Enterprise Strategy Group™ by TechTarget

WHITE PAPER

Toward Threat Modeling as Code

By Jon Oltsik, Distinguished Analyst and Fellow
Enterprise Strategy Group

April 2023

Contents

Executive Summary

Organizations have embraced cloud computing and cloud-native development, and many have adopted a cloud-first strategy for applications. Why? Many firms claim that moving workloads to the cloud and developing cloud-native applications help them lower costs and accelerate development cycles.

Unfortunately, these benefits can come at a price: increased cyber-risk. Threat modeling can help organizations address cyber-risks by visualizing applications, identifying risks, and then guiding organizations to implement the right controls in the right places. Are organizations using threat modeling to gain these benefits? This white paper concludes:

- **Cloud-native applications create security issues.** As organizations "lift and shift" workloads and develop new apps, they need to modify security policies and technologies, mature their security programs, enhance their monitoring capabilities, and establish cloud computing security skills. Many firms struggle in one or several of these areas.
- **Threat modeling remains confusing.** Security professionals are familiar with the threat modeling concept, and many perform threat modeling checkbox exercises for regulatory compliance purposes. When this happens, threat modeling doesn't lead to the level of detail necessary for organizations to truly understand how an adversary might compromise systems or even to recognize the impact this could create. Security teams must agree on a baseline of threat modeling requirements, align threat modeling best practices to DevOps and CI/CD pipelines, and apply threat modeling to all cloud migration and cloud-native development projects.
- **Threat modeling as code (TMaC) can help bridge the threat modeling gap.** Cloud service providers (CSPs) such as Amazon, Google, and Microsoft do a stellar job at providing blueprints for application developers that include appropriate security controls. But these guidelines are generic to industry needs, and they might not support regulatory compliance requirements or an organization's risk tolerance level. ThreatModeler seeks to address these issues by ingesting CSP blueprints and aligning them with similar TMaC templates. Users can then customize these templates for their individual needs. In this way, ThreatModeler can help standardize threat modeling best practices and democratize threat modeling for all types of organizations.

Cloud-native Applications Create Security Challenges

According to research from TechTarget's Enterprise Strategy Group (ESG) research, 71% of organizations surveyed currently develop and deploy cloud-native applications, 17% plan to develop and deploy cloud-native applications in the next 12 months, and 8% are interested in developing and deploying cloud-native applications sometime in the future.[1] Why are organizations developing and deploying cloud-native applications? Developers often use cloud-native application development to achieve benefits such as faster development time, rapid revisioning, and higher-quality application software. But while cloud-native application development has its advantages, ESG research revealed several associated security challenges (see Figure 1),[2] including the following:

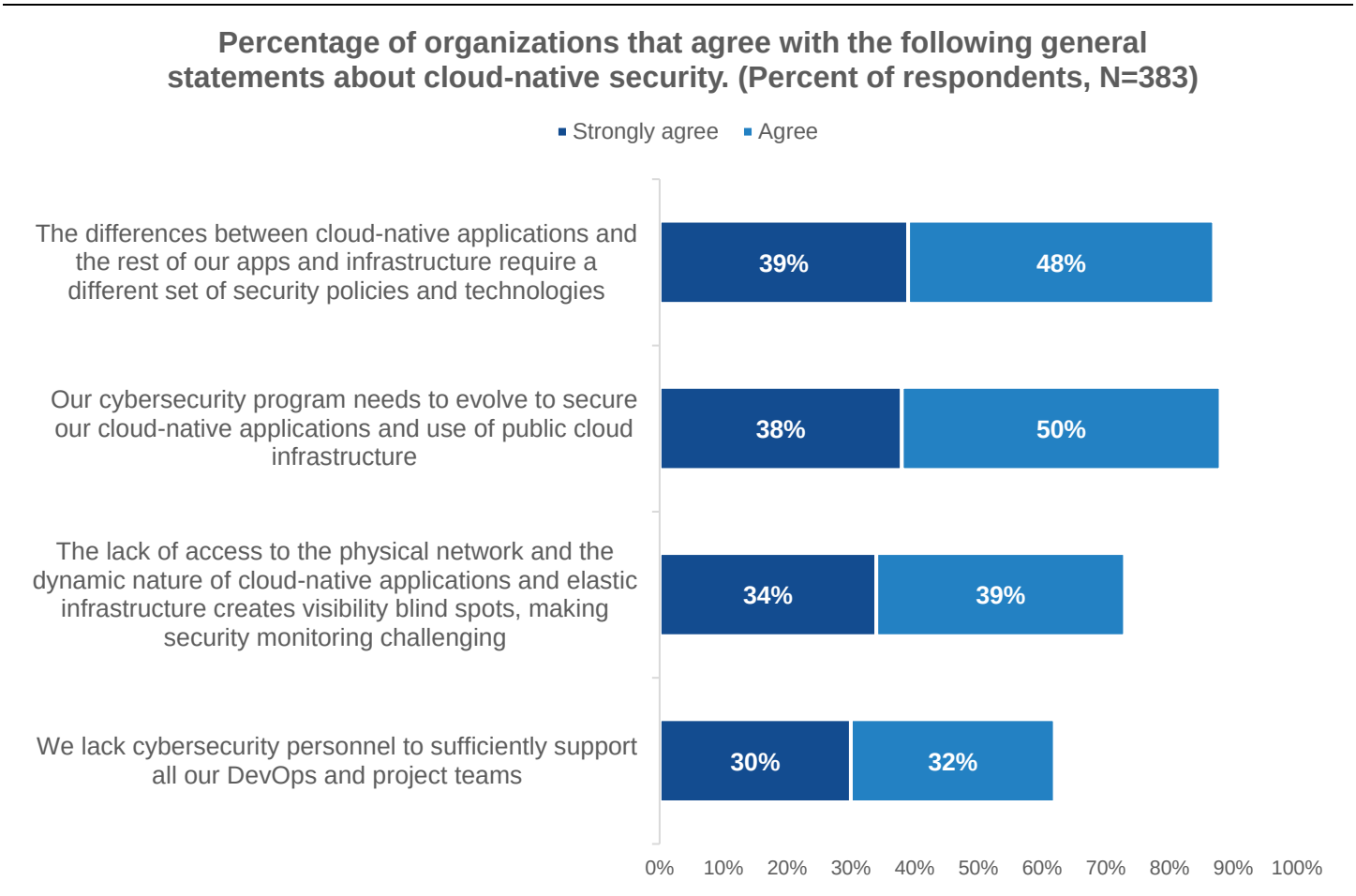
- **The need for different security policies and technologies.** Security teams can't rely on traditional policies and security technologies to keep up with a DevOps CI/CD pipeline. Rather, they need a DevSecOps methodology along with tools built for application development automation. Many security teams continue to try to force-fit traditional security tools and thus struggle to catch up with cloud-knowledgeable development teams.

[1] Source: Enterprise Strategy Group Research Report, *2023 Technology Spending Intentions Survey*, November 2022.
[2] Source: Enterprise Strategy Group Research Report, *The Maturation of Cloud-native Security: Securing Modern Applications and Infrastructure*, May 2021.

- **A requirement for maturing the security program.** ESG research indicated that the top cloud security challenges are maintaining consistency across on-premises and cloud workloads, overly permissive service accounts, and manual security processes that can't keep up with the pace of the cloud. Addressing these challenges demands the development of a mature security program consisting of standard formal processes, identity governance, and process automation.
- **Monitoring blind spots.** Nearly three-quarters (73%) of security professionals agreed that they haven't adapted security monitoring to cloud-native development innovation, such as infrastructure as code (IaC), requiring teams to monitor blind spots. Combined with immature processes, many organizations have poor visibility and no idea how cloud resources interact with each other or with existing assets.
- **A lack of the right skills and staff.** Nearly two-thirds (62%) agreed that they lack sufficient security skills and staff to support DevOps. This shortage is supported by other ESG research, which indicated that 42% of organizations said that cloud security positions have been the most difficult for their organizations to fill over the last 12-18 months as a result of the problematic skills shortage.[3]

Given all these challenges, many organizations run into a "blank sheet of paper" problem with securing cloud-native applications: They have no idea where to focus or how to start to secure them.

Figure 1. Security Challenges Associated With Cloud-native Application Development



Source: Enterprise Strategy Group, a division of TechTarget, Inc.

[3] Source: Enterprise Strategy Group Research Report, *2023 Technology Spending Intentions Survey*, November 2022.

Threat Modeling to the Rescue?

Organizations can address the "blank sheet of paper" problem described above and bridge the cloud-native security gap by integrating threat modeling into their software development and DevOps processes as a best practice. In fact, threat modeling is such an important component of cloud-native application security that it is mandated for federal agencies as part of Executive Order 14028 and NIST Internal Report 8397.

Regrettably, these guidelines aren't as detailed as they could be. Specifically, threat modeling requirements aren't really defined in either of these documents, leaving organizations in a bit of a conundrum. Security teams are forced to figure out:

- What will qualify as compliant?
- Where do we start, and what is good enough?
- Beyond regulatory compliance, can threat modeling really improve security?
- How can security teams align threat modeling with continually changing applications and CSP services?

Defining Threat Modeling

True threat modeling is not a diagram, white board discussion, or tabletop exercise done only when an application is first developed. Rather, it is a continuous lifecycle process that aligns with the DevOps cycle and a CI/CD pipeline. Simply stated, threat modeling encompasses and proceeds through phases to answer three primary questions:

1. **What are we building?** This question is targeted at the technical domain to decompose the system and diagram the architecture. This includes uncovering things such as:
   - **The structural, behavioral, and data-based aspects of applications and infrastructure.** This includes how the application should behave, how application components relate to one another, and the infrastructure used to host applications, data, services, etc.
   - **Identify user interfaces, APIs, workflows, and data feeds.** The goal here is to map out applications from end to end, including user access, how users will navigate the application to get what they want, all API calls and expected behaviors, and where and how the data flows.
   - **Identify privileged user roles and functionality.** The focus here is to define who has the keys to the kingdom and what they should and shouldn't be allowed to do.
   - **Identify sensitive data.** This involves both data at rest and in flight.
2. **What can go wrong?** This should be viewed as the business domain, where threats are identified and then applied directly to applications and workloads to assess their business impact. In this phase, organizations should:
   - **Identify adversaries, including their tactics, techniques, and procedures (TTPs).** Security teams should start with a thorough investigation into threat intelligence, looking at threat actors and campaigns targeting organizations and applications in their industry and region. When reviewing threat intelligence, it's worthwhile to frame this research using the pyramid of pain, a conceptual model used to increase the effective use of cyber threat intelligence for tasks like threat modeling. By using the pyramid of pain, security teams can uncover adversary TTPs and then develop the appropriate countermeasures.
   - **Utilize common models and resources.** Organizations should utilize all the good work available for threat modeling, including catalogues such as the OWASP Application Security Verification Standard (ASVS) top 10 vulnerabilities and the HiTrust CSF threat catalogue, for example. Security teams should also use threat modeling structures like STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, and elevation of privilege) or PASTA (Process for Attack Simulation and Threat Analysis).

- **Enumerate doomsday scenarios.** It's worthwhile to conduct tabletop exercises to better understand cyber attacks, defenses, and incident response processes. This will help uncover the attacks and targets that could be most disruptive to the business. This is especially important for risk management prioritization and investments.

3. **What are we going to do about it?** In this phase, security teams take the inputs and actions described above as part of the architecture and then envision misuse/abuse, model attacker behavior, and iterate these actions through application revisions. In response, security teams must:
   - **Map in security controls, including input validation, authentication, session handling, etc.** This is where teams add specific policy enforcement controls based on known threats or discovered vulnerabilities. Controls should be thorough, transparent to users, and validated for effectiveness.
   - **Add external and cloud services controls.** These could include access control lists, network and web application firewall rules, and native cloud security controls, among others. CSPs offer many layers of native security controls that should be explored, understood, and implemented when appropriate.

When should threat models be revisited? From a business perspective, threat models should be iterated when business objectives demand—likely every 18 to 36 months, or when a business initiative begins or changes. Drivers here could be the digitization of a business process or the introduction of a new product, for example. Technical teams should revisit threat models when changes occur to the underlying technology architecture, which is likely in a six- to 12-month cadence. Drivers could include migration to a new technology stack or a development framework change. Finally, threat model champions should remain continually engaged with threat models, conducting and iterating them in alignment with development sprints or changes in the threat landscape. Examples could include a new threat campaign, TTPs, vulnerability, etc.

Threat Modeling as Code

As part of their offerings, the major CSPs present security blueprints for application developers, especially regarding application types that utilize CSP services. Amazon has its Well-Architected Framework, which includes security guidelines; Microsoft provides Azure security patterns; and Google provides numerous security guidelines for requirements, such as securing a BigQuery data warehouse or coding Terraform blueprints and models for Google Cloud.

While these blueprints provide useful details for developers, security engineers, and blue teams, they will always need to be customized to an individual organization's business processes, regulatory requirements, and cyber-risk tolerance. Indeed, threat modeling and secure design must consider business, physical, and other domains—not just what is in the cloud. Security teams might want to pursue continuous threat modeling, but regardless of CSP blueprints, many security teams continue to struggle to identify what to focus on and how to keep pace with the DevOps CI/CD pipeline.

Recognizing this gap, ThreatModeler is pursuing a new strategy for threat modeling and DevOps that aligns well with the CSP best practices blueprints described above. ThreatModeler calls this "threat modeling as code" (TMaC), and it serves the following functions:

- Provides pre-modeled CSP patterns out of the box, with specific threat models from the ThreatModeler marketplace. This enables users to develop their own threat model templates that reflect their environment or security requirements.
- Includes automated ingestion from DevSecOps processes, such as integrated development environment (IDE) use, supply chain management (SCM) workflows, and operating a cloud identity and access management (IAM) account so that teams can see changes and their model impact immediately.

- Moves beyond runtime configuration and vulnerability guardrails like cloud security posture management (CSPM) alone. TMaC supports a full lifecycle from intended design to operational reality. This helps organizations find and fix tactical and strategic security issues during the development cycle, lower costs, and mitigates cyber-risks.

With its TMaC templates, ThreatModeler can help developers and business, operations, and security teams to:

- **Visualize the architecture.** As the saying goes, a picture is worth a thousand words. In this case, ThreatModeler provides a visual representation of process flows, trust boundaries, APIs, data exposures, administrator entitlements, and more. Armed with this visual perspective, business, IT, and security teams have a map that can help them view the system as an adversary, characterize the system, determine high-priority threats, and mitigate vulnerabilities.
- **Reuse threat models.** Once a threat model is established, it can be reused and modified for new applications or application changes as organizations integrate workloads, services, service meshes, and infrastructure.
- **Account for different shared responsibility/shared outcome postures among CSPs.** Threat models can serve as a template that can be used across CSP blueprints, establishing a baseline of best practices for developers and cloud-native applications.
- **Democratize blueprint and template threat model authorship and sharing by aligning security activity prescribed by the threat models with delivery sprints.** To accomplish TMaC, ThreatModeler provides a common platform and model to marketplace blueprints, ingested IAC diagrams, cloud state, and architect drawings. In this way, threat modeling follows the evolution of software from ideation to operation, without the need to translate between design, development, or operational forms.

Conclusion

Security practitioners recognize threat modeling as a best practice but often remain confused about what a threat model should and shouldn't include. As a result, many organizations address threat modeling as a checkbox exercise, minimizing its potential value.

It's time for organizations to establish threat modeling baselines, apply them to all application development and cloud migration projects, and modify them to accommodate application and infrastructure changes. ThreatModeler seeks to simplify this process through its threat modeling-as-code initiative that aligns specific threat models to CSP blueprints and best practices. This effort can help democratize threat modeling, enabling a DevSecOps model while mitigating cyber-risks.

About Enterprise Strategy Group
Enterprise Strategy Group is an integrated technology analysis, research, and strategy firm that provides market intelligence, actionable insight, and go-to-market content services to the global IT community. © TechTarget 2023.

contact@esg-global.com
www.esg-global.com