme, Inc. needs to threat model their Software-as-a Service (SaaS) product called WidgetsRUs. The threat model is intended to satisfy the requirements of one of their security-sensitive customers who have asked for a threat model as part of the procurement process. Moreover, Acme, Inc. must also comply with the European GDPR, California Privacy Directive, and ISO 27001 based on customer requirements.

WidgetsRUs is a .Net web application hosted on Microsoft Azure with a microservices architecture that uses a variety of third party libraries and services, including integration with Single Sign On (SSO) providers.

Legacy Approach

Initiate

An application security expert at Acme, Inc. schedules a three hour kick-off meeting with three software development leaders and two application security experts. Based on availability, the first time they can all meet for that long is in two weeks. During the meeting, the application security experts outline the goal of the threat modeling exercise, the scope, and fields questions from the software developers. GDPR, California Privacy Act, and ISO 27002 compliance are explicitly excluded from scope because the process is already very involved and none of the compliance experts are knowledgeable in threat modeling.

Model

Over three weeks, various application security experts and WidgetsRUs software developers meet several times to mutually agree upon a visual representation of the system, including the system architecture, trust boundaries, and key data flows.

Application security experts then study the resultant system model and start to draft an initial set of threats using the STRIDE taxonomy and their own knowledge.

Elicit Threats

The application security experts then meet with the software developers to present their draft set of threats, and through a series of meetings over another two week period, elicit other potential system threats based on the STRIDE taxonomy.

They then call a series of meetings to decide upon relative prioritization of the threats, followed by agreed upon mitigations.

Finalize

The application security experts create a draft report detailing the agreed upon system model diagram, threat model, and mitigations. Software development and application security stakeholders review the draft report and add comments and clarification. The report is then finalized and made ready for audit.

Implementation

The legacy threat modeling process has completed at this point. The development team decides on which mitigations they will scope into their current release, and there is no specific traceability for the application security team to follow. Instead, the application security team relies on penetration testing and static analysis testing as a gating process prior to release. There is no tie back to the threat model document, which now only exists as an audit artifact.

DCTM Approach

Define System and Generate System Model

A software development leader at Acme, Inc. logs onto the DCTM platform and describes the tech stack and compliance requirements of WidgetsRUs to the best of their knowledge. The software development leader has the option to build a visual diagram showing the architecture and data flows. The diagram is pre-populated with the components the software development leader described. Alternatively, they can skip directly to having the system model generated. In either case, system model generation happens instantly after the system properties are defined.

Classify System and Select Risk Policy

Based on the properties of the system, the DCTM platform automatically classifies the system as critical risk and applies the appropriate risk policy. This means that a number of security threats and countermeasures, as well as compliance controls, will be in-scope for WidgetsRUs that would not be in scope for some of Acme, Inc.'s other systems that have less sensitive data.

Generate Threat Model

The platform automatically generates a set of relevant threats, which are prioritized according to the profile of the system. Users can log into the platform at any time to download reports that describe the threats as well as the system model.

Recommend Prioritized Countermeasures to Implement

The platform then generates a set of relevant security countermeasures and compliance controls that are normalized and prioritized based on the risk policy. Furthermore, the system automatically identifies those countermeasures that are satisfied by system components, such as authentication requirements being satisfied by the SSO system. The software development lead connects the DCTM platform to their tracking system, JIRA, and the remaining work is automatically integrated into the backlog as a series of tickets. Each task has a requirement, relevant test, and contextual training to understand the requirement.

Implement and Test Countermeasures

Software developers work on implementing the remaining tasks in JIRA. When they complete the work, the status automatically synchronizes with the DTCM platform. Optionally, the lead software developer integrates the results of static analysis tools to show correlation between the countermeasures and application security testing.

Monitor and Measure Results

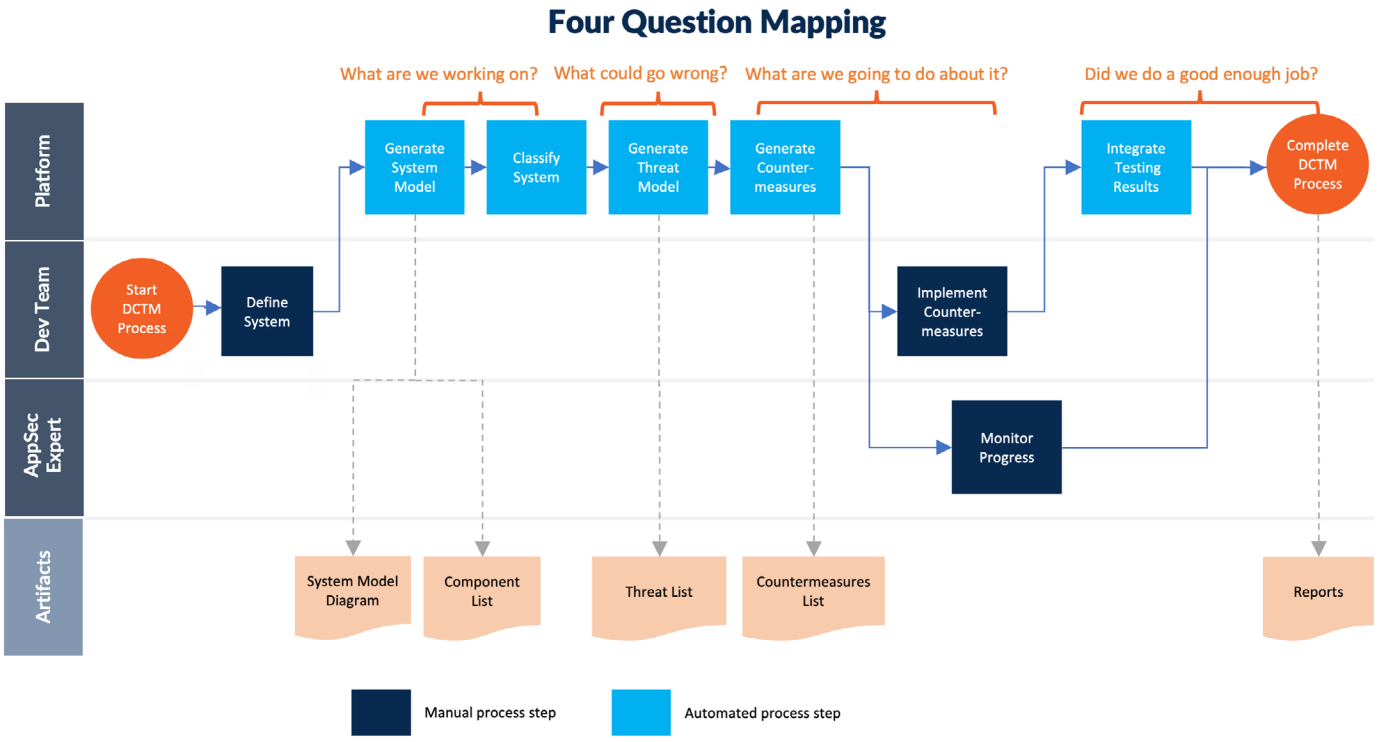
The application security team oversees the entire process from the DCTM platform. When the software development teams appear to not show progress, they can follow up and request progress updates or discuss compensating controls. They can also use the progress to report product security progress to the senior executive team.

Mapping DCTM to the Threat Manifesto Four Questions

One of the best ways we've seen the threat modeling process described is from the Threat Modeling Manifesto, because its simplicity makes it easy to understand for almost any role. It states that when you threat model, you are asking four key questions:

1. **What are we working on?**
2. **What can go wrong?**
3. **What are we going to do about it?**
4. **Did we do a good enough job?**

The steps in DCTM map cleanly to the Four Questions:



Summary of DCTM Process vs. Legacy Threat Modeling Process

Legacy threat modeling processes rely on human experts to perform analysis. Research shows that these processes took 40+ hours to complete with several manual steps. DCTM takes the approach of starting first with automation and then allowing manual analysis when necessary.

Contrasting Legacy to DCTM

Legacy TM	Initialize	Model	Elicit Threats	Finalize	Total	Implement, Monitor, Report
	3 hrs	15 hrs	16 hrs	11 hrs	47 hrs	Not in scope
Automated DCTM	Define System	Classify System	Generate Threats & Countermeasures		Total	Implement, Monitor, Report
	1 hr	Instant	Instant		1 hr	Varies
Expert Assisted DCTM	Define System	Classify System	Generate Threats	Generate Countermeasures	Total	Implement, Monitor, Report
	1 hr	Instant	4 hrs	4 hrs	9 hrs	Varies

Security Compass

About Security Compass

Security Compass, a pioneer in application security, enables organizations to shift left and build secure applications by design, integrated directly with existing DevSecOps tools and workflows. Its flagship product, SD Elements, helps organizations accelerate software time to market and reduce cyber risks by taking an automated, developer-centric approach to threat modeling, secure development, and compliance. Security Compass is the trusted solution provider to leading financial and technology organizations, the U.S. Department of Defense, government agencies, and renowned global brands across multiple industries. For more information, please visit www.securitycompass.com.

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
600 California Street
San Francisco, California
USA 94108

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