**Exposé: LLM-unterstützte Threat-Modellierung mit DFDs**

**1. Motivation und Problemstellung**

Die rasante Entwicklung der Künstlichen Intelligenz (KI) verändert zahlreiche Lebens- und Arbeitsbereiche grundlegend. Insbesondere in der industriellen Produktion, im Zuge der sogenannten Industrie 4.0, führt der zunehmende Einsatz vernetzter digitaler Systeme zu einer tiefgreifenden Transformation.

Threat Modeling ist eine zentrale Technik zur frühzeitigen Identifikation von Sicherheitsrisiken in Softwaresystemen. In der Praxis ist dieser Prozess jedoch oft aufwendig, fehleranfällig und von der Expertise der Beteiligten abhängig. Der Einsatz von Large Language Models (LLMs), wie ChatGPT, verspricht hier eine neue Art der Unterstützung – insbesondere durch Kombination mit strukturierten Inputs wie Data Flow Diagrams (DFDs). Frühere Studien zeigen zwar Potenzial (Mbaka & Tuma, 2024), aber auch Herausforderungen wie Falsch-Positive oder mangelnde Kontexttiefe (z. B. PILLAR, Yang et al., 2024).

**2. Ziel der Arbeit**

Die Arbeit untersucht, wie verschiedene Stakeholder (LLM-Expert:innen, Software Engineer:innen und Security Expert:innen) kollaborativ mit einem LLM-gestützten System zur Threat-Modellierung interagieren können.

Ziel ist es, den **Prozess und Output** zu verbessern und **herauszufinden, wie eine vertrauenswürdige Mensch-Maschine-Interaktion konkret aussehen kann**.

**3. Forschungsfragen**

* Wie zuverlässig ist das Threat Modeling durch LLMs in Kombination mit DFDs?
* Welche Rolle spielen menschliche Experten in verschiedenen Phasen der Interaktion?
* Wie unterscheiden sich Anforderungen und Wahrnehmungen der drei Gruppen (LLM/AI, Security, Software Engineering)?

**4. Methodik**

* **Ein Szenarien**, inspiriert durch verwandte Arbeiten mit einem DFDs dargestellt:
* **Tools & Technik:**
  1. ChatGPT + STRIDE-Analyse
  2. Prompt Engineering (verschiedene Varianten: Zero-Shot, Few-Shot, Chain-of-Thought)
  3. Eigene Web-App mit integriertem Interface, DFD-Upload und LLM-Integration
* **Workshops:**
  1. **Diskussions-Workshop** (Experten aus allen drei Bereichen): Diskussion und Feedback zu Konzept, Methode und Tools der geplanten Studie mit (Audioaufnahme und Konzept Präsentation) [??.05.2025 Universität Bremen]
     + Ist das Vorgehen sinnvoll? Was fehlt?
     + Wo sind Herausforderungen aus Sicht von LLM / Security / Software Engineering?
     + Welche Verbesserungen oder Alternativansätze schlagen Sie vor?
     + Welche Tools sind geeignet – warum?
     + Wann ist menschlicher Input notwendig? Wo explizit?
     + Welche Prompt-Typen führen zu brauchbarem Output?
     + Unterschiede zwischen textueller Prompt-Eingabe vs. grafischer (DFD als Bild)?
     + Chain-of-Thought nutzbar? Wie könnte das aussehen?
     + Wie kann man Vertrauen in ein KI-basiertes System schaffen?
     + Welche Visualisierungen oder Formate wären hilfreich für die Kommunikation mit dem Tool (z. B. Text, DFD, Flowcharts, Tabellen)?
     + Welche Informationen fehlen Ihnen aktuell noch, um sich ein vollständiges Bild vom geplanten Prozess zu machen?
  2. Test-Workshop mit Interface-Prototyp: Evaluierung anhand von Aufgaben und Szenarien. (5-7 Developer)
     + Funktionalität schon gut?
     + Konzept verstehen von Developer?
* **Evaluation:**
  1. Qualitativ: Interviews,Beobachtungen (z. B. Reaktion & Feedback, Akzeptanz, Vertrauen, Interaktion)
  2. Quantitativ: Precision, Recall, False Positives, Bearbeitungszeit, LLM-Output-Bewertung durch Experten -> MACHE ICH NICHT!!

**5. Erwarteter Beitrag**

* Konkrete Designempfehlungen für kollaborative Interfaces zwischen Menschen und LLMs im Bereich Software Security.
* Praktikable Prompting-Strategien zur Verbesserung der Threat-Modellierung.
* Validierte Szenario-basierte Evaluierung für Forschung und Praxis.
* Theoretischer Beitrag zur Mensch-KI-Interaktion im sicherheitskritischen Kontext.

**6. Relevante Vorarbeiten (Auswahl)**

* **Mbaka & Tuma (2024)**: LLMs + DFDs verbessern Realismus der Bedrohungseinschätzung, aber führen zu mehr False Positives.
* **Yang et al. (2024)**: Vollautomatische LLM-gestützte DFD-Erstellung – aber mangelnde Präzision bei komplexen Systemen.
* **Chis et al. (2024)**: DFDs als Basis für semantische Sicherheitsanalysen – mit Wissensgraphen.
* **Mollaeefar et al. (2024)**: PILLAR erkennt Datenschutzbedrohungen automatisch, aber zeigt Defizite bei Kontextsensitivität.
* **Tuma & Scandariato (2018)**: Interaktive STRIDE-Analysen sind genauer, aber aufwendiger als isolierte.