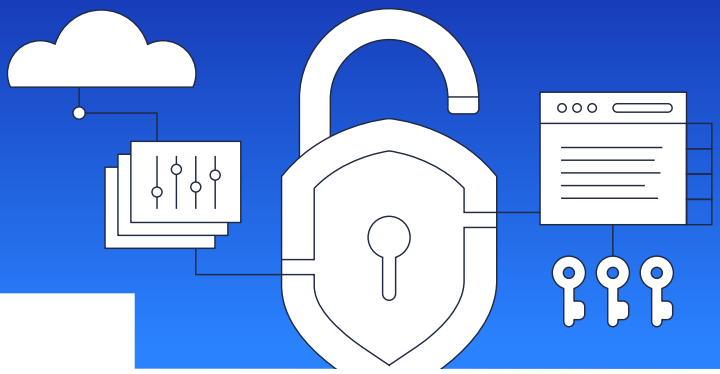
SECURE CODE WARRIOR



WHITE PAPER

Secure by Design: Defining Best Practices, Enabling Developers and Benchmarking Preventative Security Outcomes

Executive Summary

It has been two years since the United States government's Cybersecurity & Infrastructure Security Agency (CISA) released its comprehensive Secure by Design guidelines, signaling a watershed moment affecting software manufacturers. For the first time, there was visible, top-level support for raising the standards of software quality and security, with a push towards vendor—as opposed to end-user—accountability for ensuring code shipped free from vulnerabilities.

However, consensus on the real-world implementation of these principles at the enterprise level has proved elusive. Among security professionals, there does not appear to be a consensus on what constitutes Secure by Design, much less a standard pathway being followed to achieve it. Secure Code Warrior interviewed enterprise security professionals focusing on building software, diving deep into their current approaches to Secure by Design principles, including how it is being implemented into their current security posture and Software Development Life Cycle (SDLC). It became apparent that there was no widely accepted standard for implementing CISA's guidance, nor were there active benchmarks to determine successful rollout. This must be corrected rapidly if we, as an industry, are to reap the benefits of heightened security accountability and software quality.

In this research paper, Secure Code Warrior co-founders, Pieter Danhieux and Dr. Matias Madou, Ph.D., **will reveal key findings from over twenty in-depth interviews with enterprise security leaders including CISOs, a VP of Application Security, and software security professionals,** investigating common challenges, interpretations of best practices, and the role precision data and benchmarking can play in industry-wide alignment with a viable Secure by Design strategy.

Authors:

Pieter Danhieux, CEO & Co-Founder, Secure Code Warrior

Dr. Matias Madou, CTO & Co-Founder, Secure Code Warrior

Contributors:

Chris Inglis, Former US National Cyber Director, now Strategic Advisor to Paladin Capital Group

Devin Lynch, Senior Director, Paladin Global Institute

KEY TAKEAWAYS

- **Secure by Design suffers from definition sprawl—ranging from strict architectural practices to broader supply chain, authentication, and access controls—diluting its effectiveness across organizations.**
- **Threat modeling has devolved into a compliance checkbox, with enterprises experimenting with AI automation to relieve human burden. The fundamental question remains unresolved: Should this be an expert specialty or a universal developer responsibility?**
- **AI coding assistants present Application Security VPs with an immediate dilemma: Should they restrict or embrace? They must quickly distinguish helpful models from harmful ones while establishing governance mechanisms before incidents occur.**
- **While AppSec teams struggle to sanitize vulnerability data, they face mounting pressure to track broader signals across development ecosystems—especially as autonomous coding agents proliferate.**

Introduction: Defining Secure by Design with Enterprise Security Leaders

Many security professionals consider the Secure by Design movement a crucible moment—or, indeed, reckoning—for an industry frequently brought to its knees by large-scale data breaches, cyber espionage. The AppSec environment has, for decades, been unable to keep pace with the deluge of software being created. CISA's push for "radical transparency" in the software development process changed the landscape. They are placing accountability on software vendors to ship secure software from the beginning. This approach has led over 250 forward-thinking companies to sign the Secure by Design pledge, and commit to adopting a higher standard of secure software design and implementation that has perhaps been acceptable to date.

Secure Code Warrior also committed to the pledge. As part of our research into structuring ideal-state Secure by Design initiatives, we conducted over twenty interviews with enterprise customers who were exploring their own secure design capabilities and uplifting requirements. It was through this research that we uncovered significant insights into the perception of enterprise-level Secure by Design initiatives, with one common thread: **No two definitions of what constitutes Secure by Design were the same.**

While some public commentators have expressed concern with the difficulty in implementing CISA's guidelines, there has not yet been an inquiry into whether the cyber industry is on the same page with how these standards should be implemented, and realistically, perhaps that is where we must begin if we seek a smooth transition to higher security standards in modern software development.

How are enterprises currently approaching Secure by Design as an in-house initiative?

Enterprise security programs share common elements of complexity, often with unwieldy tech stacks that become a source of frustration for CISOs. A need to implement factors like role-based security awareness training, in addition to specific activations of both technology and skills verification to meet compliance requirements, further complicate security efforts.

Similar setups were discussed during our interview process with more than twenty enterprise security leaders, but insights into specific Secure by Design initiatives were revealed to be somewhat fragmented.

Some commonalities discussed were that such an initiative often starts at a higher level, with identity and access management like adopting Multi-Factor Authentication (MFA) and stringent password rules. Alternatively, organizations may do away with passwords entirely in favor of something more robust and universally controllable. These are baseline measures that assist in ensuring every person in an organization has some semblance of security guardrails. Such guardrails protect account integrity and the data they may have access to as part of their role, and they are often among the first security measures implemented in software. However, this alone is insufficient to declare any application as "secure by design". Security professionals recognize this limitation clearly. The fact remains that, industry-wide, we are struggling to stick to a standard security program rollout. Such a standardized approach would be necessary to meet the expectations set by the likes of CISA. Without it, we cannot forge a future of software that is meaningfully more secure than what is created with today's variety of approaches.

In essence, the rules of engagement and performance indicators could be a lot clearer. We observed that within AppSec teams, there is certainly a common general goal to ensure security is baked into the design, rather than languishing at the end of the SDLC where it is far more expensive and time-consuming to remediate code-level vulnerabilities.

Based on our research findings, the concept of a Secure by Design initiative in the AppSec division typically consists of five elements:

1. **Threat Modeling, Design, and Security Architecture:** A structured, consistent program involving security-skilled developers and their AppSec counterparts, actively designing–at the foundational level–software that is pre-emptively built to withstand common threat vectors pertaining to that network or system. This process did not typically involve developers, but forward-thinking companies are increasingly upskilling them to participate in this crucial security process;

2. **Security Policy:** Establishing and distributing set parameters around usage of secure third-party components and tools;

3. **Paved Path Method:** Implementing leader-endorsed tools and processes to ensure easy access to recommended components, programming libraries, and guidelines to develop secure code according to policy;

4. **Cultural Transformation:** Moving beyond an isolated security champion program into a more DevSecOps-aligned practice of security as a shared responsibility across the organization. This typically involves role-based training, and developer compliance exercises that are held at least

annually. These courses ordinarily start with basic security awareness, and, if comprehensive enough, move into foundations of secure coding principles;

5. Scanning and Remediation: SAST/DAST/IAST/RASP tools are deployed, usually resulting in a deluge of false positives and negatives requiring specialist AppSec oversight, ultimately leading to a glacial process of verification of true issues and their eventual remediation.

While these steps provide a formidable framework for a security program, we know from the rising cost and potency of data breaches that they can be something of a paper tiger against a growing threat landscape, especially with the introduction of AI coding tools and increasing risks associated with nation-state actors.

Secure by Design and the Industry: There is no "Secure by Design" Company

Despite establishing a lack of cohesion in a standard, best-practice approach to a Secure by Design initiative, there is no denying that many enterprises are motivated to at least consider elements of this movement. This is especially true for those in sensitive verticals like finance and critical infrastructure. These organizations are examining how Secure by Design principles might shape the future of their software development, and some are adopting some aspects sooner rather than later. This accelerated timeline is often driven, we found, by compliance requirements rather than voluntary improvement initiatives.

In parallel, security vendors are not exactly putting on a united front, either. While many appear to be working on methods to embed Secure by Design features into their products, the approach is fragmented and less holistic than is likely to be successful. Their AppSec-specific avenues often include automated remediation solutions, AI-powered "secure by default" capabilities, or reimagined scanning and monitoring tools that have existed for some time. To that end, in the industry today, a proprietary full-service "Secure by Design" company does not exist.

Instead, different companies hold pieces of the Secure by Design puzzle, and the hunt is on to track down and fit them together into a tech stack befitting of solving the core issue of insecure software design. Threat modeling companies were the pioneering force of "secure by design" as we know it, but they ultimately failed to deliver on making applications more secure by fundamentally altering the design (or enacting lasting positive change to the approach).

The next in the stack are Application Security Posture Management (ASPM) companies. They certainly provide solid branches of a Secure by Design initiative such as cloud posture management, SAST and secrets management. Still, there are too many missing pieces to suggest effectively that CISOs can adopt this solution in isolation and render their teams compliant and their future software safe. For example, what risks can the development cohort impose on the code? This requires additional precision tooling to collect meaningful, actionable data and insights.

Other helpful emergent technologies will inevitably complement a Secure by Design initiative, such as the OWASP Application Security Verification Standard (ASVS) Project. Currently in use by some of the companies we have interviewed, this tool provides a standard for testing application and environment security controls, with the aim of preventing long-standing, frustrating vulnerabilities like cross-site scripting (XSS)

and SQL injection. The platform provides a benchmark for assessing the trustability of software, while providing key guidance to the development team on best-practice security controls to bake into their applications. This project—led by community enthusiasts—represents a likely future of helpful secure design tech, but its power is only truly unlocked in an enterprise where developer-driven, holistic and modern security programs are in place.

To that end, we must get smarter, and we must undertake a comprehensive and collaborative effort as an industry if we are going to make a tangible impact.

The Threat Modeling Trap: Is it Just a "Checkbox" Exercise?

The Field of Threat Modeling

Threat modeling has been a hot industry topic for a few years at this point, yet it remains a field in its infancy. Every security leader is analyzing the best strategy for implementing it in their security program, each with their own strong opinion on best practices, which roles should be involved, and how success can be measured.

Some companies have an excellent handle on how threat modeling supports their overall security posture, and generally, they have actionable answers in response to the following:

- *What does 'threat modeling' mean in our organization?*
- *How does threat modeling differ from—and complement—design review?*
- *When should threat modeling take place in the SDLC?*
- *How do we achieve threat modeling best practices in the context of our organization?*

The companies that can confidently navigate internal threat modeling initiatives are the exception, and they are significantly ahead of most of the industry. During our interview process, we encountered some exceptional outliers, while the rest are driven chiefly by compliance needs to roll out at least a minimum viable threat modeling program.

In fact, **2 out of 3 of those interviewed described a program that could be considered a "checkbox" exercise.**

Comprehensive threat modeling is tied very closely to CISA's guidance on Secure by Design, with the agency prescribing "tailored threat modeling" as a core Secure by Design tactic, along with other elements like Software Bill of Materials (SBOMs), code review, and Defense-in-Depth design. Almost all of these initiatives require the input—or, indeed, full execution—of security-skilled developers, and most enterprises are struggling to upskill their cohort and effectively deploy them. Threat modeling, in this regard, is no different.

During our interviews, common developer-centric themes arose, with most including basic threat modeling tasks as part of their strategy to assist developers with thinking about their design from a

security perspective, diagramming, and providing countermeasures for the bugs being tracked on their remediation list. This is a good start, and it is quite "new school" to feature developers in threat modeling initiatives, but it amounts to little more than a compliance exercise for the company, and many security professionals struggled to articulate the value to their organizations.

When executed comprehensively and consistently, threat modeling is a powerful tool that is fundamental to producing software that meets Secure by Design standards. And, while many leaders consider it a best practice, at the moment, it would appear that this is an additional area in which a standardized approach is lacking, much in the same way it does for true Secure by Design initiatives.

Systems like DREAD, STRIDE, PASTA, and others, but fortifying a go-to, developer-driven approach that is industry-wide would be beneficial to move beyond the basic compliance required by internal compliance officers.

The Threat Modeling Process

While there are different threat modeling systems, and the steps within those systems may support different goals, there is a consensus in theory and in the lived experience of security professionals that dictates we must clarify *when* to engage in the threat modeling process. This is a vital step in cybersecurity risk management, and we cannot afford to keep guessing.

As it stands, our research revealed that in the absence of standardization in this area, the scenario often occurs where one manager can tell two team members how to prepare a threat model, and those two people will still produce two completely different approaches. While they need to be customized to the threats, tech and resources of each organization, each model should not be so wildly dissimilar that this situation is considered normal or best practice. Some companies have overcome this challenge with good tooling, good process, and standardization, and this needs to become more mainstream.

As the regulatory and legislative landscape grows more complex, it is an inevitability that threat modeling will become mandatory, especially in high-compliance industries. Developer risk management is vital in producing higher software security standards, meeting Secure by Design objectives, and reducing the remediation burden for the AppSec team. Developers should absolutely play a key role in threat modeling, ideally in conjunction with an AppSec counterpart.

Many of our subjects identified that the initial threat modeling diagram was the most painful part of the process, and issues relating to where the "source of truth" information was stored were a distraction during implementation. This is where artificial intelligence (AI) can play a safe helping hand, in addition to the auto-creation solutions offered by prominent ASPM vendors, threat modeling companies, and forward-thinking startups in the space. Getting started is half the battle, but it's time to pick up the sword and shield.

The Main Players in a Threat Modeling Solution

Perhaps the most revealing common theme we uncovered in our interview process was that, out of over twenty high-maturity enterprises, not a single one prominently included developers in threat modeling initiatives. While some AppSec professionals choose to involve them in the workshopping stage, their consultation was not central to the activity.

The reason for this can typically be pinpointed back to training. Only security-skilled developers should be involved, and across the industry, meaningful upskilling programs where continuous, relevant learning was implemented and measured are thin on the ground. In addition, both AppSec specialists and developers need threat modeling training, or the company can be assured of a suboptimal execution of the program.

Of those interviewed, 90% favored comprehensive training before threat modeling is implemented in the SDLC.

Another significant factor in faltering threat modeling initiatives could be related to how labor-intensive it can be to execute effectively.

In our study, 65% of companies used an entirely manual process with no supplementary tools or resources.

Tools were evaluated often, but many were still in the research stage of procurement. Despite this, tools or no tools, most companies indicated they could be successful with the proper standardized guidance.

Developers know their code best, and when upskilled into fully fledged security-skilled engineers, they are an asset to the threat modeling process and under the right shepherding, can take far more responsibility than has traditionally been afforded to them. There are fundamental differences in the approach developers take in vulnerability assessment, and we have observed that they are likely to need prescriptive assistance around understanding the difference between two categories of security issues, such as:

- *The syntactical mistakes in code (found by scanners) and;*
- *The architectural problems in the application itself (found by threat modeling).*

Simply put, you do not discover a SQL injection bug through threat modeling, and you do not find missing access control and authentication by scanning. A threat modeling initiative requires a dedicated, trained team—one versed in both technology and critical thinking around preventative security—to be successful. Most companies' ultimate desire is for threat modeling to be automated and more well-defined. We want it to be easy, fast, and automated, with developers positioned as the beating heart of the initiative, and the player most able to inform on specific code-level architecture and use cases.

Conclusion

Despite changing geopolitical headwinds, the fact remains that Secure by Design, as a movement and as a viable system for uplifting software quality and safety, is here to stay. To achieve its full benefit(s), enterprise security leaders must put in the work to unify and move forward with a cohesive, best-practice strategy for executing it—in full, not in part— and position developers as central players in a successful execution.

Secure Code Warrior is your partner in developer risk management, and we can assist you in uplifting, modernizing, measuring and succeeding with your Secure by Design initiative. Learn more about SCW Trust Score, SCW Trust Agent, and our suite of tools to complement your developer-driven security program.