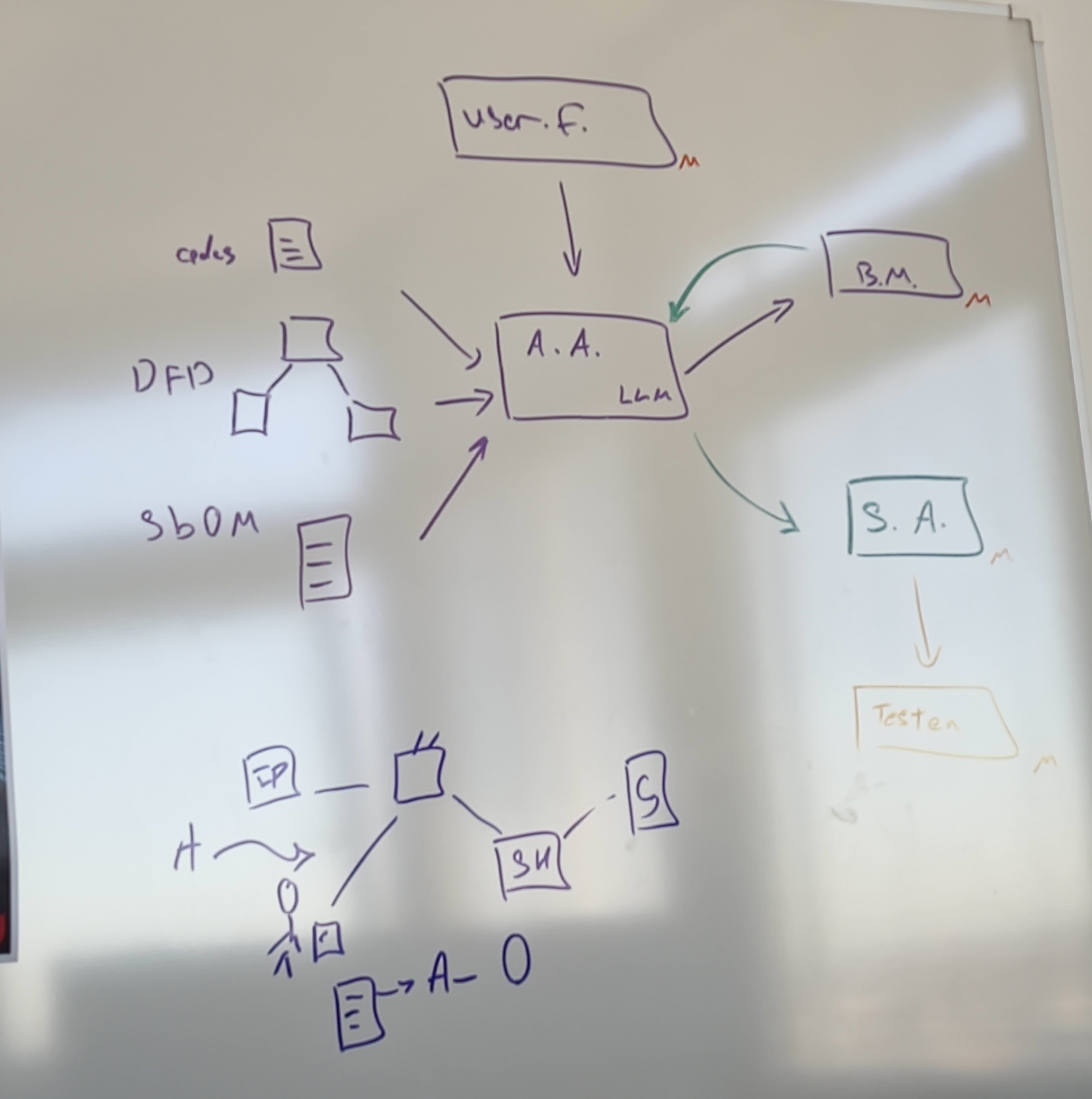
**Cybersecurity mittels DFD + LLM bessern**



[**Usefulness of data flow diagrams and large language models for security threat validation: a registered report**](https://arxiv.org/abs/2408.07537) <https://arxiv.org/pdf/2408.07537> [24.03.2025]

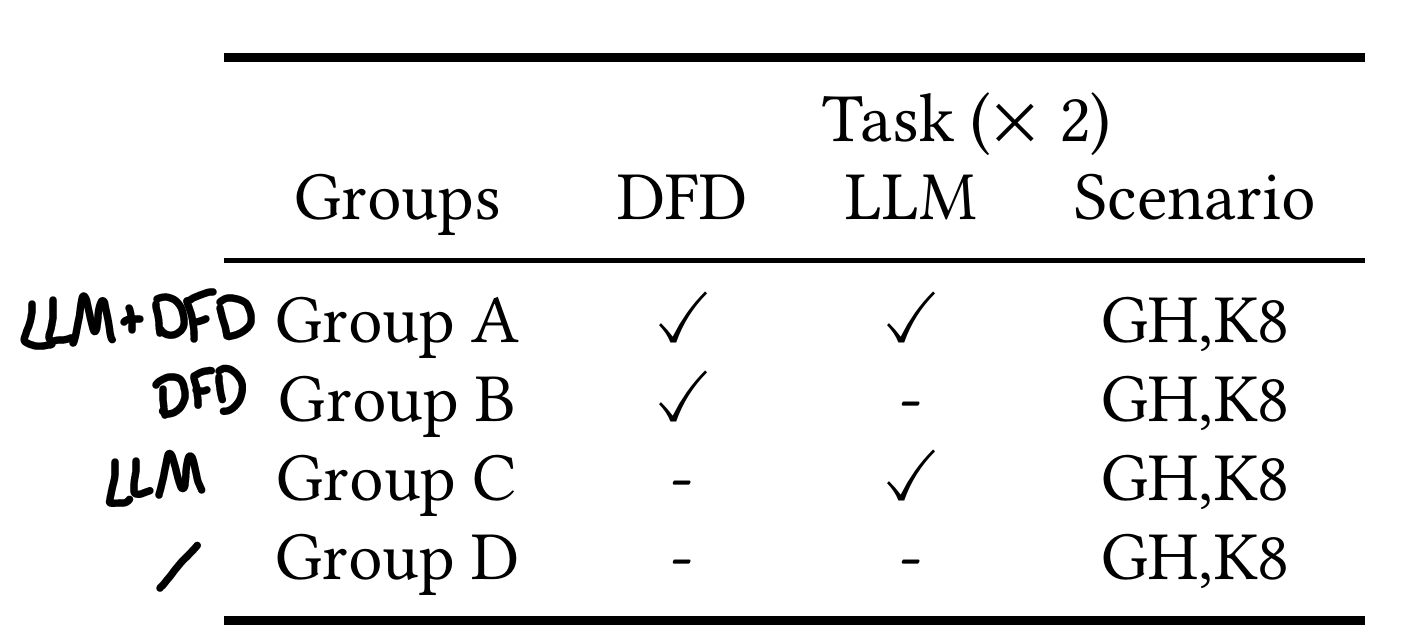
* Threat analysis and risk assessment are used to identify security
* Bisher wurde nicht untersucht, wie tief Analysten in das Material eintauchen müssen, um Sicherheitsbedrohungen effektiv zu validieren.
* ein kontrolliertes Experiment mit Praktikern (wie LLM-generierte Empfehlungen) vor, um zu prüfen, ob Analysematerial besser ist als keines und ob mehr Material (das DFD des Systems und LLM-generierte Empfehlungen) besser ist als wenig Material.
* Ein sicheres System zu bauen ist ein weltweit wichtiges Thema (EU Cybersecurity and Cyber-resilience Acts, and the US Cloud Act, ISO/SAE 21434:2021)
* Bedrohungsanalyse- und Risikobewertungsmethoden helfen, kritische Sicherheitsbedrohungen zu identifizieren und passende Gegenmaßnahmen zu finden.
* STRIDE nutzt das Datenflussdiagramm (DFD) zur grafischen Darstellung der Softwarearchitektur
* Sicherheits- und Fachexperten analysieren das DFD, um potenzielle Sicherheitsbedrohungen zu identifizieren -> zeitaufwendig und oft an Ressourcen mangelt.
* Die Bedrohungsanalyse bietet keine Vollständigkeitsgarantie -> häufig das Material erneut überprüfen, um sicherzustellen, dass keine Bedrohung übersehen wurde.
* Bei allgemeinen Bedrohungen wie SQL-Injection können Bedrohungen schnell bewertet werden, während bei spezialisierteren Bedrohungen das DFD und weitere Ressourcen erneut konsultiert werden müssen.
* Die Integration von LLMs mit DFDs könnte den Prozess der Bedrohungsanalyse automatisieren und beschleunigen.
* LLMs wurden untersucht, um Sicherheitsberatung zu liefern und Informationen zusammenzufassen.
* LLMs als Modell-in-the-loop verwenden, bei dem der Mensch basierend auf der LLM-Beratung entscheidet.
* Die Untersuchung soll klären, ob „mehr“ oder „weniger“ Analysematerial für die Bedrohungsvalidierung besser ist.
* DFDs könnten helfen, LLM-Ratschläge besser zu bewerten.
* Das Vertrauen in die erste LLM-Empfehlung könnte zu falsch-positiven Ergebnissen führen, während skeptische Validierung bessere Ergebnisse erzielt.
* Die Ergebnisse sollen helfen, das minimale Analysematerial für eine effektive Bedrohungsvalidierung zu bestimmen.

Andere Studien und Ergebnisse

* Neben STRIDE wurden auch andere Bedrohungsanalysetechniken empirisch untersucht.
* Zwei Studien verglichen die Wirksamkeit von Attack Trees und Misuse Cases.
* Karpati et al. fanden heraus, dass Angriffsbäume mehr Bedrohungen identifizierten als Misuse Cases, jedoch war der Unterschied nicht signifikant.
* Opdahl et al. bestätigten ähnliche Ergebnisse, wobei Attack Trees besser abschnitten, auch wenn die Studienaufbauten unterschiedlich waren.
* Keine der empirischen Studien zur Bedrohungsanalyse hat untersucht, wie menschliches Denken bei der Validierung von Sicherheitsbedrohungen unterstützt werden kann.
* Frühere Arbeiten haben die Nützlichkeit von LLMs bei der Erkennung von Code- und Software-Schwachstellen bewertet.
  + Omar et al. schlugen das Framework VulDetect vor, das die Stärken von GPT-2, BERT und LSTM nutzt, um Schwachstellen in C- und C++-Code zu erkennen, und erzielte dabei eine 92,65%ige Genauigkeit, die andere Techniken übertraf.
  + Cheshkov et al. untersuchten die Wirksamkeit von ChatGPT und GPT-3 bei der Schwachstellenerkennung.
  + In einer Studie von Sun und Kollegen wurde GPTScan (eine Kombination aus GPT und statischer Analyse) vorgeschlagen, um Logik-Schwachstellen in Smart Contracts zu erkennen.
  + Szabo und Bilicki evaluierten die Wirksamkeit von GPT-3.5 und GPT-4 bei der Erkennung der CWE-653-Schwäche und der Bestimmung von Schutzlevels für Front-End-Anwendungen.
  + Chen et al. untersuchten die Fähigkeit von ChatGPT und Bard, verbreitete Sicherheits- und Datenschutzmythen zu widerlegen, wobei sie berichteten, dass LLMs falsch-positive Ergebnisse liefern und Halluzinationen aufwiesen, wie das Bereitstellen ungültiger URLs.

IHRE STUDIE

* In der Studie statt den Teilnehmern mit der Entwicklung eines DFDs zu beauftragen, stellen wir ihnen ein vorbereitetes DFD zur Verfügung.
* Durch frühere Ergebnisse motiviert
* Herausforderungen: Reproduzierbarkeit von Bedrohungsanalysen, das Fehlen einer klaren „Definition of Done“, Validierung von Bedrohungen -> verlangsamen den Pozess
* Bedrohungsvalidierung ist nicht nur in der Entwurfsphase ein Problem, sondern auch während der Sammlung von Bedrohungsinformationen, wo die gesammelten Daten durch Unsicherheit beeinträchtigt sind.
* Erste Forschungsfrage: Was ist der tatsächliche Nutzen von zusätzlichem Material wie DFD oder LLMs während der Bedrohungsvalidierung?
  + Um den Nutzen zu messen, vergleichen wir die Leistung von Teilnehmern, die ohne zusätzliches Material validieren, mit denen, die DFD oder LLM erhalten haben, sowie mit denen, die beide Materialien erhalten.
  + Hypothese: Der tatsächliche Nutzen ist unterschiedlich zwischen den Gruppen ohne und mit zusätzlichem Material. Das Vorhandensein von zusätzlichem Material (DFD oder LLM) sollte die Leistung der Teilnehmer verbessern, und die Kombination beider Materialien sollte die Fähigkeit der Teilnehmer erhöhen, realistische Bedrohungen zu identifizieren.
* Zweite Forschungsfrage: Die wahrgenommene Nützlichkeit von zusätzlichem Material während der Bedrohungsvalidierung.
* Vergleichen der Durchschnittswerte der Gruppen, die nur DFD oder nur LLM erhalten haben, um die wahrgenommene Nützlichkeit zu messen und erwarten, dass diese statistisch äquivalent sind.
* ***Zielgruppen***: Informatik Master-Studenten, die an Kursen der Versuchsleiter teilnahmen. Die endgültige Studie wird mit Berufsfachleuten von Crowdsourcing-Plattformen wie Upwork oder Prolific durchgeführt. Teilnehmer mit Hintergrund in Softwareentwicklung und Cybersicherheitsrisikomanagement
* Ein Think-Aloud-Protokoll -> qualitative Rückmeldungen und potenzielle Probleme vor der endgültigen Datenerhebung zu identifizieren.
* Das Experiment hat ethische Genehmigung ; Personenbezogene Daten werden vor der Analyse entfernt



* + Zwei Szenarien: Ein Update eines Remote-Repositories auf GitHub und Pod-Deployment auf Kubernetes. Diese Aufgaben sind typisch in der Softwareentwicklung und basieren auf realen Open-Source-Plattformen.
  + LLM-Auswahl: ChatGPT-3.5 Turbo
  + Training: Ein 4,5-stündiges Training, bestehend aus einer Einführung in Threat Modeling, einer vertieften Schulung zu Sicherheitsbedrohungen und STRIDE sowie einem Walk-through der Aufgaben. Kontrollfragen zur Aufgabenverständlichkeit, Zeitressourcen und Schulungsqualität gestellt, sowie Aufmerksamkeits- und Hintergrundchecks durchgeführt, um sicherzustellen, dass die Teilnehmer die nötige technische Kompetenz besitzen
  + Ground Truth: Die Bedrohungen wurden absichtlich in korrekte und falsche unterteilt. Ein Beispiel für eine falsche Bedrohung wäre eine fehlerhafte Annahme über GitHub-Sicherheitslücken, die durch Branch Protection Rules verhindert werden.
  + Wahrgenommene Nützlichkeit: Die Teilnehmer bewerten auf einer 5-Punkte -Skala, wie hilfreich zusätzliche Materialien wie DFD oder LLMs bei der Bedrohungsvalidierung waren.
* Pilot Study Execution:
  1. Zuweisung: Jeder Teilnehmer wurde zufällig einer der vier Gruppen (A, B, C, D) zugewiesen.
  2. Jeder Teilnehmer erhielt zwei zufällig präsentierte Szenarien: GitHub-Repository-Update und Kubernetes-Pod-Deployment.
  3. Bedrohungsbewertung: Teilnehmer mussten die Korrektheit von 10 Bedrohungen (5 real, 5 falsch) beurteilen und diejenigen als realistisch markieren, die sie für wahrscheinlich hielten.
  4. LLM-Interaktion: Teilnehmer in den Gruppen A und C nutzten LLMs zur Bedrohungsvalidierung. Ihre Interaktionen mit den LLMs wurden aufgezeichnet.
* Ergebnisse:
  + Teilnehmer, die ein LLM nutzten, hatten eine höhere Erfolgsquote bei der Identifizierung realistischer Bedrohungen (A: 8,1, C: 9,4), aber auch mehr Falsch-positive (A: 7,4, C: 4,7).
  + Interessant war, dass Gruppe D (ohne Zusatzmaterial) wenig Falsch-positive berichtete, aber immer noch eine hohe Anzahl realistischer Bedrohungen korrekt identifizierte.
  + Teilnehmer, die LLMs nutzten, neigten dazu, den ersten Vorschlag des LLMs zu akzeptieren, was zu mehr Falsch-Positiven führen könnte, insbesondere bei weniger erfahrenen Teilnehmern.
* Vorläufige Beobachtungen:
  + Die Verwendung von DFDs (Datenflussdiagrammen) hatte keinen signifikanten Einfluss auf die Identifikation realistischer Bedrohungen.
  + Es wurde ein Trend beobachtet, dass weniger Analyse-Material möglicherweise ebenso effektiv sein kann wie mehr.

**PILLAR: an AI-Powered Privacy Threat Modeling Tool**

<https://arxiv.org/pdf/2410.08755> [24.03.2025]

* Die rasante Entwicklung großer Sprachmodelle (LLMs) hat neue Möglichkeiten für den Einsatz künstlicher Intelligenz in einer Vielzahl von Bereichen eröffnet, einschließlich des Privacy Engineerings.
* Da zunehmend sensible Nutzerdaten verarbeitet werden, ist der Schutz der Privatsphäre wichtiger denn je.
* Sie erfordern oft erheblichen manuellen Aufwand, Expertenwissen und detaillierte Systemkenntnisse.
* Dies macht den Prozess zeitaufwändig, umsändlich und fehleranfällig.
* Aktuelle Methoden, wie LINDDUN, basieren auf der Erstellung und Analyse komplexer Datenflussdiagramme (DFDs und Systembeschreibungen, um Datenschutzprobleme zu identifizieren.
* Es werden lange Listen of Probleme generiert ohne Anleitung bzw Sortierung nach Priortät
* PILLAR ist ein Tool, das LLMs mit dem LINDDUN-Framework kombiniert, um die Modellierung von Datenschutzbedrohungen zu vereinfachen. Es automatisiert Schritte wie die Erstellung von DFDs, die Klassifizierung von Bedrohungen und die Priorisierung von Risiken. Dadurch wird der Aufwand für Entwickler und Datenschutzexperten reduziert und die Effizienz sowie Genauigkeit verbessert.

PILLAR:

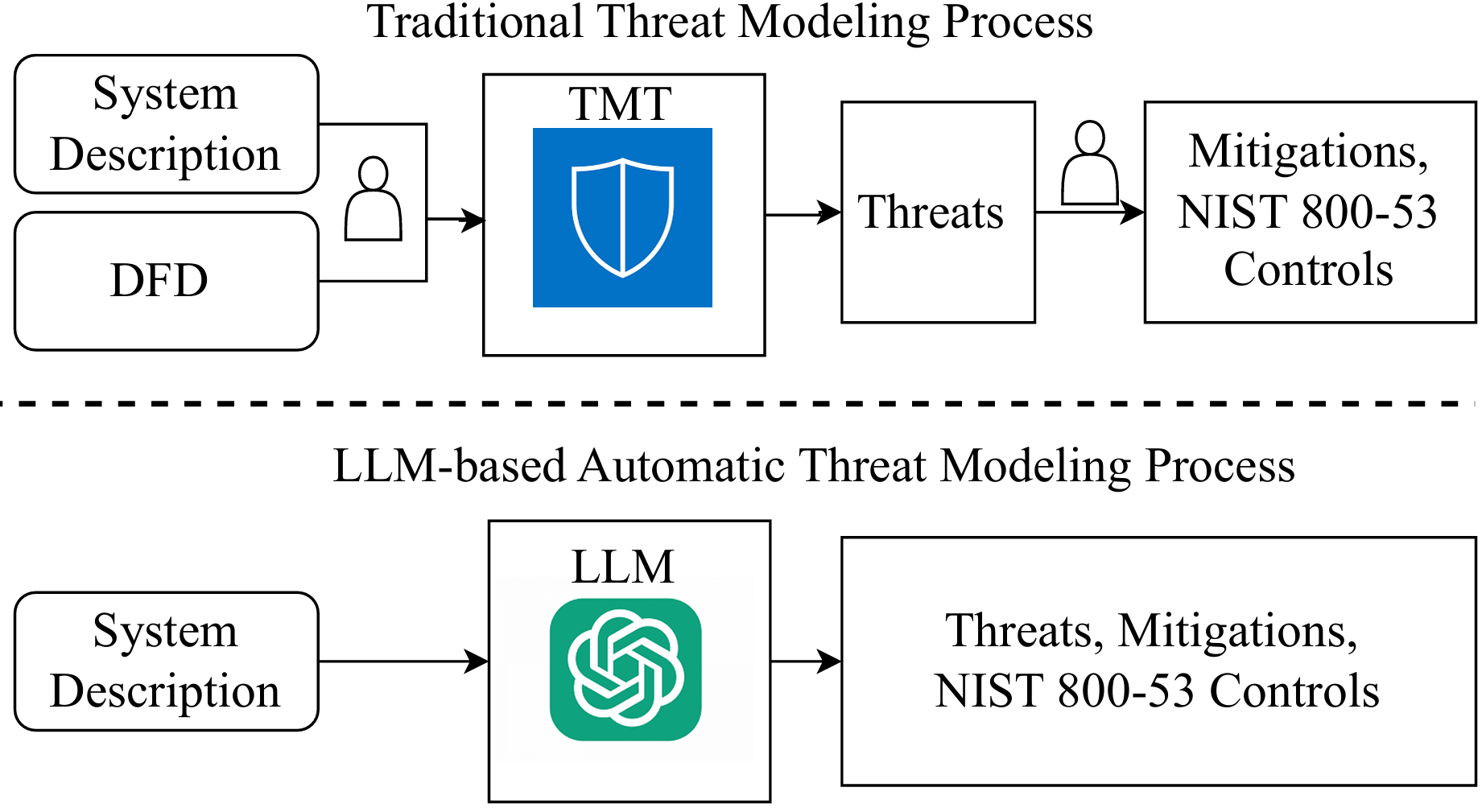
* Das Tool PILLAR verwendet LLMs zur automatisierten Bedrohungsmodellierung und analysiert Datenschutzrisiken anhand von Datenflussdiagrammen (DFDs).
* Es identifiziert potenzielle Datenschutzbedrohungen, indem es Datenflüsse, Akteure und Prozesse innerhalb eines Systems untersucht.
* Die Nutzung von LLMs ermöglicht eine schnelle Analyse und Kategorisierung von Bedrohungen in einem DFD.
* PILLAR integriert Bedrohungsbibliotheken und bestehende Sicherheitsstandards in die DFD-Analyse.
* Entwickelt für Unternehmen und Entwickler, die Datenschutz und Sicherheit in ihre Softwarearchitektur integrieren möchten.
* Durch maschinelles Lernen verbessert PILLAR kontinuierlich seine Bedrohungserkennung basierend auf neuen Sicherheitsvorfällen.
* Das Tool visualisiert die Datenflüsse zwischen Systemkomponenten, um Schwachstellen einfacher zu identifizieren.
* Es generiert automatisierte Empfehlungen zur Verbesserung der Datenschutzmaßnahmen auf Basis der DFD-Analyse.
* PILLAR analysiert DFDs, erkennt potenzielle Sicherheitslücken und schlägt Sicherheitsmaßnahmen vor.
* Die Analyse basiert auf etablierten Bedrohungsmodellen wie STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege).
* PILLAR generiert eine Risiko-Kategorisierung für jede erkannte Bedrohung.
* Entwickler können mit PILLAR Bedrohungen frühzeitig erkennen und ihre Sicherheitsarchitektur entsprechend anpassen.
* Es kann mit Bedrohungsbibliotheken und Compliance-Frameworks wie GDPR, NIST und ISO 27001 arbeiten.
* Das Tool integriert sich in Softwareentwicklungspipelines und ermöglicht kontinuierliche Sicherheitsanalysen.
* Das Tool analysiert folgende Elemente in DFDs:
  + **Datenquellen und -senke**n (woher kommen die Daten, wohin gehen sie?)
  + **Prozesse** (wie werden die Daten verarbeitet?)
  + **Datenflüsse** (welche Pfade nehmen die Daten im System?)
  + **Sicherheitsmaßnahmen** (welche Schutzmaßnahmen existieren bereits?)

SCHWACHSTELLEN:

* Falsch-positive & falsch-negative Ergebnisse: Relevante Bedrohungen werden übersehen oder harmloses Verhalten als Gefahr eingestuft.
* Begrenztes Verständnis komplexer Systeme: LLMs haben Schwierigkeiten, tiefere Kontextabhängigkeiten in komplexen Architekturen zu erkennen. Besonders bei verteilten Systemen oder Microservices ist die Bedrohungsanalyse unzuverlässig.
* Mangelnde Transparenz: Entscheidungen der KI sind oft nicht nachvollziehbar.
* Datenschutzrisiken: Gefahr von Datenlecks durch unsichere Modellnutzung.
* Nicht immer konforme Maßnahmen: Sicherheitsvorschläge sind teils unpassend für branchenspezifische Standards. Manchmal schlägt PILLAR allgemeine Lösungen vor, die nicht in jedem Kontext anwendbar sind

**ThreatModeling-LLM: Automating Threat Modeling using Large Language Models for Banking System**

<https://arxiv.org/pdf/2411.17058> [24.03.2025]



* Untersuchung: LLMs werden genutzt, um Bedrohungsmodelle auf Basis von DFDs automatisch zu erstellen und zu verbessern.
* Fokus auf die Reduzierung des manuellen Aufwands bei der Erstellung und Pflege von Threat Models.
* LLMs analysieren Systemdokumentationen und Code, um DFDs automatisch zu generieren.
* Dadurch entfällt der manuelle Prozess der Diagrammerstellung, was Zeit und Ressourcen spart.
* Die Modelle analysieren Datenflüsse zwischen Systemkomponenten und erkennen sicherheitskritische Schnittstellen.
* LLMs helfen dabei, Sicherheitsrisiken zu klassifizieren und potenzielle Angriffsvektoren zu identifizieren.
* Durch die Automatisierung von DFD-Analysen können Unternehmen Bedrohungen schneller erkennen und darauf reagieren.
* LLMs werden zur Unterstützung von Sicherheitsanalysten eingesetzt, indem sie mögliche Bedrohungsszenarien aus den DFDs ableiten.
* Die Nutzung von KI reduziert den manuellen Aufwand bei der Erstellung und Aktualisierung von Bedrohungsmodellen.
* Sicherheitsmaßnahmen werden anhand der erkannten Risiken optimiert, indem das System kontinuierlich trainiert wird.
* LLMs verbessern bestehende Threat-Modeling-Techniken, indem sie große Mengen an Sicherheitsdaten auswerten.

SCHWACHSTELLEN:

* Fehlklassifizierung von Bedrohungen: Risiken werden falsch priorisiert.
* Unzuverlässige DFD-Erstellung: Falsche oder unvollständige Systemdarstellungen.
* Mangel an Domänenwissen: Branchenspezifische Bedrohungen werden nicht immer erkannt.
* Overfitting: Erkennung basiert oft nur auf bekannten Angriffsmustern. Neue oder seltene Angriffsstrategien werden möglicherweise nicht erkannt.
* Veraltete Bedrohungsanalyse: Fehlende Echtzeit-Updates für neue Angriffsmethoden.
* Notwendigkeit menschlicher Überprüfung: LLMs können Sicherheitsanalysten nicht vollständig ersetzen.

**PDF „Sicherheitsbewertung von ChatGPT 4.0-generiertem Programmcode auf Basis von Extended Static Checking“ [31.03.2025]**

* 92% der Befragten gaben an, ein AI-Coding-Tool beruflich oder privat zu nutzen, 70% sahen darin signifikante Vorteile
* Sicherheitsziele: Vertraulichkeit = nur autorisierte Personen lesenden
* Zugriff auf bestimmte Informationen; Integrität der Daten = Daten sollen vor unautorisierter und unbemerkter Veränderung geschützt werden

**PDF “Uncover Security Design Flaws Using The STRIDE Approach” [31.03.2025]**

* Microsoft verteidigt deren Systeme mit „threat modelling“ – eine methodische Betrachtung/ Überprüfung des Systems
* Es gibt viele Methoden threat modelling anzugehen
* Können eigentlich nicht versichern das ein System sicher ist, aber es ist möglich aus Fehlern zu lernen
* Wichtig Fehler zu finden, die der Sicherheit schaden, damit Hacker/ Attacker diese nicht ausnutzen können

WAS BEDEUTET SICHERHEIT IM INTERNET?

* Softwareentwickler betrachten Codequalität als Sicherheit.
* Netzwerkadministratoren denken an Firewalls, Incident Response und Systemmanagement.
* Akademiker orientieren sich an den Designprinzipien von Saltzer und Schroeder sowie an Sicherheitsmodellen:
  + Gehe davon aus, dass Angreifer über den Quellcode und die Spezifikationen verfügen.
  + Halte es einfach.
  + Erlaube keine Operation basierend auf einer einzigen Bedingung.
  + Überprüfe alles, jedes Mal.
  + Sei vorsichtig mit gemeinsam genutzten Ressourcen.
* Sicherheit = Systeme die folgende Eigenschaften haben:
  + **Vertraulichkeit:** Daten sind nur für die vorgesehenen Personen zugänglich.
  + **Integrität:** Daten und Systemressourcen werden nur auf angemessene Weise von berechtigten Personen geändert.
  + **Verfügbarkeit**: Systeme sind bei Bedarf einsatzbereit und arbeiten mit akzeptabler Leistung.
  + **Authentifizierung**: Die Identität der Benutzer wird festgestellt (oder anonyme Benutzer werden akzeptiert).
  + **Autorisierung**: Benutzer erhalten explizit Zugriff auf Ressourcen oder dieser wird ihnen verweigert.
  + **Nichtabstreitbarkeit**: Benutzer können eine Aktion nicht ausführen und später bestreiten, sie durchgeführt zu haben.
* Threat Modelling mit STRIDE, um diese Eigenschaften zu gewährleisten
  + Spoofing - Authentication
  + Tampering - Integrity
  + Repudiation - Non‐repudiation
  + Information disclosure - Confidentiality
  + Denial of service - Availability
  + Elevation of privilege - Authorization
* Teile dein System in Komponenten auf und mindere alle Bedrohungen für jede einzelne Komponente – dann kannst du argumentieren, dass das System sicher ist.
  + Nur weil induviduelle Komponente immun sind, bedeutet das nicht, dass mehrer oder alle Komponente zusammen immun sind
* Bedrohnung erscheinen, wenn Systeme zu einem zusammengefügt werden

WAS SIND DFDs?

* Datenflussdiagramme (DFDs) werden typischerweise verwendet, um ein System grafisch darzustellen: Datenflüsse, Datenspeicher, Prozesse, Interaktionen und Vertrauensgrenzen.
* Das richtige Erstellen des DFDs ist entscheidend, um das Bedrohungsmodell korrekt zu gestalten.

**Meine Studie:**

Versuchen prozess zu verbessern (aus den papers).

Immer einen Experten dabei haben (Experteninterview: Software Sicherheit Experten).

Zwei, drei verschiedene Scenarios (In Buisness scenariom mit client wie einkaufen, web anwendung mit Datenbank (DFD) oder IoT Contexr)

1. Welches Tool sollte man benutzten?
2. Prompt Engeenierging output evaluieren.

Variante a, b, c (Scenarien aus Paper inspo, parallel tools vorbereiten) Zero-Trust, Trust over Trust

Tool Chat-GBT, DFD oder STRIDE

Ziel: Wollen zeigen der Prozess funktioniert. Das output bei input funktioniert.

Welche dieser Variante wäre schon am besten.

* Wie haben sie Daten vorbereitet?
* Welche tools benutzt?
* Was war der output?

Scenario – Threat Modelling – Output

Chat-GBT bekommt DFD (und erklärt die DFD richtig) -> man erklärt ihm ein scenario und Chat-Gbt gibt ein Threat Modelling

Was muss man als Teinehmer (hier Experte oder Grundwissen) wissen bzw. drauf achten, um mit diesem Tool zusammen zu arbeiten?

Wann, an welcher Stelle muss eine Person dabei sein, muss eine dabei sein um dieses Tool

**Meine Software:**

Eine Website mit prompt engeeniering + Chat GBT + Datenbank

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Paper | Datum | Hypothese | Scenario | Ergebnis |
| [Usefulness of data flow diagrams and large language models for security threat validation: a registered report](https://arxiv.org/abs/2408.07537) | W. Mbaka, Katja Tuma  15. August 2024, arXiv.org | Den tatsächliche Nutzen ist unterschiedlich zwischen den Gruppen ohne und mit zusätzlichem Material. Das Vorhandensein von zusätzlichem Material (DFD oder LLM) sollte die Leistung der Teilnehmer verbessern, und die Kombination beider Materialien sollte die Fähigkeit der Teilnehmer erhöhen, realistische Bedrohungen zu identifizieren | Teilnehmer mit Hintergrund in Softwareentwicklung und CybersicherheitsrisikomanagementEin Think-Aloud-Protokoll  Ein 4,5-stündiges Training, bestehend aus einer Einführung in Threat Modeling, einer vertieften Schulung zu Sicherheitsbedrohungen und STRIDE  1. Zuweisung: Jeder Teilnehmer wurde zufällig einer der vier Gruppen (A, B, C, D) zugewiesen.  2. Jeder Teilnehmer erhielt zwei zufällig präsentierte Szenarien: GitHub-Repository-Update und Kubernetes-Pod-Deployment.  3. Bedrohungsbewertung: Teilnehmer mussten die Korrektheit von 10 Bedrohungen (5 real, 5 falsch) beurteilen und diejenigen als realistisch markieren, die sie für wahrscheinlich hielten.  4. LLM-Interaktion: Teilnehmer in den Gruppen A und C nutzten LLMs (ChatGPT-3.5 Turbo ) zur Bedrohungsvalidierung. Ihre Interaktionen mit den LLMs wurden aufgezeichnet. | Teilnehmer, die ein LLM nutzten, hatten eine höhere Erfolgsquote bei der Identifizierung realistischer Bedrohungen (A: 8,1, C: 9,4), aber auch mehr Falsch-positive (A: 7,4, C: 4,7).  Interessant war, dass Gruppe D (ohne Zusatzmaterial) wenig Falsch-positive berichtete, aber immer noch eine hohe Anzahl realistischer Bedrohungen korrekt identifizierte.  Teilnehmer, die LLMs nutzten, neigten dazu, den ersten Vorschlag des LLMs zu akzeptieren, was zu mehr Falsch-Positiven führen könnte, insbesondere bei weniger erfahrenen Teilnehmern. |
| -> Reference (Stage 1):  A Knowledge Graph Approach to Cyber Threat Mitigation Derived from Data Flow Diagrams | Andrei Chis, Oliviu Ionut Stoica, Ana-Maria Ghiran, R. Buchmann  14. Juni 2024, International Conference on Automation, Quality and Testing, Robotics | KEIN ZUGRIFF AUF VOLLSTÄNDIGEN PAPER! | Ein Design-Science-Projekt nutzt Wissensgraphen, um Cybersecurity-Risiken frühzeitig zu erkennen und bestehende Systeme zu prüfen. Durch den Einsatz großer Sprachmodelle (LLMs) werden DFDs in Wissensgraphen umgewandelt, was semantische Analysen und bessere Entscheidungen ermöglicht. | Datenflussdiagramme (DFD) helfen bei der Analyse und Gestaltung von Unternehmenssystemen, insbesondere beim Übergang in die Cloud. Sie unterstützen dabei, Prozesse zu verstehen und sicherheitsrelevante Schnittstellen zu erkennen. |
| -> Reference (Stage 1):  A modeling approach to cyber threat mitigation | Andrei Chis, Oliviu Ionut Stoica, Ana-Maria Ghiran  2024, BIR Workshops | Die Anwendung von „secure by design“-Prinzipien während der Entwicklungsphase eines IT-Systems kann dazu beitragen, Cyber-Bedrohungen frühzeitig zu erkennen und zu mindern.  Die Verwendung von Modellierungssystems, Datenflussdiagrammen (DFDs) und Bedrohungsmodellen,wie STRIDE; soll dabei helfen, sowohl technische als auch nicht-technische Stakeholder eines Systems in die Sicherheitsanalyse einzubeziehen und Sicherheitsbedrohungen zu identifizieren | Ein **online Shop**, in dem verschiedene **Geschäftsprozesse** wie **Produkteinführung**, **Kundenanmeldung**, **Produktwahl**, **Bestellabwicklung** und **Datenbankabfragen durch einen Manager** modelliert werden. Diese Prozesse werden in einem **Datenflussdiagramm (DFD)** abgebildet.  Die Sicherheitsbewertung jedes Prozesses erfolgt anhand der **Mitigationsmaßnahmen**, die auf die identifizierten Bedrohungen angewendet werden. Für den **DoLogin-Prozess** beispielsweise werden **zwei Mitigationen** (wie **angemessene Authentifizierung** und **Geheimnisse nicht speichern**) bewertet, um zu bestimmen, ob sie ausreichend sind, um die Bedrohung **Spoofing Identity** zu adressieren. | **Datenflussdiagramme (DFDs)** und **Bedrohungsmodelle** sind wichtige Werkzeuge für die **Identifizierung von Sicherheitsrisiken** und die **Entwicklung von Sicherheitslösungen** in der Entwurfsphase eines Systems.  die Kombination von **Datenflussdiagrammen** und **Bedrohungsmodellierungsansätzen** ermöglicht eine Berechnung von Sicherheitscores eine objektive Bewertung der Sicherheitsmaßnahmen eines Systems. Mit dem proposierten Modellierungsansatz können Unternehmen ihre Sicherheitsstrategie kontinuierlich verbessern.  Aber es kann gewisse Bedrohungen übersehen und diese Modellierung kann zu komplex werden, wenn mehrer Bedrohungmethoden verwendet werden sollen.1 |
| -> Reference (Stage 1):  Human Aspect of Threat Analysis: A Replication | Katja Tuma, W. Mbaka  2. August 2022, arXiv.org | die menschlichen Aspekte, insbesondere Faktoren wie Geschlecht, Herkunft und Nationalität, einen Einfluss auf die Ergebnisse der Bedrohungsanalyse haben. | kontrollierte, explorative Experimente, die darauf abzielen, die Auswirkungen menschlicher Faktoren auf die Bedrohungsanalyse zu untersuchen. Die Experimente werden in einer akademischen Umgebung durchgeführt und beinhalten eine differenzierte Replikation früherer Studien zur Bedrohungsanalyse mit der STRIDE-Methode. | Die Studie wurde detailiert geplant, aber nicht ausgeführt, sodass keine tatsächlichen Resultaten vorliegen, sondern nur die Vermutungen.  (Relatet Work! Worth mentioning) |
| -> Reference (Stage 1): Towards Automating a Risk-First Threat Analysis Technique | Karanveer Singh, Margit Saal, Andrius Sakalas  18. November 2019 | die Automatisierung der eSTRIDE-Methodik (eSTRIDE uses extended Data Flow Diagrams (eDFDs) Tuma et al.) durch ein Prototyp-Tool den manuellen Aufwand der Bedrohungsanalyse verringern, die Produktivität steigern und die Genauigkeit (Präzision) und Vollständigkeit (Recall) der Bedrohungsanalyse unterstützen können | Ein Prototyp-Tool wurde entwickelt, das der Design-Science-Forschung folgt. Das Tool sollte den Nutzern helfen, erweiterte Datenflussdiagramme (eDFDs) zu erstellen oder zu ändern und die eSTRIDE-Bedrohungsanalyse durchzuführen.  Die Evaluation des Tools wurde in Formvon zwei Workshops mit zehn Teilnehmern (Studierende mit wenig bis keiner Erfahrung in der Bedrohungsanalyse) durchgeführt, bei dem die Teilnehmer das Tool nutzten, um eine Bedrohungsanalyse durchzuführen, einmal manuell und einmal mithilfe des Tools. | Präzision der Teilnehmer, die das Tool verwendeten, lag bei 74,8 %, was im Vergleich zur durchschnittlichen Präzision von 60 % in früheren Studien mit manuellen Methoden (z.B. Tuma und Scandariato) eine ähnliche Leistung darstellt.  Die durchschnittliche Recall[[1]](#footnote-1)-Rate der Teilnehmer, die mit dem Tool arbeiteten, betrug 58,5 %, was leicht unter dem Wert von 62 % lag, der für manuelle STRIDE-Analysen berichtet wurde.  Produktivität zeigte die Gruppe, die das Tool verwendete, eine durchschnittliche Rate von 1,4 Bedrohungskategorien pro Minute, was signifikant schneller war als die manuelle Methode, bei der die Produktivität auf 1,2 Bedrohungen pro Stunde geschätzt wurde. |
| -> Reference (Stage 1):  Security and Privacy Threat Analysis for Solid | Omid Mirzamohammadi, Kristof Jannes, Laurens Sion, Dimitri Van Landuyt, Aysajan Abidin, Dave Singelée  08 November 2023, IEEE Cybersecurity Development | KEIN ZUGRIFF AUF VOLLSTÄNDIGEN PAPER | Die Arbeit untersucht die Sicherheits- und Datenschutzrisiken des Solid-Protokolls. Mithilfe von Bedrohungsmodellierungs-Tools wie SPARTA, STRIDE und LINDDUN wird ein realistischer Finanzanwendungsfall analysiert. | Mehrere kritische, besonders datenschutzbezogene Schwachstellen, die bisher zu wenig beachtet wurden. Die Erkenntnisse sollen helfen, zukünftige Sicherheitsmaßnahmen gezielt zu priorisieren. |
| -> Reference (Stage 2):  Two Architectural Threat Analysis Techniques Compared | Katja Tuma, R. Scandariato  Published in European Conference on… 24 September 2018 | Beiden Ansätze hinsichtlich ihrer Effektivität bei der Identifikation von Sicherheitsbedrohungen (Nutzen) sowie dem Zeitaufwand für die Durchführung (Kosten) zu vergleichen. | zwei Varianten der STRIDE-Methode vergleicht: Variante 1 (isoliert): Betrachtet jede Systemkomponente einzeln. Variante 2 (interaktiv): Betrachtet Paarungen von Komponenten und deren Interaktionen.  (Paper nicht vollständig verfügaber!) | die interaktive Variante der STRIDE-Methode deutlich zeitaufwändiger |
| -> Reference (Stage 2):  Automating the early detection of security design flaws | Katja Tuma, Laurens Sion, R. Scandariato, Koen Yskout  16. Oktober 2020, ACM/IEEE International Conference on Model Driven Engineering Languages and Systems | Sicherheitsanalysen im Designstadium von Software durch teilweise Automatisierung effizienter und zuverlässiger zu gestaltbar.  (Paper nicht vollständig verfügbar!) | Durch die Nutzung von Modellabfragen (model query patterns) können bestimmte Sicherheitsmängel automatisch erkannt werden – z. B. unsichere Datenfreigabe –, was insbesondere bei großen Modellen die manuelle Inspektion sinnvoll ergänzt. | Vollständige Automatisierung ist derzeit nicht möglich, aber der automatisierte Ansatz kann sinnvolle Unterstützung leisten, indem er Sicherheitsexperten gezielt auf problematische Stellen im Design hinweist. |
| -> Reference (Stage 3):  A descriptive study of Microsoft’s threat modeling technique | Riccardo Scandariato, Kim Wuyts, Wouter Joosen  1 Juni 2015, Requirements Engineering | Wie effektiv und aufwendig ist die Bedrohungsmodellierung mit Microsoft STRIDE?  Wie viele valide Bedrohungen pro Stunde werden identifiziert? (Produktivität)  Wie viele falsch-positive Bedrohungen (Fehlalarme) entstehen? (Präzision)  Wie viele Bedrohungen werden übersehen (false negatives)? (Recall) | Eine deskriptives (beschreibendes) Erststudium zur Effektivität von STRIDE die über drei Jahre hinweg läuft.  57 Masterstudierende der Informatik in einer kontrollierte Laborumgebung mussten eine Bedrohungsanalyse eines mittelgroßen verteilten Systems durchführen.  Teilnehmer mussten Bedrohungen anhand von DFD-Elementen und STRIDE-Kategorien identifizieren (gemäß einer Checkliste mit Bedrohungsbäumen) und  diese als Misuse Cases dokumentieren,  Annahmen zu Sicherheitsmechanismen explizit machen (z. B. keine Authentifizierung, keine Verschlüsselung, keine Logs),  sowie “Capture Points” für Sicherheitsanforderungen angeben.  Die Teams arbeiteten eigenständig, dokumentierten ihren Aufwand (in Stunden), und bewerteten die Schwierigkeit einzelner Schritte per Fragebogen. Die Ergebnisse wurden von zwei Sicherheitsexperten bewertet.  [Die Autoren betonen methodische Herausforderungen in der Sicherheitsbewertung und regen weitere Forschung an, um Ursachen von Fehlern zu verstehen und das Threat Modeling zu verbessern.] | STRIDE ist nicht schwer zu erlernen.  STRIDE lässt sich in einer kontrollierten Umgebung prinzipiell anwenden.  Die Technik ist zeitaufwendig.  Viele Bedrohungen werden übersehen, was Fragen zur Vollständigkeit und Verlässlichkeit der Ergebnisse aufwirft.  Die Bedrohungsermittlung mittels DFD-Mapping und Bedrohungsbäumen erlaubt es, strukturierte Misuse Cases zu formulieren und relevante Sicherheitsanforderungen ("Capture Points") zu identifizieren.  Recall unter 80%, relevante Bedrohungen übersehe und Präzision ist etwa 81% hoch.  Durchschnittlich 1.2 bis 1.8 korrekt identifizierte Bedrohungen pro Stunde.  Mehr investierte Zeit führte nicht zwangsläufig zu besseren Ergebnissen.  Teilnehmer hatten oft kein realistisches Bild davon, wie viele Bedrohungen sie übersehen hatten (FN), was zu einer Überbewertung der eigenen Analyse führen kann, aber es gab mehr korrekte (True Positives, TP) als FP..  T-I-D-Kategorien (Tampering, Information Disclosure, Denial of Service) wurden deutlich stärker abgedeckt als S-R-E (Spoofing, Repudiation, Elevation of Privilege), was die Ausgewogenheit der Analyse beeinflusst. |
| PILLAR: AN AI-POWERED PRIVACY THREAT MODELING TOOL | Majid Mollaeefar a, Andrea Bissoli a, and Silvio Ranise.  11. Oktober 2024  Department of Mathematics, University of Trento, Trento, Italy | ob und wie das Tool PILLAR, das LLMs nutzt, die automatisierte Erkennung und Priorisierung von Datenschutzbedrohungen in Softwarearchitekturen ermöglichen kann – im Vergleich zu bisherigen manuellen Methoden. | Um diese Hypothese zu testen, wurde das Tool **PILLAR** evaluiert anhand folgender Funktionen und Methoden:  **Analyse von Datenflussdiagrammen (DFDs):**  Automatische Erkennung von Datenflüssen, Akteuren, Prozessen und Sicherheitsmaßnahmen.  **Automatisierte Bedrohungserkennung und -klassifikation:**  Anwendung etablierter Modelle wie **STRIDE**.  **Risikobewertung und Priorisierung:**  Bewertung jeder Bedrohung mit Risikoklassifizierung.  **Integration bestehender Sicherheitsstandards und Bibliotheken:**  Unterstützung von GDPR, NIST, ISO 27001.  **Maschinelles Lernen:**  Kontinuierliches Lernen aus neuen Sicherheitsvorfällen.  **Generierung konkreter Handlungsempfehlungen zur Verbesserung des Datenschutzes.** | • Falsch-positive & falsch-negative Ergebnisse: Relevante Bedrohungen werden übersehen oder harmloses Verhalten als Gefahr eingestuft.  • Begrenztes Verständnis komplexer Systeme: LLMs haben Schwierigkeiten, tiefere Kontextabhängigkeiten in komplexen Architekturen zu erkennen. Besonders bei verteilten Systemen oder Microservices ist die Bedrohungsanalyse unzuverlässig.  • Mangelnde Transparenz: Entscheidungen der KI sind oft nicht nachvollziehbar.  • Datenschutzrisiken: Gefahr von Datenlecks durch unsichere Modellnutzung.  • Nicht immer konforme Maßnahmen: Sicherheitsvorschläge sind teils unpassend für branchenspezifische Standards. Manchmal schlägt PILLAR allgemeine Lösungen vor, die nicht in jedem Kontext anwendbar sind |
| -> zu diesem Paper konnte nicht genügend andere gefunden werden [connectedpapers.com] | - | - | - | - |
| THREATMODELING-LLM: AUTOMATING THREAT MODELING USING LARGE LANGUAGE MODELS FOR BANKING SYSTEM | Shuiqiao Yang, Tingmin Wu, Shigang Liu, David Nguyen, Seung Jang, A. Abuadbba  26. November 2024, arXiv.org | Wie effektiv sind LLMs zur automatisierten Erstellung, Analyse und Pflege von Bedrohungsmodellen auf Basis mit dem Ziel, den manuellen Aufwand deutlich zu reduzieren und die Effizienz der Sicherheitsanalyse zu erhöhen. | **Automatisierte Erstellung von DFDs:** LLMs analysieren Systemdokumentationen und Quellcode, um automatisch Datenflussdiagramme zu generieren. Ziel: Wegfall der manuellen Erstellung durch Sicherheitsanalysten.  **Analyse von DFDs zur Bedrohungserkennung:**  LLMs erkennen sicherheitsrelevante Schnittstellen zwischen Systemkomponenten.  Identifikation potenzieller Angriffsvektoren.  Klassifikation von Risiken (z. B. Hoch-, Mittel-, Niedrigrisiko).  **Ableitung von Bedrohungsszenarien:** LLMs generieren automatisch Bedrohungsszenarien auf Basis der DFDs zur Unterstützung menschlicher Sicherheitsanalysten.  **Laufende Optimierung:** Durch kontinuierliches Training mit neuen Sicherheitsdaten verbessert sich das System über Zeit. | Automatisierung spart Zeit und Ressourcen.  Große Datenmengen können schnell verarbeitet werden.  LLMs erkennen viele gängige Bedrohungen zuverlässig.  Menschliche Experten werden entlastet, nicht ersetzt.  **Unvollständige oder falsche DFDs durch** ungenaue Modellgenerierung.  Domänenspezifische Bedrohungen (z. B. speziell für Banken) werden nicht immer erkannt.  **Overfitting,** Modelle sind auf bekannte Angriffsmuster trainiert – neue Strategien werden schlecht erkannt.  LLMs können den Menschen nicht vollständig ersetzen. |
| -> Reference (Stage 2):  A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications | Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, S. Mondal, Aman Chadha  5. Februar 2024, arXiv.org | Eine sorgfältig gestaltete Eingabeaufforderungen (Prompts) könnte den Modellen helfen, Aufgaben ohne umfangreiche Nachschulungen oder Anpassungen der Modellparameter erfolgreich zu bewältigen. | Erforscht wird, wie verschiedene Techniken des Prompt Engineerings – von Zero-Shot und Few-Shot Prompts bis zu fortgeschrittenen Methoden wie Chain-of-Thought (CoT) und LogiCoT – die Leistung von LLMs und VLMs in verschiedenen Anwendungsbereichen verbessern können. Die Studie untersucht, welche spezifischen Ansätze am besten für komplexe Aufgaben wie mathematische Problemlösungen, logisches Denken und Fragebeantwortung geeignet sind | CoT und Auto-CoT erzielten signifikante Verbesserungen in der Genauigkeit, vor allem bei komplexen Aufgaben wie Mathematik und Logik.  Self-Refine zeigte große Fortschritte bei der Verbesserung der Modellantworten in Bezug auf Kontextualisierung und Genauigkeit.  Die Graph-of-Thought (GoT)-Methode steigerte die Genauigkeit bei Aufgaben, die komplexe und nicht-lineare Denkvorgänge erforderten.  Einige Techniken, wie Chain-of-Symbol und System2Attention, hatten Probleme bei der Skalierbarkeit und der Generalisierbarkeit in breitere Anwendungsbereiche.  Die Few-Shot Prompting-Technik konnte aufgrund des Token-Bedarfs bei langen Eingabedaten kostspielig und ineffizient werden.  Herausforderungen bei Bias, Präzision und Interpretierbarkeit. Ethik bleibt ein wichtiger Aspekt bei der Entwicklung von LLMs. |
| -> Reference (Stage 2):  LLM-Adapters: An Adapter Family for Parameter-Efficient Fine-Tuning of Large Language Models | Zhiqiang Hu, Yihuai Lan, Lei Wang, Wanyu Xu, Ee-Peng Lim, R. Lee, Lidong Bing, Soujanya Poria  2023, Conference on Empirical Methods in Natural Language Processing | **adapterbasierte parameter-effiziente Fine-Tuning-Methoden (PEFT)** es ermöglichen, **kleinere Open-Source-Sprachmodelle (z. B. LLaMA, BLOOM, GPT-J)** so leistungsstark oder sogar leistungsstärker zu machen als große Modelle (z. B. GPT-3.5 oder GPT-4), **bei geringem Rechenaufwand** und mit deutlich **weniger zu trainierenden Parametern**.  Welche Adapter-Typen, Platzierungen und Konfigurationen die beste Leistung erzielen. | **Adaptertypen (PEFT-Methoden)**: **Series Adapter**, **Parallel Adapter**, **LoRA (Reparametrisierung)** und **Prompt-basierte Methoden**.  Es wurden **14 Datensätze** aus zwei Bereichen verwendet: **Mathematisches Denken (Arithmetic Reasoning)**  **Alltagslogik (Commonsense Reasoning)**  **Vergleich in verschiedenen Testszenarien:**  **In-Distribution (ID)**: Daten, die dem Trainingsset ähneln  **Out-of-Distribution (OOD)**: neue, abweichende Daten | **LLaMA-13B mit LoRA schlägt GPT-3.5 (175B) auf mehreren mathematischen Aufgaben** (z. B. MultiArith, AddSub, SingleEq)  **LLaMA-13B + Adapter schlägt ChatGPT bei Aufgaben des Alltagsverstands**, wenn mit passenden In-Distribution-Daten feinjustiert wurde.  **Leistungsstarke und kostengünstige Alternative zu großen Modellen**  Keine klare „eine beste Lösung“ für alle Aufgaben  Out-of-Distribution-Leistung ist schwächer |
| -> Reference (Stage 1):  The Accuracy and Appropriateness of ChatGPT Responses on Nonmelanoma Skin Cancer Information Using Zero-Shot Chain of Thought Prompting | Ross O'Hagan, D. Poplausky, Jade N Young, N. Gulati, Melissa Levoska, Benjamin Ungar, J. Ungar  2023, JMIR Dermatology | Wie gut kann ChatGPT medizinische Informationen zu nicht-melanotischem Hautkrebs (NMSC) liefert, und vergleicht dabei die Genauigkeit von Standard-Prompting und Zero-Shot Chain of Thought (ZS-COT) Prompting. | Es wird getestet, ob ZS-COT-Prompting genauere Antworten auf 25 klinische Fragen zu NMSC liefert als Standard-Prompting.  25 Fragen zu NMSC wurden in vier Kategorien erstellt: Allgemeines, Diagnose, Behandlung, Risikofaktoren.  ChatGPT beantwortete die Fragen mit Standard-Prompting und ZS-COT-Prompting.  Drei Dermatologen bewerteten die Genauigkeit und Eignung der Antworten für Websites und EHR-Nachrichten. | Die Genauigkeit beider Methoden war hoch (Durchschnitt 4,89 von 5).  Es gab keinen signifikanten Unterschied zwischen den beiden Methoden.  Beide Methoden waren für Websites geeignet, aber ZS-COT war nicht signifikant besser als Standard-Prompting. |

[1]

In der Studie wurden verschiedene Prompting-Methoden in unterschiedlichen Szenarien getestet.

- **Zero-Shot Prompting:** Es wurden Aufgaben ohne Trainingsbeispiele durchgeführt, wobei das Modell auf sein Vorwissen zurückgriff.

- **Few-Shot Prompting:** Hier wurden dem Modell nur wenige Beispiele gegeben, um es auf neue Aufgaben vorzubereiten und die Leistung zu testen.

- **Chain-of-Thought (CoT) Prompting:** Es wurde überprüft, wie LLMs durch Schritt-für-Schritt-Denken zu besseren und detaillierteren Antworten geführt werden.

- **LogiCoT (Logical Chain-of-Thought):** Diese Methode wurde verwendet, um logische Fehler in den Schlussfolgerungen des Modells zu reduzieren, indem ein Prozess der Überprüfung und Korrektur eingeführt wurde.

- **Auto-CoT:** Ein automatisierter Ansatz, um verschiedene Ketten von Überlegungen zu generieren, wurde getestet, um Fehler zu minimieren und die Leistung zu steigern.

- **Self-Consistency und Self-Refine:** Diese Techniken testeten, wie Modelle durch selbstkonsistente und selbstverfeinernde Mechanismen ihre Antworten verbessern konnten.

- **Graph-of-Thoughts (GoT) und Tree-of-Thoughts (ToT):** Komplexe Problemlösungen wurden durch dynamische und strukturierte Überlegungsprozesse untersucht, die eine systematische Exploration der Lösungen ermöglichen.

- **ECHO (Self-Harmonized Chain-of-Thought Prompting):** Hierbei wurde untersucht, wie ein selbstharmonisierter Ansatz die Leistung von automatisierten Ketten von Gedanken (Auto-CoT) verbessern kann, um mit diversifizierten und fehlerhaften Rationalen besser umzugehen.

- **Logic-of-Thought Prompting:** Diese Methode verbesserte die Genauigkeit von CoT-Prompts und zeigte Verbesserungen von 4.35% auf dem ReClor-Benchmark und 5% auf LogiQA.

- **Code Prompting:** Es zeigte sich, dass die Verwendung von Code-Prompts die Leistung von LLMs signifikant steigerte. Beispielsweise erzielte GPT-3.5 im Durchschnitt eine Verbesserung von 8.42 F1-Punkten, während Mistral eine Verbesserung von 4.22 Punkten zeigte.

 **Active-Prompting**: Auswahl unsicherer Fragen zur Verbesserung der LLM-Leistung.

 **Automatic Prompt Engineer (APE)**: Automatische Generierung und Auswahl von Prompts zur Anpassung an Aufgaben.

 **Emotion Prompting**: Hinzufügen emotionaler Anreize zur Verbesserung der emotionalen Intelligenz von LLMs.

 **Scratchpad und PoT**: Verbesserung der Lösung komplexer mathematischer und programmiertechnischer Aufgaben.

 **OPRO**: Optimierung von Aufgaben durch natürliche Sprachprompts.

 **RaR**: Verbesserung des Verständnisses und der Antwortgenauigkeit durch Umformulierung von Fragen.

1. ("undetected" blieben). Der **Recall-Wert** zeigt also, wie gut ein System dabei ist, alle relevanten Bedrohungskategorien zu erfassen. [↑](#footnote-ref-1)