WHITEPAPER

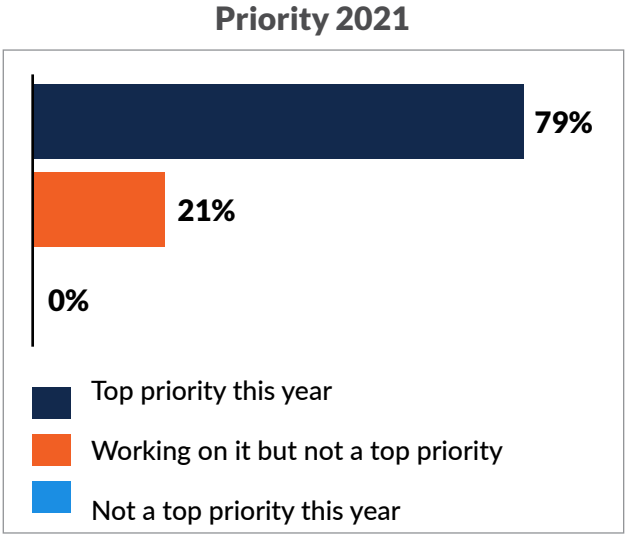
Why Traditional Threat Modeling Fails and How to Get it Right

Security Compass

Why threat modeling is important

Security professionals know they can accelerate development and reduce unnecessary security rework through threat modeling. Threat modeling examines an application's technology stack, deployment environment, and applicable regulatory requirements to anticipate security issues. This allows developers to build appropriate controls as part of the normal development cycle. It is unsurprising then, that threat modeling is a top priority for organizations.

Traditional threat modeling requires development and security resources to diagram an application, establish trust boundaries, and assign threat mitigation controls. While valuable, this process can take weeks. In the rapidly changing development environment found in most organizations today, traditional threat modeling faces a multitude of challenges.



The 2021 State of Threat Modeling: An interactive e-book publication

Why traditional threat modeling fails

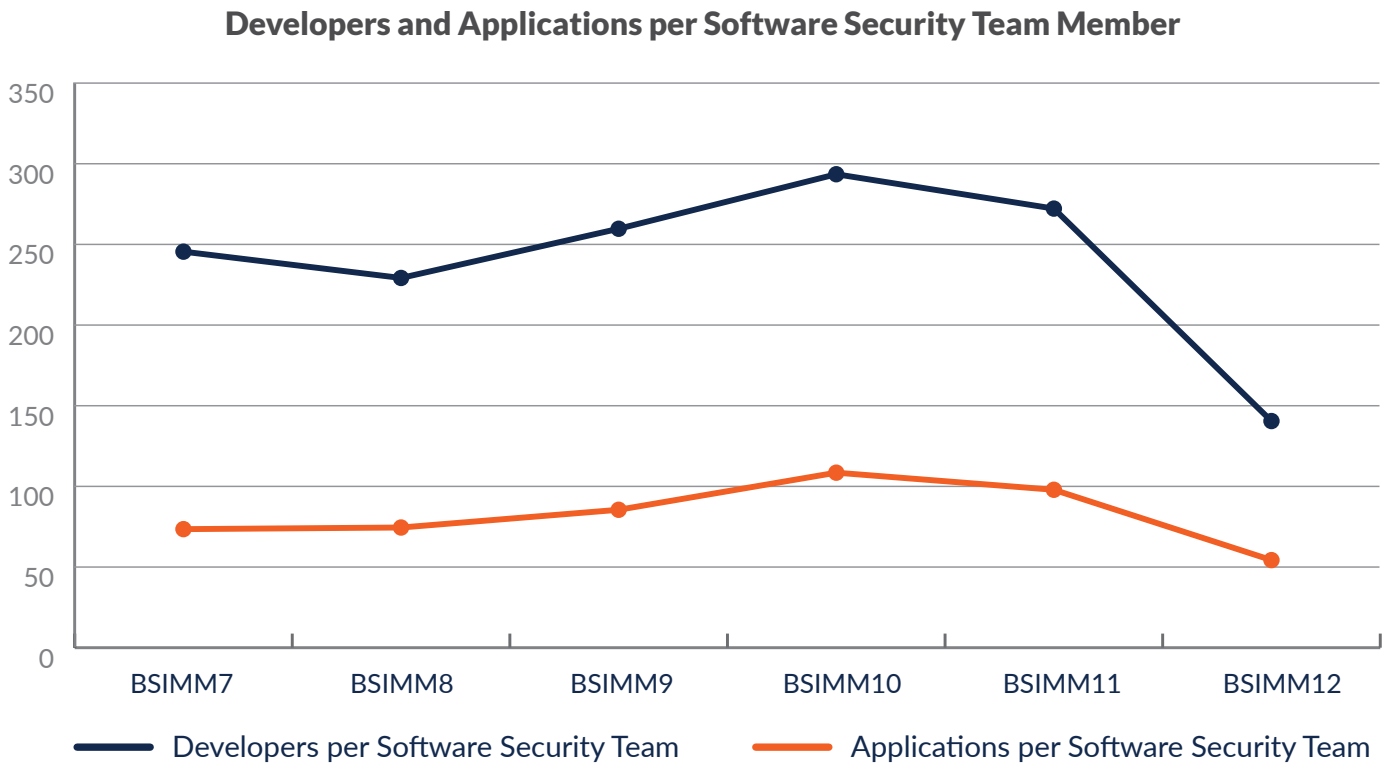
The threat modeling process has value for security personnel and business owners. Increasingly hostile cyber criminals can disrupt operations, damage brand reputation, and bring increased regulatory scrutiny. Testing for security issues late in the development process results in additional blockers in the product release pipeline.

Popular threat modeling techniques like STRIDE and PASTA are largely manual. A few threat modeling tools have been digitized (from paper- or whiteboard-based exercises) and incorporate some automation capabilities. However, business owners, technical managers, and security personnel still face several challenges when implementing manual threat modeling initiatives.

Scalability

Scalability is one of the common challenges with traditional threat modeling. Security expertise remains scarce—even in organizations with formal software security teams. The 2021 BSIMM survey found an average ratio of one Software Security Group member per 140 software developers. Looking at the findings another way, each software security team member supported over 50 applications.

With ratios like this, it is obvious that organizations lack the resources to devote senior personnel for multi-day exercises for each project or application. As a result, only the most critical applications are modeled prior to projects beginning. This leaves the majority of an organization's application portfolio potentially vulnerable to attacks, and ignores legacy applications already deployed.



The 2021 BSIMM survey

Completeness and control implementation

By design, manual threat modeling typically focuses on a subset of security threats and mitigations for software and its environment and falls short on detailing and prioritizing technical steps that software development teams can execute. Similarly, mitigations generated by semi-automated threat modeling tools are not prescriptive enough for developers. This typically requires that security experts support developers' post-threat analysis to implement the mitigations.

Consistency

The thoroughness and effectiveness of a manual threat modeling exercise is dependent on the experience, skill, and judgement of those conducting it. Different people building the threat models have different expectations of what a threat is, what the threat model should look like, and how the threats should be ranked. This inconsistency has a downstream repercussion on aligning on the biggest threats and therefore which threats development teams should prioritize. A lack of standardized processes leads to inconsistent outputs. Incomplete and inconsistent threat and control identification results in missed threats and non-standard controls. Changing regulatory standards further complicates assigning correct controls, even when organizational policies are in place.

Validation and auditability

The validation of appropriate risk mitigation controls should be part of every project. Unfortunately, most threat modeling solutions do not integrate well with issue trackers (to streamline the assignment of recommended controls to development teams) or scanning tools (to verify that mitigations have been implemented). Verifying controls using spreadsheets and shared documents is inconsistent, and updates by email are subject to misinterpretation, providing poor evidence of compliance with corporate policies and regulatory standards. Manual methods also make it difficult to provide management and auditors with a clear picture of an organization's security profile across all projects.

Complexity

Manual threat modeling was adequate for large codebases with a well-understood and static architecture. Today's applications do not follow this model. The move to the cloud, adoption of rapid development methodologies and the need to adapt quickly to changing market demands mean new features and functionality are constantly added. Increased use of microservices and layered architecture results in an ever-changing threat landscape. Manual threat models requiring days or weeks of effort no longer work.

Manual threat models reflect the original intent of a project's design and implementation—a snapshot in time. As a project evolves, the original threat model quickly becomes obsolete. Integrations with other applications can alter the threat landscape. Moving from an on-premises deployment to the cloud – or changing cloud providers – requires updated threat models. With a single application this can be difficult. With hundreds or thousands of applications, manual threat models are impossible.

In a survey of over 7,000 executives, only 3% reported using a single private or public cloud in 2021, down from 29% in 2019.

IBM Institute for Business Value

IBM Study: C-Suite Executives Declare One-Vendor Approach to Cloud is Dead

Developer pushback

Software development teams are incentivized to deliver a certain set of features and functionality by a specific date. Development friction occurs when there are dependencies on other teams to achieve the objectives of the project. Dependencies can create bottlenecks or increase the time and effort required to complete the activities.

In threat modeling exercises, these dependencies occur during the information gathering phase when development input is required to scope out threats in the system under assessment, or during the implementation of recommended mitigations when security guidance is required.

Gartner® reported that "three out of four software engineering teams report that they experience development friction, defined as unnecessary time and effort they must exert to achieve their objectives."[1] Further, almost half of the software engineering personnel surveyed reported experiencing friction in meeting architecture and security requirements.

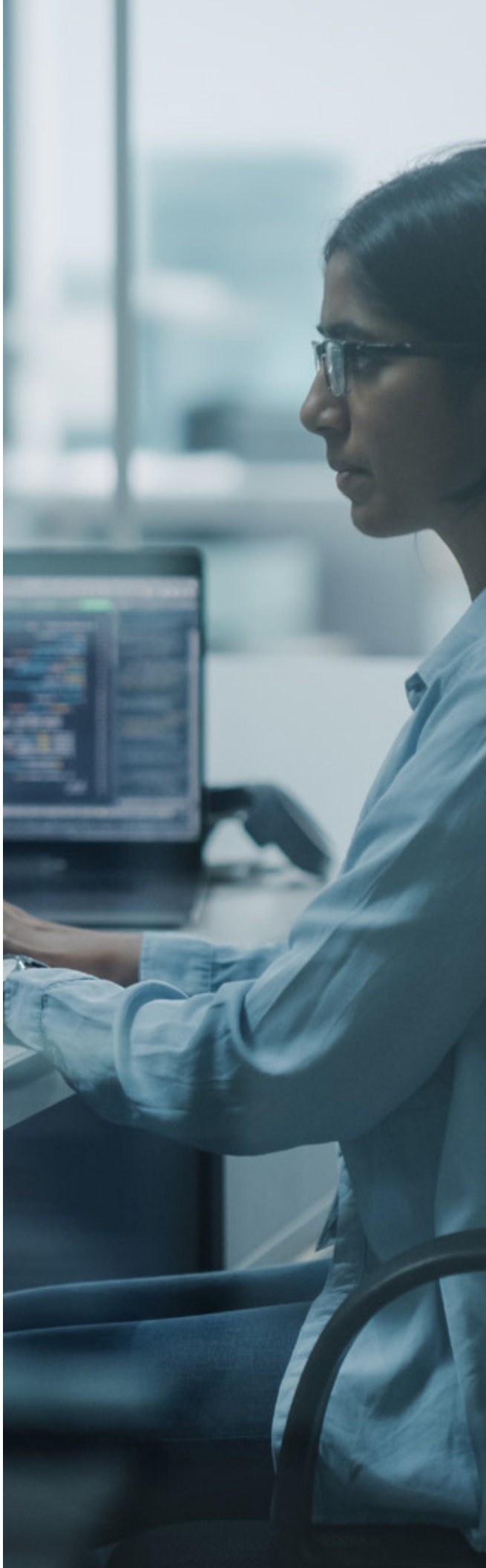
How to threat model in a complex environment

To reduce cybersecurity risk at scale, organizations need a different approach to modeling software risk that tightly integrates with product workflows and empowers product teams to deliver secure products at high velocity. In a rapidly changing environment, manual and semi-automated processes cannot keep up.

Automation of the threat modeling process with SD Elements provides this speed and security, along with scalability, consistency, and the ability to quickly understand the security profile of a single project or an entire portfolio.

Learn more

Contact us today to discuss your current threat modeling challenges and learn more about how SD Elements can help.



1 Gartner, "Reduce Friction to Boost Software Engineering Team Productivity", Applications and Software Engineering Research Team, Published 25 May 2021. GARTNER is a registered trademark and service mark of Gartner, Inc. and/or its affiliates in the U.S. and internationally and is used herein with permission. All rights reserved.

SecurityCompass

Security Compass

Go Fast. Stay Safe.

Security Compass, a leading provider of cybersecurity solutions and advisory services, enables organizations to build more secure software faster. Our flagship product, SD Elements, helps automate significant portions of proactive manual processes for security and compliance that improves time to market for new technology. Security Compass is the trusted solution provider to leading financial organizations, technology enablers, and renowned global brands. Security Compass is headquartered in Toronto, with offices in the U.S. and India. Follow Security Compass on Twitter @ securitycompass or visit us at securitycompass.com to learn more.

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