Threat Models as Mandatory Disclosures: A Vision for Security Transparency

By Dinis Cruz and ChatGPT Deep Research

Executive Summary

Digital products today suffer from a lack of security transparency, creating a classic "market for lemons" scenario in cybersecurity. Vendors often know far more about their product's security (or lack thereof) than buyers do, leading inferior security offerings to thrive and outcompete higher-quality ones [1]. This information asymmetry erodes trust and leaves consumers unable to distinguish secure products from insecure ones. To correct this market failure, we propose a long-term vision where publishing threat models becomes a regulatory requirement for companies – much like financial statements and food ingredient labels are mandated today. Requiring organizations to disclose standardized threat models would introduce a reliable **signal** of security quality into the market, allowing customers, investors, and regulators to make informed comparisons. Publishing a threat model provides concrete evidence of the threats a company has considered and mitigated, substantiating security claims that today are often vague or unverified [2]. In the same way that transparency in finance and food safety improved those industries, **security transparency via public threat models** can drive accountability and higher baseline security across the software ecosystem. This document outlines the rationale, historical analogies, a maturity roadmap, stakeholder impacts, technical enablers, and practical steps toward making **threat model disclosures a norm in corporate practice.** It blends visionary foresight with concrete next steps for the threat modeling community and policymakers.

Historical Parallels: From Opaque Risk to Mandatory Transparency

Financial Reporting: In the early 20th century, financial markets were rife with hidden risks and corporate misdeeds. The 1929 stock market crash and ensuing Great Depression exposed how *lack of transparency* enabled fraud and shattered public confidence [3] [4]. The regulatory response – including the Securities Act of 1933 and the Securities Exchange Act of 1934 – fundamentally changed the game by **mandating truthful financial disclosures**. Companies selling securities were now required by law to provide audited financial statements and reveal material risks, so that investors could make informed decisions. For example, the 1933 Act *"required registration of most securities sales"* and insisted that *"investors must receive truthful financial data about public securities"* being offered [5]. Over subsequent decades, financial transparency became routine: quarterly reports, annual audits, and standardized accounting rules (GAAP/IFRS) turned corporate finance from a black box into a well-lit storefront. This didn't eliminate all fraud or failure, but it raised the overall trust and stability of markets by exposing lemons and rewarding sound management. Today, no serious company could imagine raising capital without publishing its financial health – it's simply an expected duty of market participation.

Food and Product Labeling: A similar evolution occurred in food safety and consumer products. In the past, buyers had little insight into what they were ingesting – harmful ingredients, poor hygiene, and false claims were common. Public outcry and health scandals eventually led to regulations forcing

transparency in labeling and standards. A landmark in the United States was the Nutrition Labeling and Education Act of 1990, which *"required standardization of the health and nutritional information food manufacturers provide to consumers on food packages"*, giving birth to the now-familiar Nutrition Facts label [6]. Suddenly, companies had to openly disclose calories, ingredients, and daily values, subject to uniform definitions. This empowered consumers to compare products and made nutrition a competitive factor, incentivizing improvements. Analogous mandates exist for product safety (e.g. appliance energy ratings, car safety crash ratings, etc.), all based on the principle that **transparency drives quality**. What used to be proprietary or inconsistent information became a common baseline of disclosure. The food industry did not crumble under these requirements; rather, it adapted and innovated, and consumers rewarded those who delivered healthier, safer products. The progression from an opaque "buyer beware" market to one of labeled, accountable goods is a powerful precedent for cybersecurity.

Cybersecurity Today: By comparison, the digital products industry is still in a Wild West stage regarding transparency. When a software vendor claims to be "secure" or "enterprise-grade," buyers largely have to take it on faith. Aside from basic compliance checkboxes or penetration test certificates (which are often private), there is no widely mandated disclosure of *how* a product was secured or what threats were considered. As a result, the software market exhibits the same failures once seen in finance and food: security quality is largely unverifiable, so bad practices hide behind marketing rhetoric. Studies and experts have noted that cybersecurity markets suffer from serious information asymmetry – **buyers can't verify claims, so weaker products often prevail on cost or features** [7] [8]. Just as the best cars were driven out of the used-car market before transparency, the most secure software can be undervalued if nobody knows the difference [9]. The absence of a "security label" or public threat model keeps everyone in the dark. The next sections lay out a vision for moving cybersecurity onto a path already tread by finance and food: from opacity to transparency, from voluntary best-effort to **regulated disclosure of security posture**, with threat models as the centerpiece of that transparency.

Roadmap to Regulatory Norm: From Ad-Hoc to Mandatory Threat Models

Achieving mandatory threat model disclosures will be a journey. We can chart the evolution of this practice in maturity stages (borrowing Wardley Mapping's concepts of Genesis, Custom-Built, Product, and Commodity). Below is a roadmap from the current state to a future state where publishing threat models is an established regulatory norm:

- **Genesis (Novel Practice):** *Current State.* Threat modeling today is in a nascent stage in terms of industry-wide adoption and standardization. Pockets of security-conscious organizations (often big tech companies or high-assurance industries) perform threat modeling, but approaches are inconsistent and often bespoke. There is no universal format or expectation that these analyses be shared externally – they are typically internal documents, if they exist at all. In this Genesis phase, publishing a threat model openly is almost unheard of, and those few cases are voluntary "thought leadership" rather than norm. The concept of security transparency via threat models is embryonic, championed only by visionaries. For most companies, threat modeling itself might be done informally or not at all. The field is experimenting with various methodologies (STRIDE, PASTA, LINDDUN, etc.), and tooling is rudimentary or custom-built scripts. **Key characteristics:** highly manual process, no standard outputs, and zero regulatory pressure to disclose findings.

- **Custom-Built (Emerging Adoption):** *Near-Term Transition.* As the value of threat modeling gains recognition, more organizations start implementing it, though each in their own way. We enter a phase where early adopters build their own internal threat modeling programs and tools, integrating them into DevSecOps workflows. Still, sharing these models with outsiders is rare. However, cracks in the secrecy begin to show: forward-thinking companies may experiment with partial disclosure to customers (e.g. including a high-level threat analysis in whitepapers or security certifications). Industry groups and thought leaders call for greater transparency. We also see the first proposals for standards. In this stage, regulators and customers **start asking** whether a vendor has a threat model and how often it's updated, even if not yet requiring it. The practice remains primarily "custom-built" – each organization defines "threat model" in its own context – but the idea of a common approach is germinating. This is analogous to the period when different banks and firms had their own ways of reporting finances, before GAAP made them comparable.

- **Product (Standardized & Tool-Supported):** *Mid-Term (Improved Standardization).* The industry coalesces around common frameworks and tooling for threat modeling. Here, the concept moves into the **product phase**: vendors provide dedicated threat modeling tools, services, or platforms (several already exist, and this trend accelerates). Crucially, **standards emerge** for describing threat models – perhaps an open schema or an ontology – enabling interchange of threat model data between tools and organizations. We might see something like an "Open Threat Model Format" become widely adopted [10], analogous to how financial XML or XBRL formats standardized financial reporting. With standardization, regulators can more easily point to an objective requirement (e.g. "Threat models must at minimum contain A, B, C and follow format X"). At this stage, some regulators may begin requiring threat modeling as part of security compliance (for example, a sector regulator mandates that critical infrastructure providers perform threat modeling and attest to it, or the SEC requires a statement of cyber risks in filings). Publishing of threat models starts to happen in pockets, perhaps as part of certifications or due diligence processes – for example, a cloud provider might publish a sanitized threat model of its core platform to reassure customers. The practice is not yet universal, but it's moving from a craft to an engineered, repeatable process with supporting products. Stakeholders start to view threat model documentation as a normal part of software documentation, even if not everyone reads it in detail (similar to how a prospectus exists for every stock, even if few retail investors read it cover-to-cover).

- **Commodity (Ubiquitous Utility):** *Long-Term Future State.* Threat modeling (and its disclosure) becomes a commodity-like practice – ubiquitous, highly automated, and mandated by regulations in multiple jurisdictions. In this phase, **publishing a threat model is as routine as publishing financial reports** for a public company. Regulatory bodies or industry consortia specify what needs to be disclosed: e.g. an annual "Security Threat Model Report" accompanies financial 10-K filings, or product-specific threat models must be posted in an online registry before a product can be licensed/sold. These threat model disclosures are standardized, thanks to the previous stage – possibly filed in a machine-readable format that can be aggregated and analyzed across the industry. The underlying process is commoditized: automated tools (potentially powered by AI) assist in generating and updating threat models continuously, integrated into development pipelines ("threat modeling as code"). The output is a living document/graph that updates with each significant system change, and a sanitized version of it is always available to stakeholders. **Verification ecosystems** might arise, analogous to financial auditors – e.g. certified third parties or automated scanners validate that a published threat model is a fair representation of the system's risks. At the commodity stage, transparency in security is simply expected. Companies that fail to provide a threat model are as suspect as a company failing to publish financials – it would be seen as either negligence or an attempt to hide problems [11]. The market dynamic shifts: vendors compete on security in a substantive way, knowing customers or regulators will examine their threat model. Good security practices are rewarded by market preference or even command higher prices, whereas products with skimpy or no disclosed threat model are shunned ("no threat model? Let's look elsewhere…they aren't even trying" [11]). Ultimately, security assurance becomes an integral part of product value, enabled by the common expectation of threat model disclosure.

Transitioning through these stages will require effort and advocacy. Early in the roadmap, voluntary action and industry collaboration are key – developing standards, proving feasibility, and demonstrating the benefits. Later, top-down pressure from regulators will likely cement the practice (just as financial disclosure was eventually enforced by law). **Wardley Mapping this journey** helps to anticipate the tipping points: for example, moving from Custom to Product stage might coincide with the release of a widely used threat modeling platform or an open standard that gains broad support. Moving from Product to Commodity will likely require regulatory intervention or a major paradigm shift (perhaps a scandal that prompts legislation, akin to financial crashes or food safety incidents). The trajectory is clear: to make security a visible, comparable attribute of products, we must evolve threat modeling from today's niche art into tomorrow's mandatory business practice.

Stakeholder Perspectives: Benefits and Trade-offs

Adopting mandatory threat model disclosures would impact various stakeholders in different ways. We outline the key benefits and potential trade-offs for four primary groups:

- **Developers & Engineering Teams:** For software creators, publishing threat models encourages building security in from the start and keeping documentation current. **Benefits:** Developers gain a clearer understanding of security requirements and assumptions in the design – the threat model acts as a blueprint of what must be defended and why. This can reduce security bugs (since threats are identified upfront) and cut down miscommunication; for example, it clarifies who is responsible for validating inputs or handling a particular risk, avoiding the "I thought you handled that" syndrome [12] [13]. Having to prepare a model for disclosure also elevates security as a first-class quality attribute, which can lead to better training and a security-aware engineering culture. **Trade-offs:** On the downside, developers face extra work to create and maintain threat models, which can feel burdensome in fast-paced environments. There may be concerns that public threat models expose too much detail, creating anxiety that every design flaw will be scrutinized. Also, if threat models become part of compliance, development teams might chafe under what feels like additional bureaucracy. Mitigating this requires good tool support and automation so that maintaining a threat model is as natural as maintaining code documentation or tests. Overall, when done well, developers benefit from clearer guidance and fewer firefights, at the cost of some upfront effort and process discipline.

- **Consumers & Users:** For end users (individuals or enterprises purchasing software/hardware), transparent threat models provide a window into product security that is unprecedented today. **Benefits:** Consumers would be empowered to make comparisons between products on security merits, not just marketing claims. Much like a nutrition label allows shoppers to choose healthier foods, a threat model disclosure would let buyers see if a product meets their risk appetite (e.g. does this home IoT camera consider privacy threats? Does this SaaS app model insider threats and data leaks?). This could improve trust – a vendor willing to show their security homework signals confidence. As security transparency increases, consumers may also see overall safer products on the market, as vendors compete to avoid the embarrassment of a weak published threat model. **Trade-offs:** The challenge is that average consumers or even enterprise buyers may not have the expertise to interpret threat models in detail. There's a risk of overwhelming users with technical data, or of disclosures becoming a formal checklist that nobody reads (like long privacy policies). To address this, the format of disclosure might include multi-layered information: a high-level "security facts summary" (akin to a nutrition label with key points) for quick understanding, with the detailed threat model available for auditors or curious experts. Another trade-off is the potential false sense of security – a published threat model is not a guarantee of actual security, so consumers must avoid complacency. Nonetheless, on balance, giving users *something* is better than the status quo of blind trust, especially if usability of these disclosures is taken seriously (for example, standardized labels or ratings can simplify choices [14] [15]).

- **CISOs & Security Teams:** For Chief Information Security Officers and corporate security teams, mandated threat model disclosures would reshape how security is managed internally and evaluated externally. **Benefits:** Internally, the CISO can leverage the requirement as a mandate to secure resources – e.g., "We must invest in threat modeling, it's required and our reputation (or compliance) depends on it." It provides a structured way to catalogue and communicate risks. A living threat model, once in place, helps the security team track mitigations and check that development is addressing the highest risks. Externally, when evaluating third-party products or suppliers, the CISO gains a powerful tool: instead of sending endless security questionnaires, they could **review the supplier's published threat model** to understand its security posture [16]. This can streamline vendor risk management (focusing follow-up questions on any gaps the threat model reveals). It might also foster industry sharing: security teams could compare notes on threat models, improving collective knowledge of attack patterns. **Trade-offs:** One concern is that publishing a threat model could increase liability – if a breach occurs via a threat that was in the model, it could be seen as negligence ("you knew about this threat but still got breached"), and if a threat wasn't in the model, it could also be seen as negligence ("you failed to anticipate this threat"). This damned-if-you-do/don't scenario might make some CISOs nervous. There's also the fear of exposing system details that attackers might exploit or of arming competitors with architectural insights. Security teams would have to work carefully to sanitize and partition what is shared (more on that in a later section). Additionally, preparing a public disclosure means more scrutiny on the security team's work – some may worry it exposes them to criticism. However, proponents argue that **sunlight is ultimately beneficial**: it drives teams to be more rigorous and enables constructive external feedback. Over time, as publishing threat models becomes routine, CISO concerns are likely to be eased by clear guidelines on safe disclosure and perhaps legal safe-harbors (similar to how vulnerability disclosure is encouraged today under coordinated programs).

- **Regulators & Policy Makers:** Regulators stand to play a pivotal role in this vision, and they too face benefits and trade-offs. **Benefits:** Mandating threat model disclosures could dramatically improve baseline security across the industry without having to micromanage technical solutions. It creates a market-driven mechanism: regulators set the transparency requirement, and market forces do a lot of the work to reward good security (companies with poor threat models will feel pressure to improve or face loss of business). It also provides regulators with valuable data – systemic risks might be observable by analyzing common threat gaps across published models, informing further policy. In critical sectors (finance, healthcare, energy), having operators publish threat models would enhance oversight: agencies could quickly assess if key threats (like those to safety or critical data) are being universally addressed. Additionally, it shifts some responsibility onto companies to police themselves, which is more scalable than government trying to inspect everyone's code. **Trade-offs:** On the other hand, regulators must be prepared to define standards for what a "good" threat model looks like and to audit compliance – a non-trivial task. There's a risk of creating a paperwork exercise if not done thoughtfully: companies might generate low-quality threat models just to satisfy the rule, akin to a "check-the-box" compliance mentality. Regulators will need to engage with technical experts and industry to devise the frameworks and possibly phase in requirements gradually. They also have to consider the global context – if only one jurisdiction mandates this, it might disadvantage its companies unless harmonized internationally. Finally, regulators might face pushback about public safety: would publishing threat models publicly hand attackers a roadmap? (We address this concern in the next section; in short, a properly handled disclosure should not significantly aid attackers, just as publishing an ingredients list doesn't teach someone how to poison the food.) Despite these challenges, regulators have precedent to draw on: *transparency requirements in other domains have worked* and become self-sustaining norms. The key will be structuring the mandates in a way that maximizes genuine security insight while minimizing bureaucracy and sensitive detail exposure.

Technical Enablers for Security Transparency

Turning the vision of mandatory threat model disclosure into reality will require significant technical groundwork. Several enablers must be in place to support standardization, automation, and safe sharing of threat models. Here we outline the key technical components and initiatives needed:

- **Standardized Formats and Taxonomies:** Just as financial disclosures rely on standard accounting schemas, threat models will need a common language. A standardized threat modeling format (such as a machine-readable schema or an open ontology) is crucial so that disclosures are comparable across organizations. Work is already underway in this area – for example, the OWASP **Ontology-Driven Threat Modeling (OdTM)** project defines an approach to formalize security knowledge (system components, threats, mitigations) as ontologies in the OWL language [17]. In practice, this means security experts can encode their knowledge of, say, web application threats, and share that schema. A company's specific threat model can then instantiate those standard classes (e.g., an "SQL injection" threat node linked to a "database" component node with a "parameter validation" mitigation). Standard taxonomies for threats (like STRIDE categories or ATT&CK techniques), for assets, and for mitigations would enable everyone to describe risks in a consistent way. This not only makes it easier to understand each other's threat models but also allows tooling to automatically check for completeness (e.g., "You have a web app, did you consider XSS? It's in the standard web threat list."). Regulators could reference these standards by requiring that certain baseline threats be addressed or certain formats be used. **Open Threat Model** formats – such as the JSON-based standard proposed by some tool vendors [10] – might converge into an industry-wide accepted schema. Achieving consensus on taxonomy is admittedly hard, but essential; initiatives like MITRE's Common Weakness Enumeration (CWE) or OWASP's Top Ten show it's possible to agree on classifications for common issues. Extending that to a full threat model structure (which includes assets, actors, controls, etc.) will lay the foundation for broad adoption and regulatory codification.

- **Graph-Based Modeling and Knowledge Reuse:** Modern systems and their security considerations form complex webs of relationships – ideal for graph representations. Embracing graph-based threat modeling can greatly enhance both the expressiveness and analyzability of models. Security leader Dinis Cruz has demonstrated the power of this approach: at Photobox, he refactored risk workflows into a graph database, representing security data as queryable nodes and relationships [18]. This allowed the security team to visualize how risks, mitigations, and system components interlink, and to query the "security graph" for patterns (for example, to find all systems lacking a mitigation for a certain threat). By managing threat model data as a graph, organizations can more easily keep it updated and connected to live data feeds (CI/CD pipelines, ticketing systems, etc.). Graph-based threat models also enable the use of **graph analytics** and even reasoning engines to infer missing links or potential attack paths. For instance, an automated reasoning system could traverse a graph to identify an unmitigated path from an external attacker node to a sensitive data node, flagging a gap in the threat model. This approach aligns with the idea of **"connected ontologies"** – multiple domain-specific ontologies (for web apps, for IoT, for cloud, etc.) can be linked, allowing a threat model to integrate knowledge from many sources. As Cruz's work on "Semantic OWASP" suggests, leveraging knowledge graphs can help customize and scale security knowledge across different contexts (e.g., using OWASP's vast knowledge base in a semantic way rather than static checklists). In practice, the industry will need to invest in tools and platforms that support graph-based threat modeling and make it easy to map system architecture into these models (perhaps via code annotation or architecture-as-code descriptions). The endgame is a rich interlinked dataset of security information that can be both human-readable and machine-analysable – a far cry from today's static diagram or spreadsheet that many threat models reside in.

- **Semantic Interoperability and Ontologies:** Closely related to standardization and graphs is the need for semantic interoperability – ensuring that when one system's threat model references a "SQL Injection" and another references an "Injection flaw on database," they are understood as the same concept. A formal ontology (or set of ontologies) for threat modeling provides this shared vocabulary. By defining concepts like *Threat*, *Asset*, *Mitigation*, *Actor* and specific instances (like *Threat:SQL_Injection* as a subclass of *Threat:Injection*), we allow different organizations and tools to map their terms to a common reference. This is already happening in research and some tooling: for example, academic efforts have introduced ontologies to automate threat modeling, where a system model (say a Data Flow Diagram) can be fed into an inference engine that, using ontological rules, generates likely threats and mitigations [17]. The OdTM framework explicitly envisions collecting and sharing security knowledge through domain-specific ontologies and using automatic reasoning to build threat models of systems [19]. What this means for making threat models a regulatory artifact is that a regulator or industry group could publish an official ontology that organizations must use or map to. Semantic web technologies (OWL/RDF) could allow published threat models to be linked data – for instance, a company's threat model file could link each threat to a URI identifying a threat type (perhaps hosted by OWASP or NIST), which provides the definition. This level of semantic richness would enable third-party analysis at scale: an auditor could aggregate all published threat models and automatically check which known threats are most commonly unmitigated, or correlate threats to incidents. It also helps with **tool interoperability**: a company might use Tool A internally, but when disclosing, export to the common format and ontology so that regulators or partners using Tool B can interpret it correctly. Achieving semantic interoperability will require cooperation across industry, government, and academia – but there is momentum, with projects like OWASP's and various research papers laying the groundwork for common ontologies for cybersecurity [17].

- **Automation and Tooling:** For threat model disclosures to be viable at scale, especially for companies with large or rapidly evolving products, automation is key. We need tooling that makes threat modeling efficient and integrates with development workflows. This includes **threat modeling as code** (where threat models are defined in a declarative form alongside the system code or architecture definitions), automated threat identification (using rule engines or AI to suggest threats based on system components), and continuous validation (ensuring the implemented code/configuration matches the assumptions in the threat model). Automation reduces the human burden of keeping models up to date and lowers the skill barrier so that even smaller companies can comply. In the future, we can envision AI assistants that parse architecture diagrams or even source code and produce an initial threat model draft, complete with suggested mitigations from known libraries – the security team then reviews and publishes this with adjustments. Another aspect of tooling is making the **published output user-friendly**: perhaps interactive web-based threat models where a reader can toggle views (executive summary vs. technical details) or query how a particular threat is handled. The existence of user-friendly tools will also help alleviate the trade-off we noted for consumers, by presenting information in digestible ways. Furthermore, automation can help generate the *sanitized* public version of a threat model from the internal version, by stripping or abstracting sensitive details according to predefined rules or tags (for example, auto-redacting server names or exact configurations, but leaving the fact that “a database stores personal data encrypted at rest” which is useful to readers). In summary, robust tooling and automation are the engines that will carry threat modeling from an artisan practice to an industrial practice ready for regulatory spotlight. As Dinis Cruz and others often highlight, the goal is to **embed security knowledge into workflows** – when that happens, producing a threat model (and keeping it updated) can be as natural as running test suites or CI/CD pipelines [20] .

- **Security and Privacy of Disclosures:** Finally, an important technical aspect is how to enable transparency without courting new risks. This includes methods like **partial disclosure, anonymization, and tiered access**. Technically, solutions could involve having public threat model data that references detailed internal data without fully exposing it. For example, a threat model might state “customer data is encrypted with industry-standard algorithms in transit and at rest” – which is informative – without publishing the actual encryption keys or specific network topology. Technical guidelines or standards can define what *must* be disclosed versus what should be omitted for safety. This might be aided by classification tools that scan a threat model for sensitive info before publication. Additionally, in some cases a *two-tiered disclosure* might be appropriate: a high-level public threat model for general transparency, and a more detailed version available under NDA to regulators or critical customers. Technologies like secure multi-party computation or federated audits could even allow verification of a threat model’s completeness without revealing everything publicly (though this gets complex). In essence, enabling “secure transparency” is itself a technical challenge – one that involves carefully balancing openness with protection (more on this balance in the next section). The good news is that threat models, by nature, discuss classes of threats and mitigations which are often *not* proprietary secrets. As one expert noted, *“Threats are mainly inherent for a software component by virtue of its functionality, so listing threats should not endanger valuable trade secrets. … you don’t have to detail the mitigation technology, just assert recognizing and addressing relevant threats.”* [21] . In other words, a well-designed disclosure reveals *what* you’re doing about security without necessarily revealing exactly *how*. Embracing that principle in standards and tools will be vital for adoption.

Partial Disclosure and Safe Transparency

A frequent concern is that publishing a threat model could hand attackers a “map” to exploit the system. It’s a valid point to address – **how do we enable transparency while maintaining security?** The answer lies in partial disclosure and careful anonymization, which allow organizations to share the *essence* of their security posture without revealing sensitive details that could aid an attacker. This approach is similar to how companies disclose financial risks: enough detail to inform investors, but not the exact combination to the safe.

The Role of Partial/Anonymized Threat Models: In practice, companies can maintain multiple versions of a threat model for different audiences [22] . The most detailed model, including granular system specifics, remains internal for engineering use. From that, organizations can derive a **sanitized version** for public disclosure. This sanitized threat model would focus on threats, impacts, and mitigations in general terms, omitting specific technical information like IP addresses, exact software versions, or other data that might facilitate an attack. As the *Designing Secure Software* initiative suggests, you might tag sections of the master threat model by audience, then compile a high-level overview for executives and customers, and a more technical (but still scrubbed) version for integrators or regulators [22] . Key to this approach is that the *threats and mitigations themselves* are not secret – indeed, they **shouldn't be**. If your product relies on "security by obscurity" (keeping the very existence of a vulnerability secret), that's a fragile strategy [23] . Instead, openly acknowledging a threat (e.g. "Threat: data exfiltration by insider") and stating that you have a mitigation (without necessarily detailing the exact monitoring rules) does not weaken your security – it strengthens trust. It's akin to a bank saying "We're threatened by robbers, so we have armed guards and vaults"; that doesn't tell the robber how to rob the bank, it just assures customers their money is protected.

Analogy to Open Source and Vulnerability Disclosure: There is an instructive analogy in the world of open source software and vulnerability disclosure. Releasing source code could, in theory, help attackers find flaws; yet it also enables a vast community to inspect and strengthen the code. Similarly, disclosing a vulnerability publicly could alert attackers, but it's accepted (with coordinated disclosure) because it leads to fixes and awareness. Publishing threat models sits somewhere in between – it's a proactive disclosure of *potential* issues and defenses. Yes, attackers could read it, but they likely learn little that they wouldn't already assume. Attackers usually know that, say, a web application might have XSS or SQL injection; a threat model confirming "we considered SQL injection and have input validation" doesn't give away a zero-day, it just tells the world the developers are mindful of that class of attack. As one commentary put it, *"open threat modeling should never be divulging any crown jewels, it's more like a 'spec sheet' for security considerations."* [21] . A spec sheet for a car lists the safety features (airbags, ABS, etc.) – useful to the buyer, and not particularly useful to a thief. Likewise, a security spec sheet (threat model) lists security features and risk assumptions. By carefully stripping out implementation specifics, companies can avoid providing a roadmap for attackers while still demonstrating due diligence.

Building Secure Transparency Practices: To implement partial disclosure effectively, organizations will develop guidelines and perhaps automated tools as mentioned. For example, a guideline might state: "Include threat categories, attacker types, potential impacts, and mitigations. Omit exact configurations, detection thresholds, or anything that would significantly aid in bypassing a control." In cases where a mitigation is sensitive (maybe a proprietary anomaly detection algorithm), the public model can state "anomaly detection in place" without detailing how. The assumption is that attackers, if sophisticated, will *assume* you have some detection anyway – saying so doesn't increase their chances. Meanwhile, honest stakeholders get confidence that you have considered the issue. Companies can also explicitly call out what is *not* addressed in the public model. For instance, *"Out-of-scope: this product is not designed to resist hardware tampering"* – such a statement can be important for user awareness (just as a food label might say "contains peanuts"). It's better to be transparent about exclusions than for users to find out the hard way. Indeed, a public threat model might include an **"accepted risks"** section for threats that remain (this can actually spur useful market discussion: perhaps a competitor will advertise that they *do* mitigate that risk, pushing others to improve).

The notion of having internal vs external threat models is already advocated by experts [24] . Internal models can be fine-grained and even include "attack playbooks" or red-team findings – things you'd never publish. External models focus on broad strokes. We can also leverage legal frameworks: for example, maybe only a summary is public, and regulators get to see a fuller version under confidentiality. Over time, as comfort grows, the line between what's public and private may shift (likely towards more public, as we realize it doesn't harm security). Companies may even decide to open up more once they see the benefits – similar to how open-source software went from fringe to mainstream as the benefits outweighed the perceived risks.

Addressing Legal and Competitive Fears: Partial disclosure helps with concerns beyond just security – it also addresses intellectual property and liability worries. By removing specifics, companies aren't giving competitors a blueprint of their system; they're merely sharing principles and measures that any competent firm *should* have. In terms of liability, one might worry "if we publish a threat model and later a breach happens via something not in it, are we exposed to lawsuits?" This is where careful wording and perhaps regulatory safe harbors will help: the threat model can include disclaimers that it's not exhaustive or is as of a certain date. Regulators could provide that a good-faith threat model disclosure won't be used to penalize a company unless there was gross negligence or deception. In fact, transparency can be *protection* against negligence claims: it shows you exercised due diligence in thinking about security. If anything, it's companies that hide their practices that suffer worse in court of public opinion after an incident ("what did they know and when? why didn't they at least warn users of this risk?"). With open threat modeling, the dialogue changes to a more collaborative tone – customers and third-party experts might even help improve your security by reviewing your threat model and providing feedback or spotting omissions (much like open source contributions).

In summary, **secure transparency is achievable**. By publishing threat models in a controlled, anonymized way, organizations can reap the benefits of market trust and informed stakeholders without materially increasing their attack surface. The process is akin to walking a path that other industries have walked: find the right level of abstraction where useful information is shared but sensitive details remain confidential. As the *"Flaunt your Threat Models"* article encapsulates, if done properly, sharing threat models *"amounts to security by transparency, not by obscurity – a stronger posture where you're not betting on secrecy to save you"*. Instead of hoping attackers never figure out your weak points, you're proactively shoring them up and confident enough to show your work. That cultural shift, from secrecy to openness, is at the heart of the vision for mandatory threat model disclosures.

Conclusion: From Vision to Reality – Next Steps

Mandating threat model disclosures as a norm will not happen overnight, but the path is becoming clear. This vision is both **ambitious and practical**: ambitious in that it foresees a world with radically higher security accountability, yet practical because it builds on patterns seen in other domains and incremental progress already underway. To transform this foresight into reality, a blend of community effort, industry initiatives, and regulatory action will be needed.

Industry and Community Initiatives: The threat modeling community can start by promoting voluntary transparency as a competitive advantage. Security-conscious companies should consider publishing (even partial) threat models for flagship products as a pilot, demonstrating the concept and learning the best practices of sanitization and presentation. Early movers can publicly "flaunt" their threat models to show leadership [25], much as some companies today publish transparency reports or open-source security tools for goodwill. Industry groups (like the OWASP Threat Modeling community, the Threat Modeling Connect conferences, etc.) can collaborate on developing the open standards discussed – finalizing a common threat model schema, perhaps through a consortium or an open project. Tool vendors should align on export/import formats (the work by IriusRisk on an Open Threat Model standard is a good example [10]). Additionally, creating reference libraries of threat models for common systems (analogous to design patterns or the OWASP Top Ten but for threat models) can help newcomers and provide templates to reduce the burden. Efforts by individuals like Dinis Cruz in pushing graph-based knowledge-sharing, and projects like OWASP OdTM, should be supported and extended – they are building the infrastructure on which this transparency will run. The community should also engage in educational outreach: training developers and security staff not just *how* to threat model, but *how to communicate* threat models to different stakeholders. This is a new literacy we must develop, akin to how the financial world had to develop investor relations communications.

Policymaker and Regulator Actions: Regulators and governments can start laying groundwork by incorporating threat modeling into existing frameworks. For example, the SEC's recent cybersecurity risk disclosure rules could be expanded in the future – today they require describing processes and incidents, but tomorrow they could encourage or mandate including a summary of the organization's threat assessment and mitigations (essentially a high-level threat model) in annual reports [26] [27]. Sector-specific regulators (energy grid, aviation, healthcare) might begin by asking companies to produce threat models internally as part of compliance, and eventually for critical areas consider public executive summaries of those models. Governments can also fund or endorse the development of standards (NIST, for instance, could publish guidelines on threat model disclosure and maybe pilot a "cyber nutrition label" program [14]). The idea of **security labels** is already gaining traction, especially for consumer IoT devices [28] [15] – threat model disclosure is a logical extension/upscale of that concept for more complex software and services. Lawmakers should explore liability protections to encourage transparency – analogous to how the SAFETY Act in the US provides some liability limits for approved security measures, we might provide safe harbor for those who do transparent threat modeling in good faith. Furthermore, regulators can convene industry panels to hammer out what a reasonable disclosure looks like in different contexts (the needs of a cloud service vs. a medical device will differ). Finally, an eye should be kept on international harmonization: cybersecurity is global, so aligning standards across major markets (e.g., US, EU, UK, Asia) will prevent fragmentation and reduce the burden on multinational firms.

Visionary but Incremental: The long-term vision painted here is one of a safer digital ecosystem where security is not a hidden attribute but a visible, comparable aspect of products. It draws on lessons from the past – when transparency took hold in finance and food, both industries saw improved outcomes and greater trust. Achieving the same for cybersecurity will require overcoming cultural inertia ("we've always kept security info secret") and addressing real challenges (standardization, not giving attackers undue info). But the benefits are compelling: an end to the lemons market in security, empowered consumers, and a race to the top for product security quality. The trajectory will likely be incremental: from a few companies voluntarily publishing threat models, to industry standards emerging, to partial regulatory adoption (perhaps in critical sectors or for large public companies first), and eventually broad regulation that solidifies the practice for all. Along this path, the threat modeling community has a critical role to play in advocating, prototyping, and refining the approach.

In conclusion, mandatory threat model disclosure represents a paradigm shift in how we think about cybersecurity accountability. It moves us toward a future where security due diligence is transparent and continuously improving through feedback loops. Much like a **food nutrition label** doesn't guarantee health but empowers choices, a **security threat model label** would empower stakeholders to choose and demand better security. The journey to get there will entail developing new standards and habits, but each step – whether it's adopting a common threat language, using graphs and ontologies for knowledge sharing, or simply deciding to publish a threat summary for your next software release – is a step toward correcting the imbalance of today's market. It is a future where **the norm is to share, not to hide, our understanding of threats**. In that future, attackers will face a more united and informed defense, and the market will reward those who do security right. The message to the industry is clear: let's start treating threat models not as private memos, but as public contracts of trust. The sooner we begin, the sooner the "market for lemons" in software can become a market of safety and assurance.

Sources:

- Schneier, Bruce. *A Security Market for Lemons.* Schneier on Security (2007) – Discusses how information asymmetry in cybersecurity allows inferior products to thrive [1] [8].
- Omer Singer. *Navigating a Market for Lemons* – Describes the impact of acute information asymmetry between security buyers and sellers [7].

- *Flaunt your Threat Models!* – Designing Secure Software (Brook Schoenfield, 2023) – Argues for publishing threat models as evidence of security due diligence [2] [21] and addresses concerns around disclosing too much [21] [24] .
- U.S. SEC, Securities Act of 1933 – Requires companies to disclose truthful financial information to investors [5] , establishing financial transparency.
- U.S. Nutrition Labeling and Education Act (1990) – Mandates standardized nutrition facts labels on food products, improving consumer information [6] .
- OWASP Ontology-Driven Threat Modeling (OdTM) Project – Provides a framework for formalizing and sharing security knowledge as ontologies, enabling automatic threat model generation [17] [19] .
- Dinis Cruz, *Creating a Graph-Based Security Organisation* (OWASP London, 2019) – Demonstrates using graph databases and hyperlinked security taxonomies to drive data-driven security decisions [18] .
- Consumer Reports & Carnegie Mellon IoT Security “Nutrition Labels” – Ongoing efforts to create easy-to-understand security labels for IoT devices, analogous to nutrition facts [14] [15] .
- SEC Cybersecurity Disclosure Rules (2023) – New regulations that begin to require companies to report on cybersecurity risk management and incidents [26] [27] , hinting at a trend toward more disclosure.

---

[1] [8] [9] A Security Market for Lemons - Schneier on Security
https://www.schneier.com/blog/archives/2007/04/a_security_mark.html

[2] [11] [12] [13] [16] [21] [22] [23] [24] [25] Flaunt your Threat Models! - Designing Secure Software
https://designingsecuresoftware.com/writings/flaunt/

[3] [4] [5] Securities and Exchange Commission - SEC, Definition & Purpose
https://www.history.com/articles/securities-and-exchange-commission

[6] Consumer Information and Labeling - Background | Economic Research Service
http://www.ers.usda.gov/topics/food-choices-health/consumer-information-and-labeling/background

[7] Navigating a Market for Lemons - by Omer Singer
https://www.omeronsecurity.com/p/navigating-a-market-for-lemons

[10] The Open Threat Model (OTM) Standard By IriusRisk
https://www.iriusrisk.com/the-open-threat-model-standard

[14] [15] [28] IoT Nutrition Labels - Innovation at Consumer Reports
https://innovation.consumerreports.org/initiatives/iot-nutrition-labels/

[17] [19] OWASP Ontology Driven Threat Modeling Framework | OWASP Foundation
https://owasp.org/www-project-ontology-driven-threat-modeling-framework/

[18] Creating a graph based security organisation - Apr 2019 (OWASP London chapter meeting) | PPT
https://www.slideshare.net/slideshow/creating-a-graph-based-security-organisation-apr-2019-owasp-london-chapter-meeting/140239010

[20] OWASP London | OWASP Foundation
https://owasp.org/www-chapter-london/

[26] SEC cybersecurity disclosure rules - KPMG International
https://kpmg.com/us/en/media/news/sec-cybersecurity-disclosure-rules-2024.html

[27] What the New SEC Regulation on Cyber Reporting Means for the ...
https://www.fairinstitute.org/blog/what-the-new-sec-regulation-on-cyber-reporting-means-for-the-risk-management-profession