TOREON

# Enhancing resilience with threat modeling in the financial services industry

# Table of Contents

# Management summary

In today's evolving threat landscape - especially in financial services across the Nordics and Europe - **proactively securing applications and services is vital**. Resilience starts with being proactive.

Threat modeling is a practical way to get ahead of risks by tapping into the collective knowledge of your teams to identify vulnerabilities before they lead to real issues. **It offers a structured, collaborative approach to identifying and mitigating risks early in the design phase**—before they become costly vulnerabilities or regulatory concerns. By systematically asking "what could go wrong," financial organizations can preempt critical threats like fraud, unauthorized transactions, and money laundering.

Threat modeling not only strengthens resilience but also aligns cross-functional teams on shared security goals. It enables prioritization of risks, creates traceable mitigation plans, and embeds security into development workflows. With growing regulatory expectations (e.g., DORA in the EU), threat modeling also serves as tangible evidence of "security by design" practices.

Successful implementation requires structured programs, trained personnel, integration into DevSecOps and GRC processes, and the adoption of proven frameworks such as STRIDE. Done right, threat modeling becomes a **scalable, repeatable, and business-aligned discipline—transforming security from a reactive burden to a strategic enabler of trust and resilience.**

Sebastien Deleersnyder

CTO Toreon

# Threat modeling: security by design

Threat modeling is a proactive, structured process to identify and assess security risks in a system's design – whether a new application or an existing one. It involves systematically thinking through "what can go wrong" in an architecture before those design flaws turn into breaches. Unlike reactive security measures or after-the-fact testing, threat modeling is done early (and often) to **build security by design**.

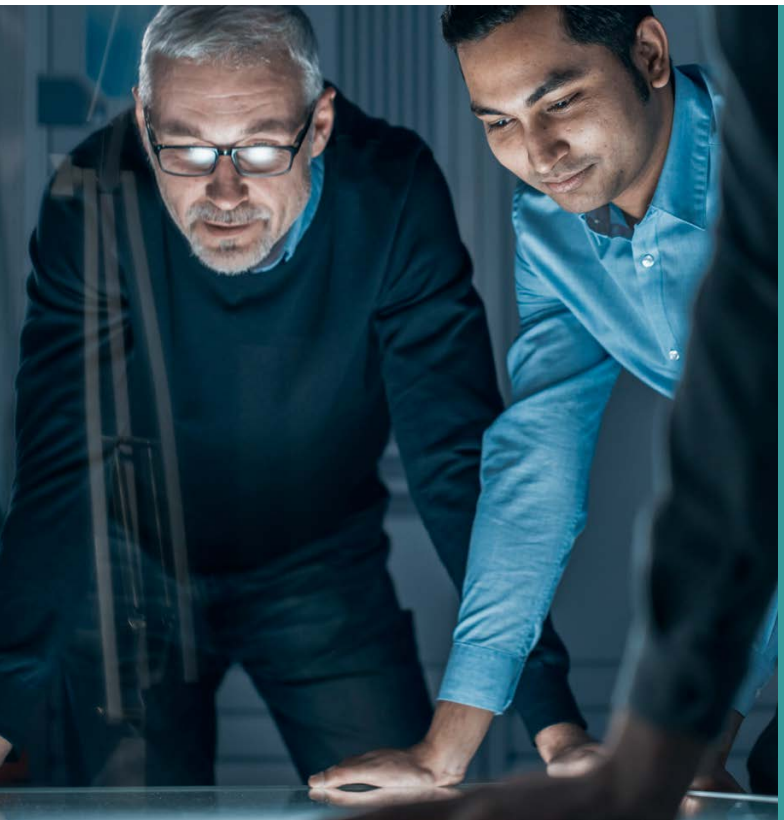
Organizations uncover vulnerabilities before code is written, preventing costly fixes later.

It typically takes the form of facilitated workshops where cross-functional teams (developers, architects, admins, and security experts) brainstorm potential threats to the system. By doing this during design or planning, organizations uncover vulnerabilities **before code is written**, preventing costly fixes later. The approach is iterative and flexible: you revisit the threat model whenever the system changes or new threats emerge.

## At its core, threat modeling asks four key questions:

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

This structured mindset ensures teams consider all angles of risk. The process usually produces visual diagrams and lists of potential threats with recommended mitigations. It's a highly collaborative exercise – not a solo security audit – which means it brings together people from different departments to share perspectives on security.



### Threat modeling in financial services

In a financial context (think of a mobile banking app or a payroll system), threat modeling sessions might include not just IT architects but also fraud analysts, business product owners, and compliance officers.

This cross-functional approach is key: everyone contributes to identifying how an attacker might abuse the system and what defenses are needed.

# Key benefits for financial services companies: from flaw prevention to priority focus

## 1. Preventing design flaws & mapping mitigations

### Preventing security design flaws

One of the most valuable outcomes of threat modeling is its ability to catch systemic weaknesses early, before they evolve into real-world vulnerabilities. Many of the most critical security issues arise not from bad code, but from flawed design assumptions ("we didn't think someone would try that!").

Research shows that nearly half of software security issues stem from design flaws). Threat modeling compels teams to think adversarially from the outset - flagging risks such as unrestricted login attempts or missing authorization boundaries - long before these are baked into the system. The result is fewer rework cycles, more robust design, and a culture of secure-by-design thinking across teams.

### Clear mapping of risks to mitigations

Beyond identifying flaws, threat modeling excels in translating threats into actionable controls. Each scenario, whether it's a spoofing attack or data tampering risk, is matched with a concrete mitigation, such as "implement secure cookies" or "log access attempts with tamper-proof trails."

This not only guides developers with precision, it also creates traceability for auditors and stakeholders. One of our customers, Jeroen Verwoest, product owner threat modeling at ABN AMRO, pointed out that the process turns security ideas into implementable requirements that teams understand and own.

## 2. Prioritization & operational efficiency

### Prioritized development and testing efforts

Not every threat is equal. Threat modeling allows teams to assess which risks carry the greatest potential impact, creating a clear security priority list. This enables engineering teams to focus on hardening the most critical areas first, while also informing test planning. Testers are empowered to probe the defenses that matter most, and avoid wasting time on low-risk areas.

For example, if unauthorized fund transfers are ranked "high," then extra scrutiny goes to role validation and transaction audit trails. This focused approach drives better ROI from both security engineering and testing activities.

### Cross-departmental visibility and consensus

Threat modeling naturally brings development, security, fraud, compliance, and risk stakeholders into the same conversation. The collaborative nature of workshops ensures security becomes a shared concern, not a siloed checklist. Misunderstandings between teams are resolved upfront, and disagreements on security trade-offs are addressed during design.

Additionally, integration with risk registers ensures that any unresolved threats are formally tracked and escalated. One European bank, for instance, directly linked threat model outcomes to its governance and risk tooling, ensuring open risks didn't fall through the cracks.

## 3. Regulatory & client assurance

### Regulatory compliance support

Threat modeling satisfies more than internal needs, it also addresses regulatory expectations. Frameworks like DORA require structured ICT risk assessments early in the design lifecycle. Threat modeling directly meets this need by documenting threat scenarios, controls, and residual risks in a traceable way. This not only satisfies auditors, but also allows organizations to demand similar rigor from their vendors and third parties.

Threat modeling aligns naturally with broader regulatory goals, including "privacy by design" (GDPR) and supply chain risk management. As DORA and similar mandates mature, threat modeling offers a clear, defensible response.

### Demonstrating security-by-design to clients and partners

In competitive financial ecosystems, clients and partners increasingly expect transparency around security posture. Threat modeling provides a compelling, high-level narrative: "We think through threats before coding begins; we build controls in from day one."

For fintechs and platform providers, this level of diligence can be a differentiator. It also enables proactive communication with partners about how risks like fraud or data leakage are handled by design—not patched reactively.

## 4. Reinforcing resilience through culture and practice

### Evolving resilience over time

Security isn't static—and neither is threat modeling. As systems evolve, so do their risk profiles. Repeating threat modeling across lifecycle phases (initial design, feature expansions, architectural shifts) ensures that defenses remain aligned with actual threats. Moreover, integrating threat modeling into agile, DevSecOps, and governance processes makes it sustainable and repeatable. The practice grows into an organizational habit, driving a long-term shift toward a security-first mindset.

"We think through threats before coding begins; we build controls in from day one."

# Collaborative risk assessment in financial applications

Financial services applications – like mobile banking, payment gateways, or salary disbursement platforms – are prime targets for attackers. Threat modeling these systems helps teams **anticipate critical risks** such as fraud, unauthorized withdrawals from accounts, or even misuse of the system for money laundering.

## Real-life example: a salary payment app

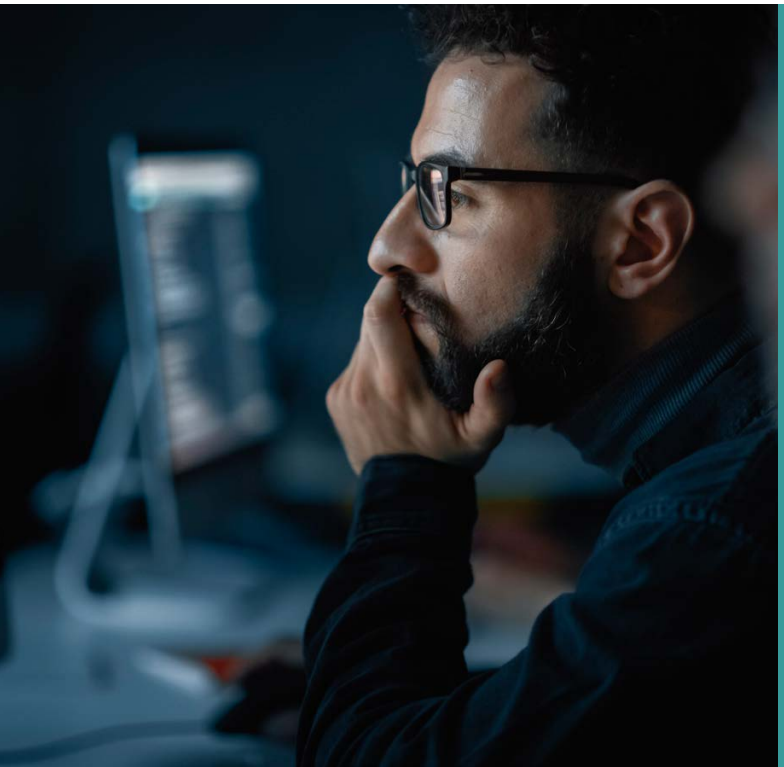
For example, imagine a salary payment app: a threat model might expose a scenario where an attacker tries to spoof identity to reroute funds (fraud), manipulate transaction data in transit (tampering), or abuse the app's logic to perform illicit transfers (money laundering).

By mapping out the data flows and entry points, the team can ask "how could someone subvert this?" and uncover weaknesses in authentication, authorization, or auditing that could enable those attacks.

Crucially, this process isn't purely technical. It ties into business concerns. In our example, preventing fraudulent salary withdrawals isn't just an IT issue, it's a business priority. Threat modeling provides a forum where developers and security engineers consider these business-impact scenarios (often called "doomsday scenarios"). It ensures that mitigations are baked in for the highest-impact threats.

In a banking app, that might mean building checks for anomalous transaction patterns (to catch fraud), enforcing strict identity verification (to stop spoofing or unauthorized access), and logging all fund movements with tamper-evident trails (to detect and deter money laundering attempts).

By involving stakeholders from compliance and fraud departments in threat modeling workshops, the team gains a 360° view of potential abuse cases. The result is not only a list of technical threats (like "SQL injection on transaction history DB") but also higher-level abuse cases ("attacker tries to withdraw funds from someone else's account") with corresponding defenses.



## Threat modeling is inherently collaborative

Threat modeling breaks down silos. Developers bring knowledge of how the system works, business owners clarify what "normal" use is versus "malicious" use, and security folks contribute knowledge of attack techniques. This collaboration leads to cross-departmental visibility into security: everyone ends up on the same page about what the major threats and priorities are.

A developer might realize why a seemingly minor feature could be a fraud risk, or a business manager might better appreciate the need for a control that was initially seen as burdensome. This consensus-building is invaluable in financial organizations where security must balance with user experience and business needs.

# How threat modeling addresses critical risks for the financial services industry

## Fraud & identity abuse

Financial fraud often starts with impersonation or stolen credentials. In threat modeling, the team would consider spoofing threats – e.g. an attacker pretending to be a legitimate user or even an internal service. They might ask, "How could someone fake their identity to fool our system?"

This leads to mitigations like stronger authentication, anomaly detection for user behavior, or multi-factor authentication for sensitive transactions. In fact, spoofing is the first category in the popular STRIDE model of threats (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege).

Using STRIDE, teams systematically check each category. For instance, to counter spoofing and fraud, a banking app might enforce strict identity verification and session management. Threat modeling makes sure such controls are identified early as requirements, not added ad-hoc later.


## Unauthorized withdrawals

In banking, an "unauthorized withdrawal" could occur if an attacker exploits a flaw to initiate payments they shouldn't. Threat modeling flushes out where such unauthorized actions could be possible.

For example, the team might identify a potential "elevation of privilege" threat: an attacker finding a way to perform actions beyond their account permissions (like a standard user invoking an admin-only fund transfer function).

By mapping out user roles, entry points, and trust boundaries, the model highlights where an extra authorization check or validation is needed. The result is a clear mitigation ("Ensure the transfer service verifies the requester's role and account ownership before processing withdrawals.") Each threat gets a corresponding countermeasure, so the risk of illicit withdrawals is significantly reduced.

## Money laundering & abuse of business logic

Money laundering often involves abusing legitimate features (splitting transactions, using mule accounts, etc.) rather than exploiting a technical bug. This is where threat modeling shines by not limiting scope to technical vulnerabilities, but also considering **misuse cases**.

In a threat modeling session for, say, an international funds transfer system, participants might brainstorm how criminals could try to circumvent limits or detection (e.g., breaking a large transfer into many small ones to avoid reporting thresholds). These are essentially "Repudiation or information disclosure" concerns in STRIDE terms (covering audit evasion and illicit information use). By documenting these scenarios, the organization can introduce controls like transaction monitoring alerts, limits on transfer frequencies, and required approvals for suspicious patterns. In other words, threat modeling bridges the gap between technical threats and fraud risk management. It ensures that anti-fraud and AML (anti-money laundering) considerations are part of the design.

## Visualising threat modeling

To visualize how threat modeling captures these concerns, we use a **data flow diagram (DFD)** of a banking application.

Our DFD shows a simplified online banking system with trust boundaries (dashed red lines) segregating the web front-end, application server, and back-end processing. Each arrow is a data flow that could be targeted by an attacker for fraud or abuse. The red dashed lines indicate trust boundaries (e.g. DMZ vs internal network).

Threat modeling such a system helps identify points where an attacker might spoof identity, tamper with data, or misuse transactions, and drives the design of appropriate security controls. By scrutinizing each component and data flow in diagrams like the above, cross-functional teams can spot where security assumptions might break.

For example, the DFD highlights a flow from the application server to a "Financial Transactions Host". Threat modeling would ask "What if an attacker intercepts or alters that data? Do we have encryption, authentication, and validation on it?"

This process results in a clear **mapping of risks to mitigation measures** documented for each part of the system.

# Integrating threat modeling with testing and development

Threat modeling isn't meant to exist in a vacuum or to replace other security activities. Instead, it enhances them. A great example is how it integrates with penetration testing and DevSecOps practices.

Jeroen Verwoest, who has spearheaded threat modeling at ABN AMRO, shares insights on using threat models to scope and focus pentests effectively. The idea is straightforward: use the threat model's results to inform what the pen testers should target and then use pen test findings to improve the threat model

## During development

The output of threat modeling (those lists of threats and requirements) feeds directly into the development backlog. Security requirements identified via threat modeling can be written as user stories or acceptance criteria.

For instance, if threat modeling a banking API uncovers a threat of information disclosure (the team will add an item to "Implement data encryption in transit and at rest" or "Mask personal data in logs". By integrating these into agile planning, you bake security into implementation.

Many organizations tie threat modeling into their DevSecOps pipeline: for critical applications, a threat modeling step (or at least a review of an existing model) is mandated at design time, and the identified controls must be in place (and perhaps even checked by automated security tests) before deployment.

This approach ensures that dev teams treat security tasks from threat modeling with the same priority as features. It shifts security left in the development lifecycle, which is exactly the goal of DevSecOps – to avoid late-stage surprises. Threat modeling provides DevSecOps teams the intel they need to embed controls from the start, so security isn't an afterthought

This targeted approach makes pen testing more efficient and effective, improving its ROI.

## Improving penetration testing

Traditional pentests sometimes suffer from unclear scope or findings that don't resonate with the business. Threat modeling can solve these issues. During the intake phase of a security assessment, if a threat model exists, it can be shared with the testing team.

This gives pen testers a map of the application's architecture, critical assets, and the threats already considered by the development team. They can plan attacks that either validate those defenses or attempt scenarios the team thought were unlikely. Verwoest explains that using threat model outputs, pentesters can avoid redundant tests and focus on simulating real-world attacker goals.

For example, if the threat model shows that "unauthorized money transfer" is a high-risk scenario with certain countermeasures in place, the pen tester will specifically probe those countermeasures (Can I bypass the authorization? Can I manipulate transaction data?). This targeted approach makes pen testing more efficient and effective, improving its ROI . It also ensures the test is aligned with what the business cares about (no "unrelatable test report" full of technical jargon – each finding can be traced to a threat the business understands).

On the flip side, after a penetration test, the findings should be fed back into the threat model. If testers discover a new threat scenario that the model missed, the threat model gets updated – and that knowledge is now captured for future use. This creates a powerful feedback loop: threat models guide testing, and testing refines threat models. Over time, this synergy greatly increases an organization's security maturity.

## DevSecOps integration points

Threat modeling can tie into various stages of the CI/CD pipeline. For example, during code review, developers can check that the code indeed implements the mitigations the threat model calls for. During continuous integration, security unit tests or static analysis rules could verify certain threat mitigations (like ensuring input validation is present for all inputs identified in the model).

Additionally, as infrastructure is codified (IaC) and deployments are automated, threat models can inform security gates – e.g., if a threat model expects a certain network segmentation or encryption, the pipeline can enforce those configurations before promoting code.

Some teams even store threat models in version control alongside code; when a significant change occurs (say a new microservice is added), it triggers an update/review of the threat model. Modern tooling and platforms (such as IriusRisk, ThreatModeler, OWASP Threat Dragon, etc.) support API integrations that allow linking threat modeling with issue trackers and GRC systems.

In a webinar on banking security, experts noted that integrating threat modeling data with GRC tools (like Archer or ServiceNow) via APIs can automate tracking of mitigation tasks and escalation of exceptions. For instance, if a dev team doesn't complete a high-priority security fix identified in the threat model, the system can flag it in the risk register for management attention. This kind of orchestration ensures that the outputs of threat modeling (the "to-dos" and risks) are not forgotten – they become part of the workflow, visible to all stakeholders.

Experts noted that integrating threat modeling data with GRC tools (like Archer or ServiceNow) via APIs can automate tracking of mitigation tasks and escalation of exceptions.

## Summary

When threat modeling is woven into development and testing, it creates a cohesive security lifecycle: design secure, build secure, validate security, and improve continuously.

Threat modeling amplifies the effectiveness of each phase – developers code defensively with threats in mind, testers attack intelligently with knowledge of the system's design assumptions, and the organization learns and adapts quickly.

The payoff is a robust DevSecOps culture where security is truly everyone's responsibility, guided by a living threat model.

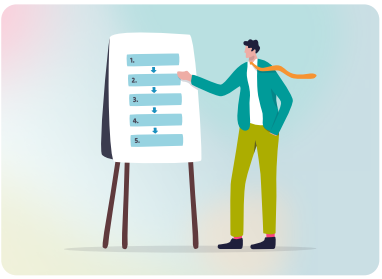
# Building an in-house threat modeling capability in 4 simple steps

Establishing threat modeling as a repeatable practice in your organization (rather than a one-off exercise) requires some planning and investment. At Toreon, we introduce threat modeling in an organization by following a practical, repeatable approach anchored in our threat modeling playbook. The focus is on integrating the practice in a way that works for people, processes, and tools. It's not a one-off activity, but something we embed into how we build and operate secure systems.

By taking this people-process-technology approach, threat modeling becomes a natural part of how systems are built and secured. The goal isn't just to catch flaws—it's to shift the culture so security is considered from day one, by everyone involved.

## Step 1: Aligning with stakeholders

We start by identifying where threat modeling adds the most value, which is typically in high-impact systems or major design initiatives. From there, we engage key stakeholders early: security teams, architects, developers, risk and compliance. This alignment ensures everyone understands the "why" behind threat modeling and how it fits into their responsibilities. We often begin with a pilot project to demonstrate impact and build internal momentum.



## Step 2: Enabling people through training

Next, we invest in people. Not everyone starts out knowing how to approach threat modeling—that's why practical training is key. We run interactive sessions where teams learn to think like an attacker, draw out system flows, and identify weak spots before they become real problems. We also identify champions across teams who help coach others and sustain the practice over time.

## Step 3: Embedding in the process

We define a simple, repeatable process that fits how the organization already works—whether that's agile, DevSecOps, or more traditional delivery models. Threat modeling is introduced early in the design phase, and outputs (like identified risks and mitigations) flow directly into planning and tracking tools. This ensures threat modeling becomes part of how features are scoped and delivered, not a last-minute checklist.

## Step 4: Supporting with the right tools

Finally, we look at the tooling. Once the people and process foundations are in place, we introduce tools that help teams scale their efforts—whether that's drawing diagrams, documenting threats, or tracking mitigations. These tools don't replace the thinking, but they help make the work repeatable and visible across teams. Automation plays a role too, flagging when models need to be updated or when mitigations are overdue.

# Conclusion

For financial services companies in the Nordics and across Europe, enhancing resilience means being proactive. Threat modeling offers a pragmatic way to achieve that by **bringing together the collective knowledge of your organization to anticipate and thwart attacks before they happen**.

Financial regulators and frameworks (like DORA) are increasingly expecting this level of diligence – they want to see that firms are considering threats early and often. By adopting threat modeling, you're not only checking that compliance box, but **truly embracing security by design**, which ultimately protects your customers, your reputation, and your bottom line.

The process scales from small apps to large systems and can adapt to new technologies (cloud, AI, etc.) by applying the same principles. As you build your in-house threat modeling capability, leverage the wisdom of **industry best practices** to jumpstart your journey and avoid common pitfalls.

# Next steps

If you're looking to accelerate your threat modeling maturity, we encourage you to take the next step.

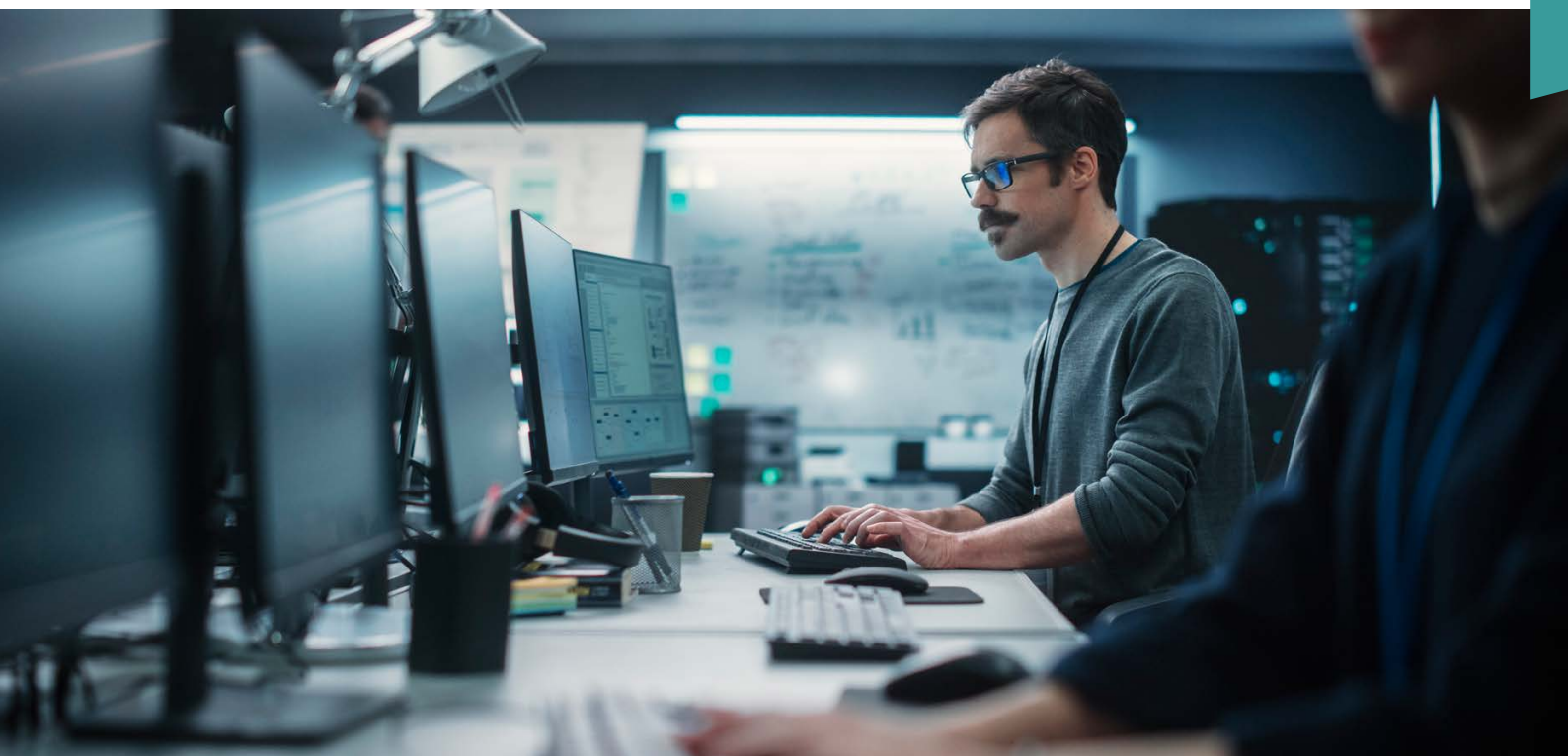
Pilot a threat modeling workshop on an upcoming project or assess an existing critical application with fresh eyes through a threat modeling lens. Secure design is a journey – start rolling the DICE and make threat modeling a cornerstone of your security strategy.

Our team has extensive experience helping organizations institute effective security-by-design practices and we offer training programs to equip your teams with the skills to succeed. We're happy to discuss how a structured threat modeling program could look for your environment and share further best practices.

You might be surprised by the gaps uncovered and the creative solutions your team comes up with. Over time, those insights compound into a significant increase in your organization's security posture.

We look forward to helping you build secure, robust financial systems that can confidently withstand the threats of today and tomorrow.

Stay safe, and happy threat modeling!



# Take the next step in maturing your threat modeling practice

### Stay up-to-date on threat modeling

This whitepaper is part of a series of blogs, whitepapers, cases and other content that we publish on threat modeling.

Want to stay informed and know when we publish new content? Then please make sure to subscribe to our monthly Threat Modeling Insider newsletter.

We promise we will never ever send you more than one mail a week. No sales pitches, just relevant, informative and qualitative content.

**Subscribe to THREAT MODELING INSIDER**

### Discover our threat modeling training

We offer two tailored security training options: an in-company program for teams or a 20-hour online course for individuals.

Both provide hands-on experience, expert guidance, and custom threat modeling to elevate your security skills.

**Discover our TRAINING OFFERING on threat modeling**

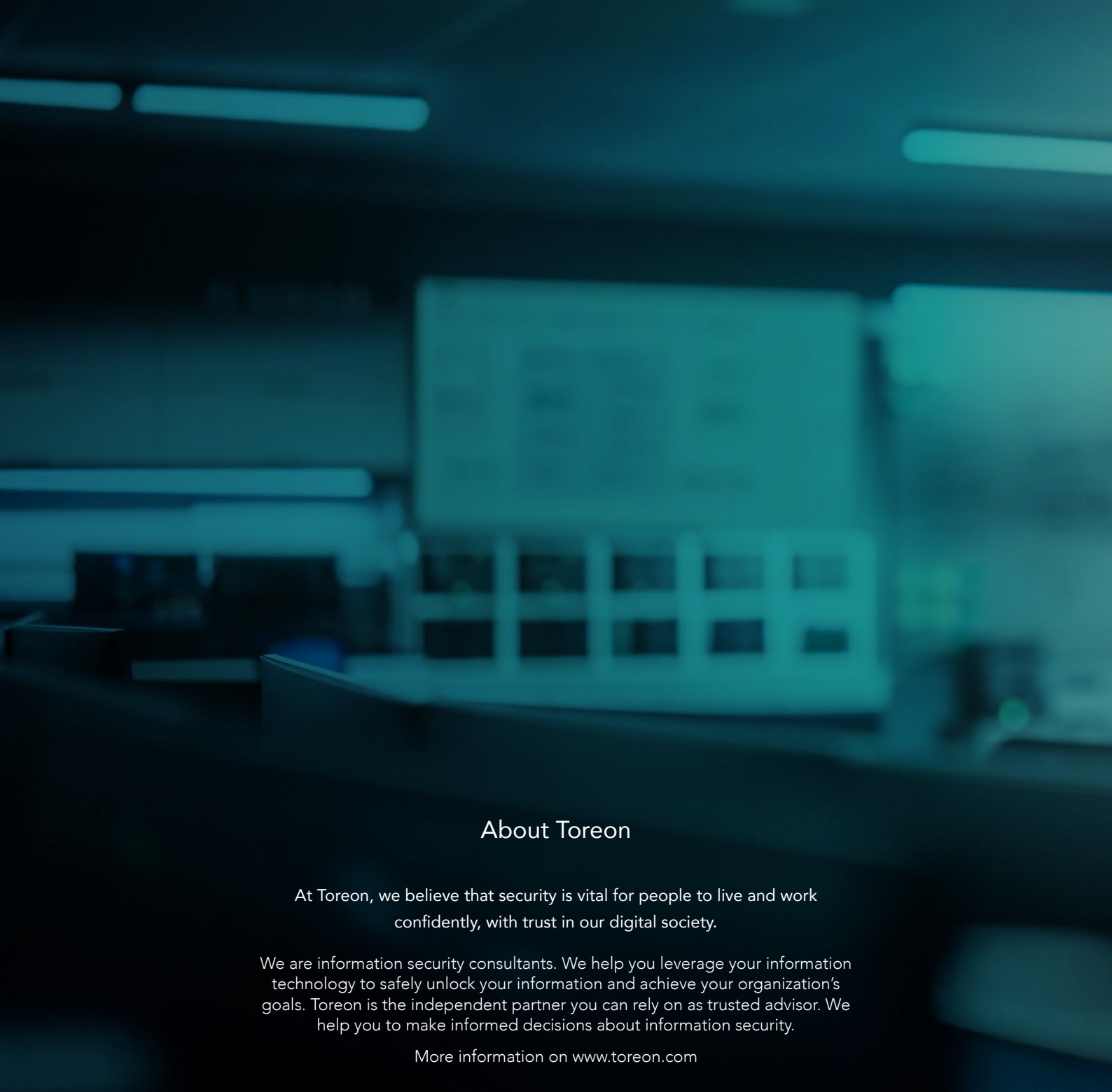
### Got security-related questions?

Got specific questions on threat modeling or cybersecurity in general?

Don't hesitate to book a free call with one of our experts.

We're sure we can help or advice.

**Book a free EXPERT CALL**



## About Toreon

At Toreon, we believe that security is vital for people to live and work confidently, with trust in our digital society.

We are information security consultants. We help you leverage your information technology to safely unlock your information and achieve your organization's goals. Toreon is the independent partner you can rely on as trusted advisor. We help you to make informed decisions about information security.

More information on www.toreon.com

TOREON

**Toreon BV**

Grotehondstraat 44 1/1, B-2018 Antwerpen, Belgium | T +32 33 69 33 96

VAT BE0542.795.568 info@toreon.com | www.toreon.com