threat model

Aditya Pangavhane

August 27, 2025

Abstract

In cybersecurity, finding vulnerabilities is not just about scanning or exploiting; it begins with structured steps that build awareness of the target system. After planning the approach and gathering information about the application, the next step is threat modeling. At this stage we shift perspective from attacker to protector: instead of asking "how can we break the system," we ask "where could the system break, and how can we defend it."

Threat modeling works like mapping the weak points of a house before a storm. We examine how attackers might attempt entry, what they could achieve, and the possible business impact if they succeed. This report studies established methodologies such as STRIDE, PASTA, OWASP threat categories, and NIST risk guidance. Each framework provides a structured way to recognize potential weaknesses in a web application before attackers exploit them.

We describe threats in everyday terms—such as using stolen credentials to impersonate a user or injecting malicious input to expose private data—and connect them directly to business risks like loss of user trust, downtime, or regulatory penalties. For each identified threat, we highlight realistic security controls including stronger authentication, data validation, encryption, and active monitoring. The purpose of this report is not only technical defense but also to provide non-technical stakeholders with clarity on why these risks matter. Threat modeling, therefore, becomes a preventive lens in the vulnerability assessment lifecycle, ensuring organizations stay one step ahead of attackers.

1 Introduction

Cybersecurity is not just a technical necessity, but a protective shield that safeguards individuals, organizations, and entire societies from the growing threat landscape of the digital world. In today's hyperconnected era, every action—from sending an email to managing critical infrastructure—relies on systems that are constantly under the radar of attackers. The need to adopt a defender's mindset, rather than simply a user's perspective, is at the heart of modern security practices. As Bruce Schneier aptly states, "Security is a process, not a product." [**?**]

The purpose of this document is to simplify the complex discipline of **threat modeling** and present it as an accessible tool for both security professionals and beginners. Instead of treating cybersecurity as a purely technical subject, this introduction emphasizes its role in shaping a mindset of awareness, vigilance, and protection.

1.1 Why Cybersecurity Matters

The significance of cybersecurity can be understood by observing how the threat landscape has evolved. According to the **2024 Verizon Data Breach Investigations Report**, over 70% of breaches involved human factors such as phishing, credential theft, and social engineering. These attacks demonstrate that vulnerabilities are not limited to machines, but also extend to people and processes.

- **Protecting Individuals:** Personal data breaches can lead to identity theft, financial loss, and emotional harm.
- **Protecting Organizations:** Businesses face reputational damage, loss of customer trust, and significant financial penalties.
- **Protecting Nations:** Critical infrastructure like power grids and healthcare systems can be paralyzed by cyberattacks, threatening national security.

1.2 Threat Modeling as a Defender's Tool

Threat modeling is the systematic process of identifying, evaluating, and mitigating potential threats before attackers exploit them. As Adam Shostack highlights in his book *Threat Modeling: Designing for Security* (2014), this process empowers defenders to think like attackers while staying ahead in the protection game.

Rather than perceiving threat modeling as a purely technical exercise, this document frames it as:

- A **mindset** that anticipates risks.
- A **framework** that organizes defensive strategies.
- A **bridge** that connects theory to real-world protection.

1.3 Scope of this Work

The chapters ahead will cover:

- A review of literature and existing approaches to threat modeling.
- A practical methodology simplified for beginners and professionals.
- Demonstrations and real-world analogies for clarity.
- Limitations, challenges, and future directions in integrating AI with threat modeling.

1.4 Conclusion of the Introduction

Cybersecurity cannot be achieved through tools alone—it demands awareness and preparedness. By exploring threat modeling through both academic research and practical analogies, this work aims to make readers not just informed users, but vigilant defenders in the digital age.

2 Background and Literature Review

2.1 Evolution of Threat Modeling

Threat modeling did not appear overnight—it is the result of decades of learning from system failures, breaches, and the growing sophistication of attackers. In its earliest form, organizations relied heavily on perimeter-based defenses such as firewalls and intrusion detection systems. The assumption was simple: if the network boundary is strong, internal systems remain safe. However, as technology expanded and threats became more complex, this model quickly proved insufficient.

The idea of systematically anticipating threats first gained traction in the 1990s, when security engineers at Microsoft began formalizing structured approaches to identify weaknesses in software. This gave birth to the STRIDE model (1999), introduced by Loren Kohnfelder and Praerit Garg. STRIDE categorized threats into six buckets: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. This was revolutionary because, for the first time, developers could reason about threats in a repeatable and structured manner rather than reacting to incidents post-fact.

By the early 2000s, academic researchers and security practitioners started extending the idea. Methodologies like OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation) emerged from Carnegie Mellon's Software Engineering Institute. Unlike STRIDE, which focused on software-level threats, OCTAVE was designed for organizational risk assessment—helping businesses align their technology risks with strategic objectives.

As cyberattacks became more targeted, especially with the rise of advanced persistent threats (APTs) in the 2010s, the need for more business-contextual models increased. This led to frameworks like PASTA (Process for Attack Simulation and Threat Analysis), which emphasize adversarial perspective and simulate how real attackers approach systems. Similarly, tools like Trike and VAST were designed to scale for enterprise-wide systems.

Over time, threat modeling has expanded beyond just IT security. With the growth of IoT, cloud services, mobile applications, and AI systems, organizations realized that anticipating threats must

adapt to diverse and rapidly evolving ecosystems. Today, it is not just a "developer's checklist," but a strategic practice embedded into the Secure Development Lifecycle (SDL), mandated by standards such as NIST, ISO 27001, and strongly promoted by communities like OWASP.In summary, threat modeling evolved from a defensive reactionary practice into a proactive, systematic discipline that combines technical rigor, attacker psychology, and business priorities.

2.2 Key Threat Modeling Frameworks

Over the years, several frameworks have been developed to guide organizations in systematically identifying and mitigating threats. Each framework reflects different priorities—some emphasize technical software flaws, while others focus on business risk or adversary behavior. The following are the most widely recognized frameworks in modern cybersecurity.

1. STRIDE

Origin: Developed by Microsoft in 1999.

Focus: Application-level threats.

Approach: STRIDE classifies threats into six categories:

Spoofing: Impersonating users or systems.

Tampering: Unauthorized modification of data or systems.

Repudiation: Denying actions without accountability.

Information Disclosure: Unauthorized access to confidential data.

Denial of Service: Disrupting availability of systems.

Elevation of Privilege: Gaining higher access rights than intended.

Use Case: Useful for developers and security engineers during the software design phase.

Limitations: Does not deeply address business impact or attacker motivation.

2. PASTA (Process for Attack Simulation and Threat Analysis)

Origin: Introduced by Tony UcedaVélez and Marco Morana (2012).

Focus: Risk-centric and attacker-centric.

Approach: PASTA follows seven stages, from defining business objectives to simulating attacks and identifying countermeasures. It emphasizes:

Business impact analysis.

Attack simulation based on real-world adversaries.

Prioritization of threats by risk levels.

Use Case: Ideal for organizations with high-value assets and regulatory concerns.

Limitations: Requires more time, resources, and skilled personnel than lightweight methods.

3. Trike

Origin: Created as a security auditing framework for risk management.

Focus: Risk-based approach with strong modeling of system requirements.

Approach: Trike builds threat models using:

Requirement Models: Define acceptable risk.

Attack Models: Represent possible attacks.

Risk Models: Assess threats against risk tolerance.

Use Case: Suitable for system auditors and security analysts needing quantitative risk analysis.

Limitations: Less popular compared to STRIDE and PASTA, with limited adoption in industry.

4. VAST (Visual, Agile, and Simple Threat)

Origin: Developed to integrate into Agile and DevOps environments.

Focus: Scalability and integration into modern software development pipelines.

Approach: VAST models both application threats and operational threats using visualization techniques.

Use Case: Large organizations with complex systems and rapid development cycles.

Limitations: Less detailed than PASTA in adversarial simulation, but easier to integrate into agile workflows.

5. OCTAVE (Operationally Critical Threat, Asset, and Vulnerability Evaluation)

Origin: Developed by Carnegie Mellon University's Software Engineering Institute.

Focus: Organizational and strategic risks.

Approach: OCTAVE emphasizes:

Asset identification.

Organizational risk assessment.

Prioritization of threats based on impact to mission and business objectives.

Use Case: Enterprises that need a broad organizational security strategy, not just technical threat modeling.

Limitations: Less suitable for detailed application-level threat analysis.

Comparative Insight

STRIDE is developer-friendly, great for application design.

PASTA is business-driven, great for high-stakes industries.

Trike is risk-focused, useful for auditors.

VAST is agile-oriented, scalable for enterprises.

OCTAVE is organizational, aligning threats with business missions.

2.3 Core Principles of Threat Modeling

References

References