LLM AI Security Framework

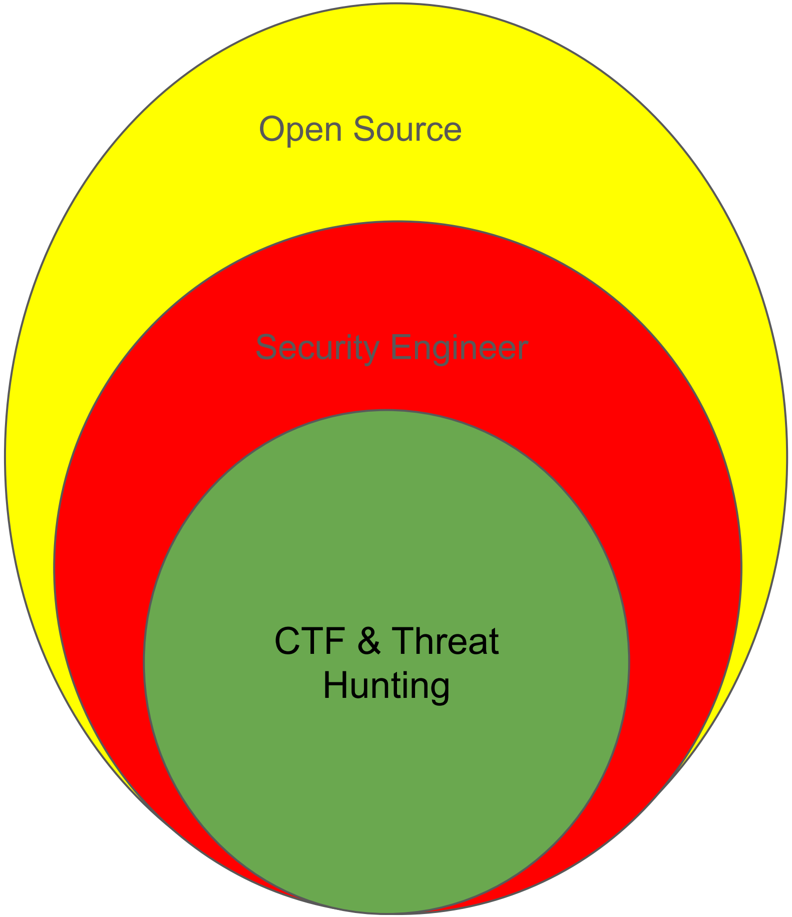
"Secure prompt? Seriously, are you sure?"

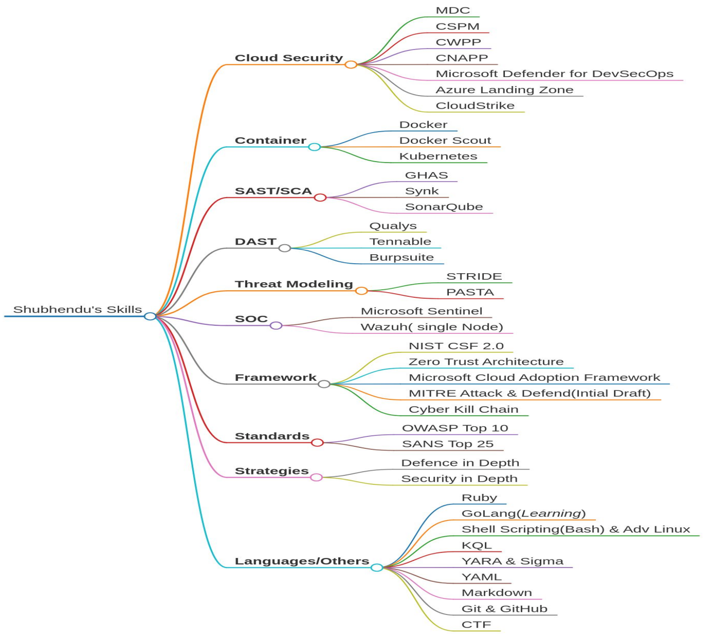
$whoami

Shubhendu Shubham

“sudo rm -rf / problems”

aka “Troubleshooter”





- Shubhendu's Skills
  - Cloud Security
    - MDC
    - CSPM
    - CWPP
    - CNAPP
    - Microsoft Defender for DevSecOps
    - Azure Landing Zone
    - CloudStrike
  - Container
    - Docker
    - Docker Scout
    - Kubernetes
  - SAST/SCA
    - GHAS
    - Synk
    - SonarQube
  - DAST
    - Qualys
    - Tennable
    - Burpsuite
  - Threat Modeling
    - STRIDE
    - PASTA
  - SOC
    - Microsoft Sentinel
    - Wazuh(single Node)
  - Framework
    - NIST CSF 2.0
    - Zero Trust Architecture
    - Microsoft Cloud Adoption Framework
    - MITRE Attack & Defend(Intial Draft)
    - Cyber Kill Chain
  - Standards
    - OWASP Top 10
    - SANS Top 25
  - Strategies
    - Defence in Depth
    - Security in Depth
  - Languages/Others
    - Ruby
    - GoLang(*Learning*)
    - Shell Scripting(Bash) & Adv Linux
    - KQL
    - YARA & Sigma
    - YAML
    - Markdown
    - Git & GitHub
    - CTF

CTF BADGES

AZ 700

AZ 500

CERTIFICATIONS

SC 100

AZ 305

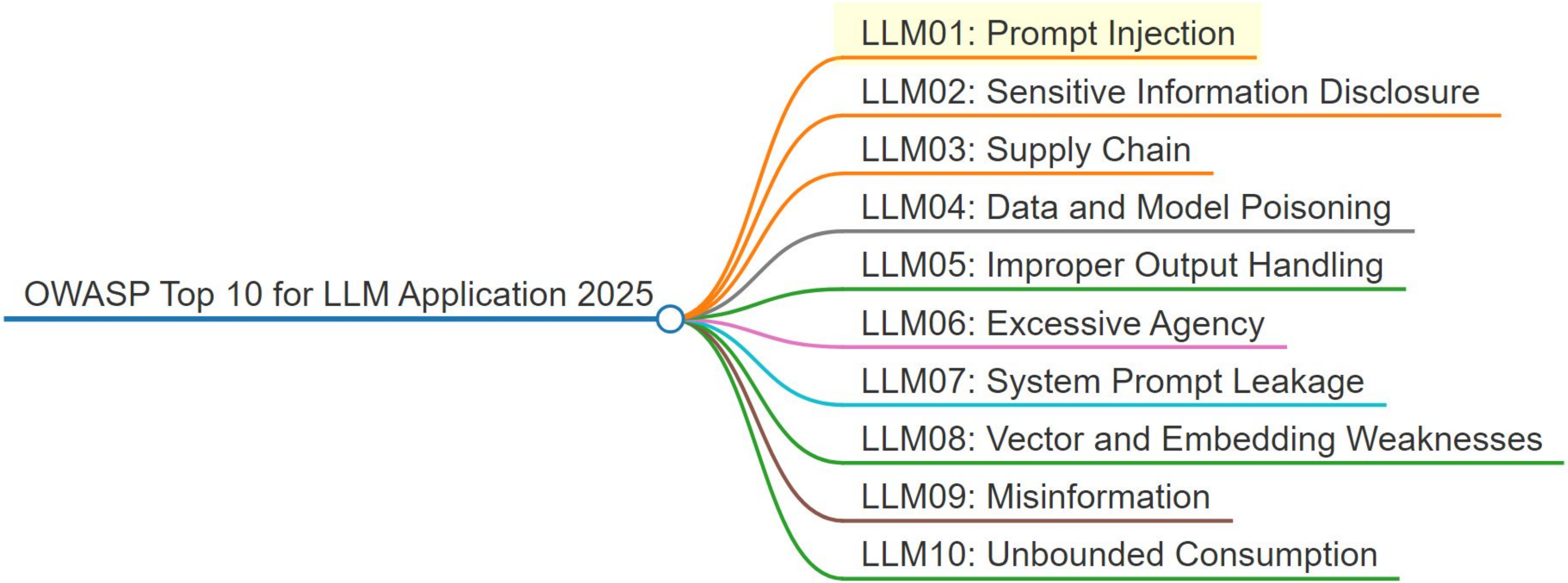
AZ 104

Community

Disclaimer

> “You can’t protect
>
> what you don’t know you have.”
>
> — not sure

OWASP Top 10 for LLM Applications 2025



OWASP Top 10 for LLM Application 2025

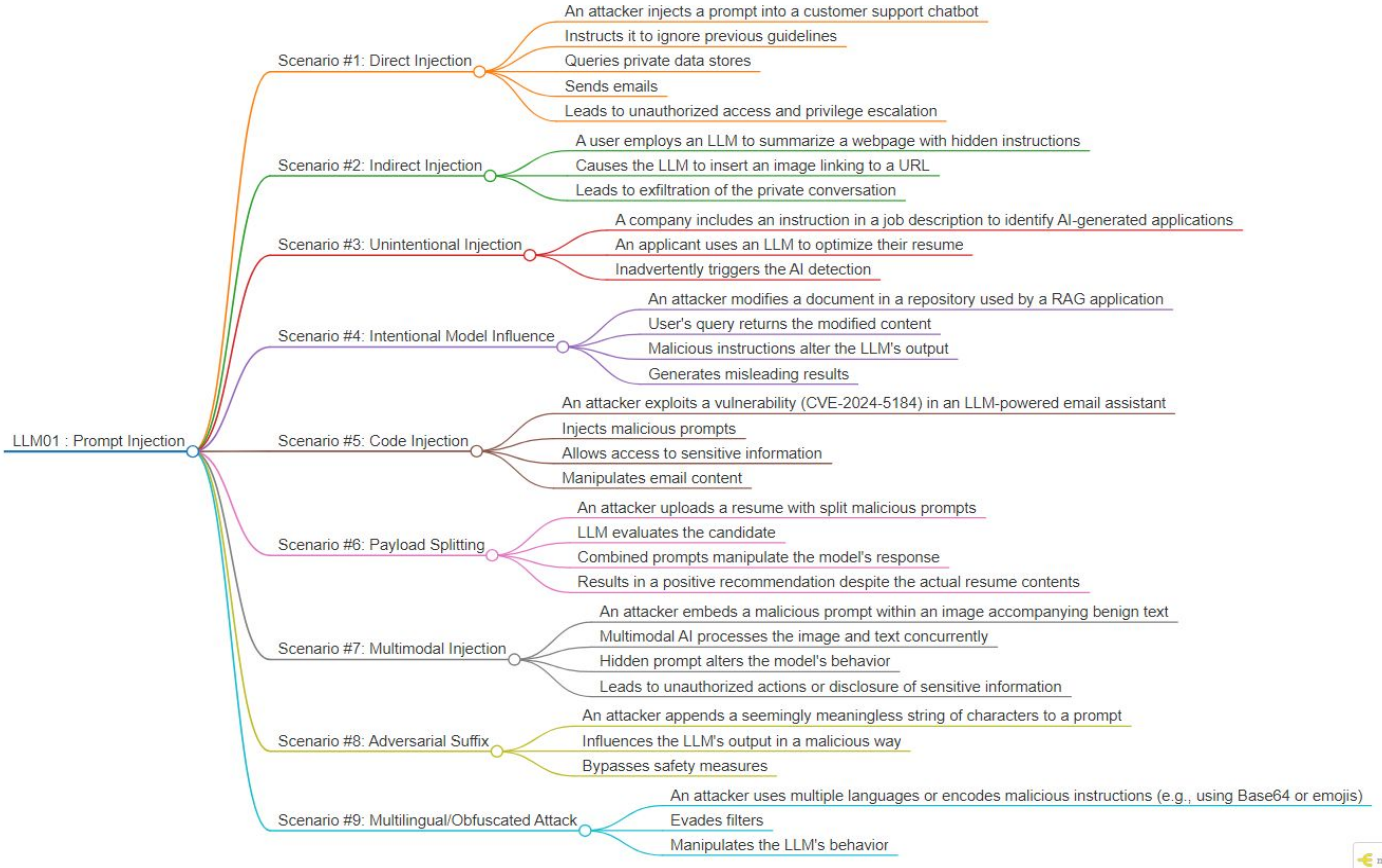
- LLM01: Prompt Injection
- LLM02: Sensitive Information Disclosure
- LLM03: Supply Chain
- LLM04: Data and Model Poisoning
- LLM05: Improper Output Handling
- LLM06: Excessive Agency
- LLM07: System Prompt Leakage
- LLM08: Vector and Embedding Weaknesses
- LLM09: Misinformation
- LLM10: Unbounded Consumption

LLM01: Prompt Injection

Prompt Injection vulnerabilities exist in how models process prompts, and how input may force the model to incorrectly pass prompt data to other parts of the model, potentially causing them to violate guidelines, generate harmful content, enable unauthorized access, or influence critical decisions. While techniques like Retrieval Augmented Generation (RAG) and fine-tuning aim to make LLM outputs more relevant and accurate, research shows that they do not fully mitigate prompt injection vulnerabilities.

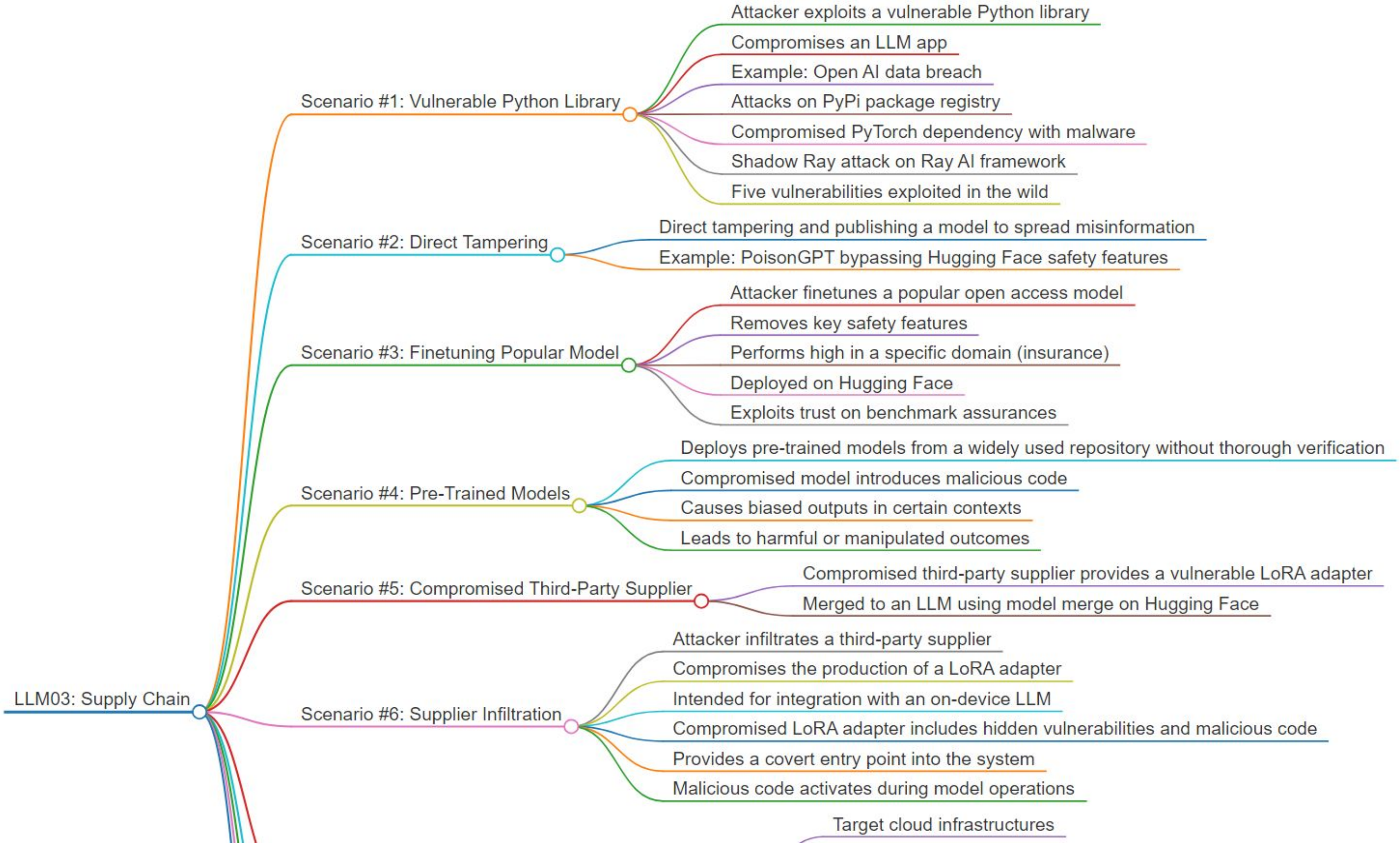
Attack Scenarios

LLM01



- LLM01 : Prompt Injection
  - Scenario #1: Direct Injection
    - An attacker injects a prompt into a customer support chatbot
    - Instructs it to ignore previous guidelines
    - Queries private data stores
    - Sends emails
    - Leads to unauthorized access and privilege escalation
  - Scenario #2: Indirect Injection
    - A user employs an LLM to summarize a webpage with hidden instructions
    - Causes the LLM to insert an image linking to a URL
    - Leads to exfiltration of the private conversation
  - Scenario #3: Unintentional Injection
    - A company includes an instruction in a job description to identify AI-generated applications
    - An applicant uses an LLM to optimize their resume
    - Inadvertently triggers the AI detection
  - Scenario #4: Intentional Model Influence
    - An attacker modifies a document in a repository used by a RAG application
    - User's query returns the modified content
    - Malicious instructions alter the LLM's output
    - Generates misleading results
  - Scenario #5: Code Injection
    - An attacker exploits a vulnerability (CVE-2024-5184) in an LLM-powered email assistant
    - Injects malicious prompts
    - Allows access to sensitive information
    - Manipulates email content
  - Scenario #6: Payload Splitting
    - An attacker uploads a resume with split malicious prompts
    - LLM evaluates the candidate
    - Combined prompts manipulate the model's response
    - Results in a positive recommendation despite the actual resume contents
  - Scenario #7: Multimodal Injection
    - An attacker embeds a malicious prompt within an image accompanying benign text
    - Multimodal AI processes the image and text concurrently
    - Hidden prompt alters the model's behavior
    - Leads to unauthorized actions or disclosure of sensitive information
  - Scenario #8: Adversarial Suffix
    - An attacker appends a seemingly meaningless string of characters to a prompt
    - Influences the LLM's output in a malicious way
    - Bypasses safety measures
  - Scenario #9: Multilingual/Obfuscated Attack
    - An attacker uses multiple languages or encodes malicious instructions (e.g., using Base64 or emojis)
    - Evades filters
    - Manipulates the LLM's behavior

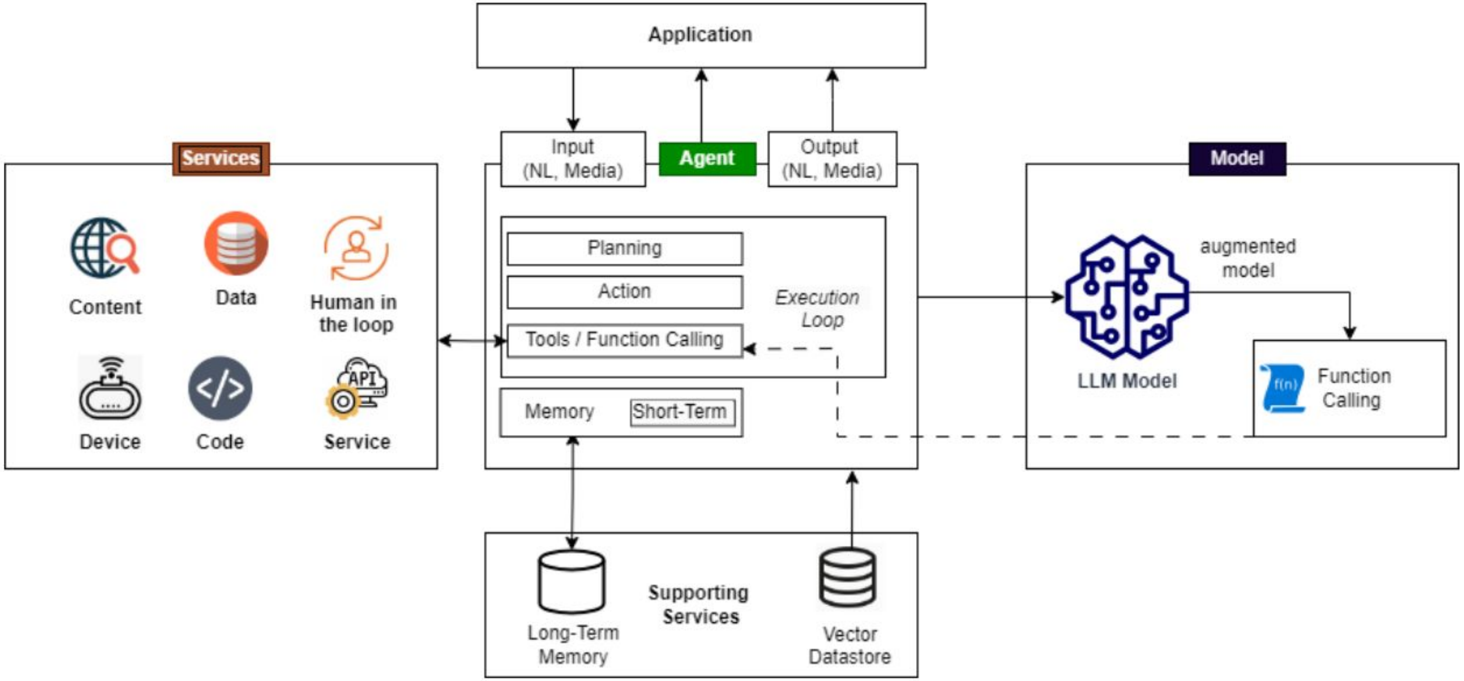
LLM03: Supply Chain



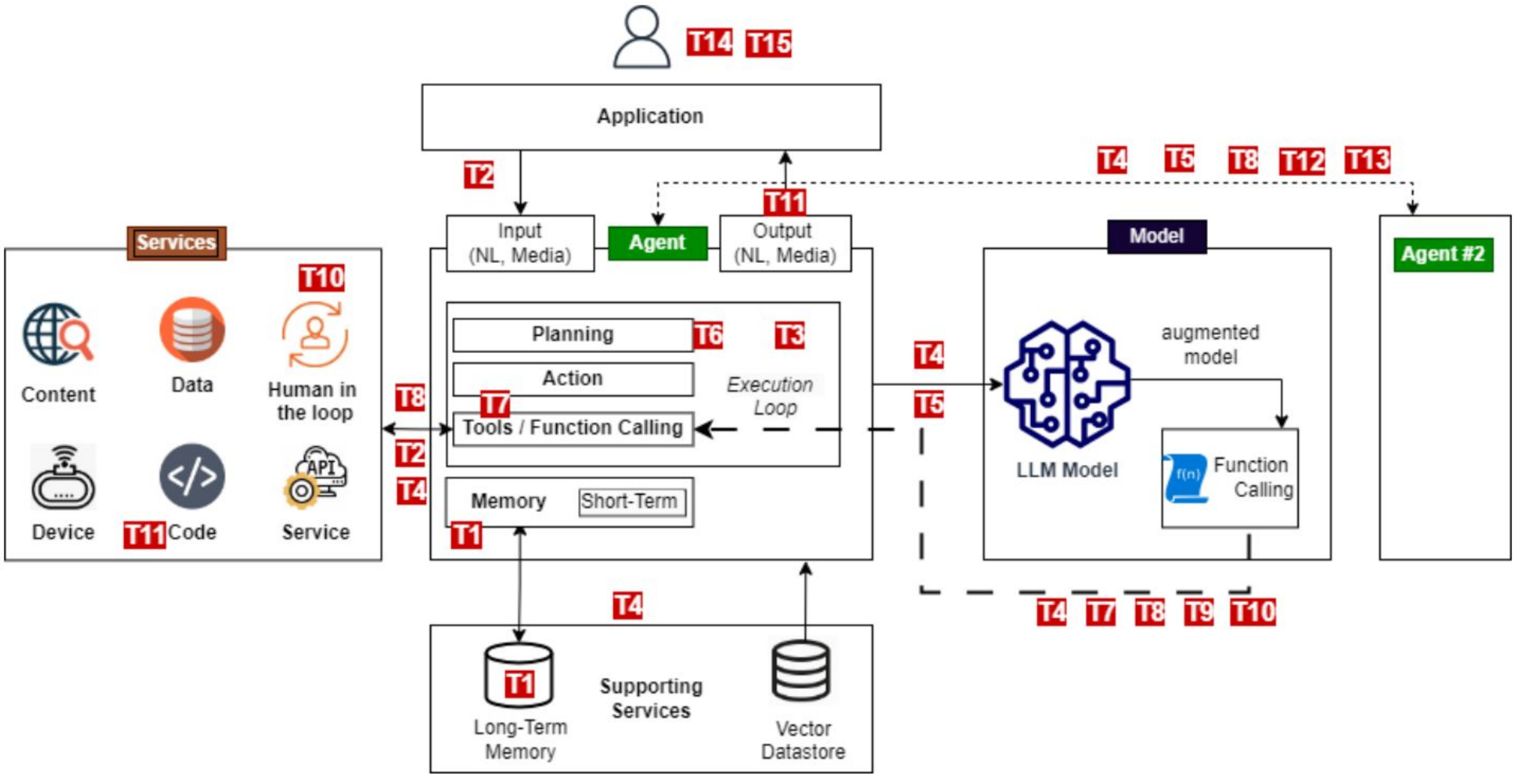
- LLM03: Supply Chain
  - Scenario #1: Vulnerable Python Library
    - Attacker exploits a vulnerable Python library
    - Compromises an LLM app
    - Example: Open AI data breach
    - Attacks on PyPi package registry
    - Compromised PyTorch dependency with malware
    - Shadow Ray attack on Ray AI framework
    - Five vulnerabilities exploited in the wild
  - Scenario #2: Direct Tampering
    - Direct tampering and publishing a model to spread misinformation
    - Example: PoisonGPT bypassing Hugging Face safety features
  - Scenario #3: Finetuning Popular Model
    - Attacker finetunes a popular open access model
    - Removes key safety features
    - Performs high in a specific domain (insurance)
    - Deployed on Hugging Face
    - Exploits trust on benchmark assurances
  - Scenario #4: Pre-Trained Models
    - Deploys pre-trained models from a widely used repository without thorough verification
    - Compromised model introduces malicious code
    - Causes biased outputs in certain contexts
    - Leads to harmful or manipulated outcomes
  - Scenario #5: Compromised Third-Party Supplier
    - Compromised third-party supplier provides a vulnerable LoRA adapter
    - Merged to an LLM using model merge on Hugging Face
  - Scenario #6: Supplier Infiltration
    - Attacker infiltrates a third-party supplier
    - Compromises the production of a LoRA adapter
    - Intended for integration with an on-device LLM
    - Compromised LoRA adapter includes hidden vulnerabilities and malicious code
    - Provides a covert entry point into the system
    - Malicious code activates during model operations
  - Target cloud infrastructures

Agentic AI – Threats

Single Agent Architecture



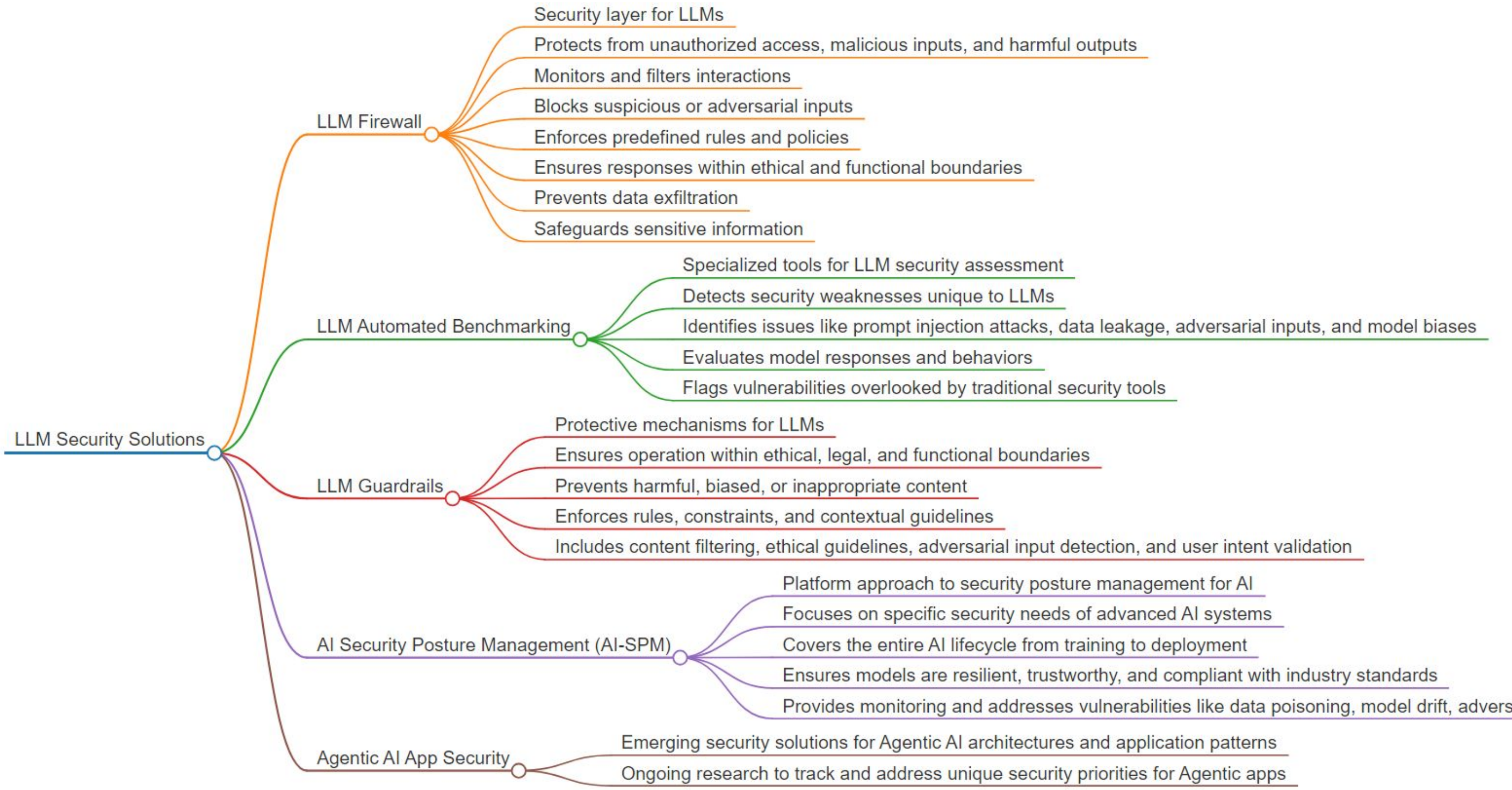
Threat Modeling



file:///C:/Users/ShubhenduShubham/Downloads/agentic-ai-threat-modeling.html

@newt_s
RREY KEHNA KYA CHAHTE HO

Solutions



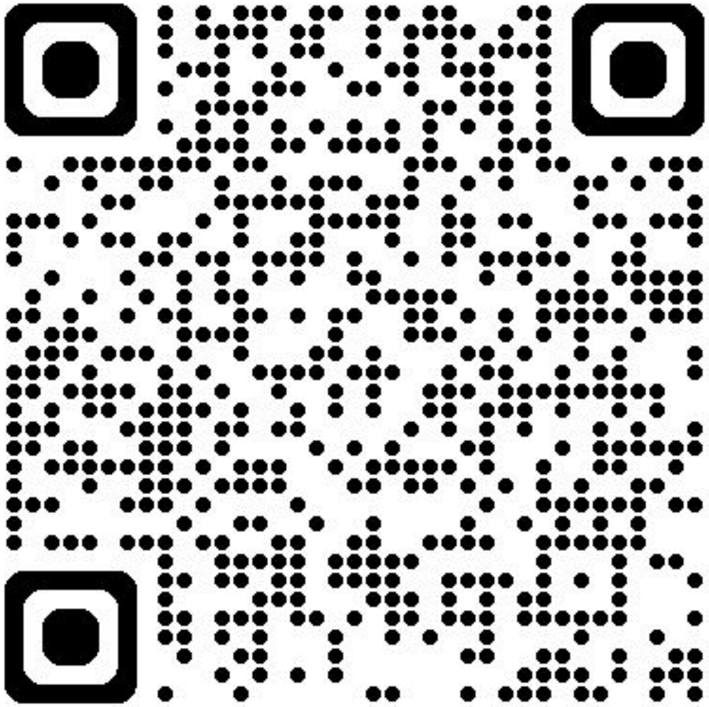
- LLM Security Solutions
  - LLM Firewall
    - Security layer for LLMs
    - Protects from unauthorized access, malicious inputs, and harmful outputs
    - Monitors and filters interactions
    - Blocks suspicious or adversarial inputs
    - Enforces predefined rules and policies
    - Ensures responses within ethical and functional boundaries
    - Prevents data exfiltration
    - Safeguards sensitive information
  - LLM Automated Benchmarking
    - Specialized tools for LLM security assessment
    - Detects security weaknesses unique to LLMs
    - Identifies issues like prompt injection attacks, data leakage, adversarial inputs, and model biases
    - Evaluates model responses and behaviors
    - Flags vulnerabilities overlooked by traditional security tools
  - LLM Guardrails
    - Protective mechanisms for LLMs
    - Ensures operation within ethical, legal, and functional boundaries
    - Prevents harmful, biased, or inappropriate content
    - Enforces rules, constraints, and contextual guidelines
    - Includes content filtering, ethical guidelines, adversarial input detection, and user intent validation
  - AI Security Posture Management (AI-SPM)
    - Platform approach to security posture management for AI
    - Focuses on specific security needs of advanced AI systems
    - Covers the entire AI lifecycle from training to deployment
    - Ensures models are resilient, trustworthy, and compliant with industry standards
    - Provides monitoring and addresses vulnerabilities like data poisoning, model drift, advers
  - Agentic AI App Security
    - Emerging security solutions for Agentic AI architectures and application patterns
    - Ongoing research to track and address unique security priorities for Agentic apps

References

1. AI Security Solution Cheat Sheet Q1-2025 - OWASP Top 10 for LLM & Generative AI Security
2. Agentic AI - Threats and Mitigations - OWASP Top 10 for LLM & Generative AI Security
3. OWASP Top 10: LLM & Generative AI Security Risks
4. LLM Applications Cybersecurity and Governance Checklist v1.1 - English - OWASP Top 10 for LLM & Generative AI Security
5. Solutions Landscape - OWASP Top 10 for LLM & Generative AI Security
6. LLMRisks Archive - OWASP Top 10 for LLM & Generative AI Security

What’s Next ?

This is not Phishing QR ,It’s my LinkedIn **Don’t Trust** Always verify



Thank you!

snowflake®

LocalStack

docker

Microsoft | Reactor