



# LLM and GenAI Security Solutions Landscape Guide 2025

---

Updated Quarterly

Version Q4 2024

Online Catalog: <https://genai.owasp.org/ai-security-solutions-landscape/>

## Revision History

Revision	Date	Authors	Description
.01	6/4/2024	Scott Clinton	Initial Draft, Charter
.05	8/10/2024	Scott Clinton, Contributors Inputs	Updated with initial feedback
.06	10/15/2024	Scott Clinton, Contributors, Reviewer Inputs	Re-factor Solutions Landscape categories,
1.0	10/15/2024	Contributors, Reviewers	Final Release Candidate

The information provided in this document does not, and is not intended to, constitute legal advice. All information is for general informational purposes only. This document contains links to other third-party websites. Such links are only for convenience, and OWASP does not recommend or endorse the contents of the third-party sites.

## License and Usage

This document is licensed under Creative Commons, CC BY-SA 4.0

You are free to:

- Share – copy and redistribute the material in any medium or format
- Adapt – remix, transform, and build upon the material for any purpose, even commercially.
- Under the following terms:
  - Attribution – You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner but not in any way that suggests the licensor endorses you or your use.
  - Attribution Guidelines - must include the project name as well as the name of the asset Referenced
    - OWASP Top 10 for LLMs - LLMSecOps Solutions Landscape
    - OWASP Top 10 for LLMs - CyberSecurity Solution and LLMSecOps Landscape Guide
- ShareAlike – If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

Link to full license text: <https://creativecommons.org/licenses/by-sa/4.0/legalcode>

The information provided in this document does not, and is not intended to, constitute legal advice. All information is for general informational purposes only. This document contains links to other third-party websites. Such links are only for convenience and OWASP does not recommend or endorse the contents of the third-party sites.

# Contents

<b>Who Is This Document For?</b>	<b>3</b>
<b>Objectives</b>	<b>3</b>
<b>Scope</b>	<b>3</b>
<b>Introduction</b>	<b>4</b>
<b>Defining the Security Solutions Landscape</b>	<b>4</b>
Landscape Considerations	4
<b>LLM Application Categories, Security Challenges</b>	<b>5</b>
Static Prompt Augmentation Applications	6
Agentic Applications	7
LLM Plug-ins, Extensions	8
Complex Applications	9
<b>LLM Development and Consumption Models</b>	<b>10</b>
<b>LLMOps and LLMSecOps Defined</b>	<b>11</b>
A Quick Ops Primer - Foundation for LLMOps	11
LLMOps Life Cycle Stages - Foundation for LLMDevSecOps	12
Scoping/Planning	13
Data Augmentation and Fine-Tuning	14
Application Development and Experimentation	14
Test and Evaluation	15
Release	15
Deploy	16
Operate	16
Monitor	17
Govern	18
Mapping to the OWASP Top 10 for LLM Threat Model	18
Application Services	19
Production Services	19
<b>OWASP Top 10 for LLMs Solutions Landscape</b>	<b>20</b>
Emerging GenAI/LLM-Specific Security Solutions	21
LLM & Generative AI Security Solutions	22
Solution Landscape Matrix Definitions	22
Landscape Solution Matrix	23
<b>Acknowledgements</b>	<b>29</b>
<b>OWASP Top 10 for LLM Project Sponsors</b>	<b>30</b>
<b>References</b>	<b>31</b>
<b>Project Supporters</b>	<b>32</b>

## Who Is This Document For?

This document is tailored for a diverse audience comprising developers, AppSec professionals, DevSecOps and MLSecOps teams, data engineers, data scientists, CISOs, and security leaders who are focused on developing strategies to secure Large Language Models (LLMs) and Generative AI applications. It provides a reference guide of the solutions available to aid in securing LLM applications, equipping them with the knowledge and tools necessary to build robust, secure AI applications.

## Objectives

This document is intended to be a companion to the OWASP Top 10 for Large Language Model (LLM) Applications List and the CISO Cybersecurity & Governance Checklist. Its primary objective is to provide a reference resource for organizations seeking to address the identified risks and enhance their security programs. While not designed to be an all-inclusive resource, this document offers a researched point of view based on the top security categories and emerging threat areas. It captures the most impactful existing and emerging categories. By categorizing, defining, and aligning applicable technology solution areas with the emerging LLM and generative AI threat landscape, this document aims to simplify research efforts and serve as a solutions reference guide.

## Scope

The scope of this document is to create a shared definition of solution category areas that address the security of the LLM and generative AI life cycle, from development to deployment and usage. This alignment supports the OWASP Top 10 List For LLMs outcomes and the CISO Cybersecurity and Governance Checklist. To achieve this, the document will create an initial framework and category descriptors, utilizing both open-source solutions and providing mechanisms for solution providers to align their offerings with specific coverage areas as examples to support each category.

The document adheres to several key rules to maintain its integrity and usefulness:

- **Vendor-Agnostic and Open Approach:** It maintains a neutral stance, avoiding recommendations of one technology over another, instead providing category guidance with choices and options.
- **Straightforward, Actionable Guidance:** The document offers clear, actionable advice that organizations can readily implement.
- **Coordinated Knowledge Graph:** It includes coordinated terms, definitions, and descriptions for key concepts.
- **Point to Existing Standards:** Where existing standards or sources of truth are available, the document references these instead of creating new sources, ensuring consistency and reliability.

# Introduction

With the growth of Generative AI adoption, usage, and application development comes new risks that affect how organizations strategize and invest. As these risks evolve, so do risk mitigation solutions, technologies, frameworks, and taxonomies. To aid security leaders in prioritization, conversations about emerging technology and solution areas must be aligned appropriately to clearly understood business outcomes for AI security solutions. The business outcomes of AI security solutions must be properly defined to aid security leaders in budgeting

Many organizations have already invested heavily in various security tools, such as vulnerability management systems, identity and access management (IAM) solutions, endpoint security, Dynamic Application Security Testing (DAST), observability platforms, and secure CI/CD (Continuous Integration/Continuous Deployment) tools, to name a few. However, these traditional security tools may not be sufficient to fully address the complexities of AI applications, leading to gaps in protection that malicious actors can exploit. For example, traditional security tools may not sufficiently address the unique data security and sensitive information disclosure protection in the context of LLM and Gen AI applications. This includes but is not limited to the challenges of securing sensitive data within prompts, outputs, and model training data, and the specific mitigation strategies such as encryption, redaction, and access control mechanisms.

Emergent solutions like LLM Firewalls, AI-specific threat detection systems, secure model deployment platforms, and AI governance frameworks attempt to address the unique security needs of AI/ML applications. However, the rapid evolution of AI/ML technology and its applications has driven an explosion of solution approaches, which has only added to the confusion faced by organizations in determining where to allocate their security budgets.

## Defining the Security Solutions Landscape

There have been many approaches to characterizing the solutions landscape for Large Language Model tools and infrastructure. In order to develop a solutions landscape that focuses on the security of LLM applications across the lifecycle from planning, development, deployment, and operation, there are four key areas of input we have focused on to develop both a definition for Large Language Model DevSecOps and related solutions landscape categories.

### Landscape Considerations

**Application Types and Scope** - which impacts the people, processes, and tools needed based on the complexity of the application and the LLM environment, as-a-service, self-hosted, or custom-built.

**Emerging LLMSecOps Process** - while this is a work in progress, many are looking to adapt and adopt existing DevOps and MLOps and associated security practices. We expect our definition to evolve as the development processes for LLM applications begin to mature.

**Threat and Risk Modeling** – understanding the risks posed by LLM systems, application usage, or misuse like those outlined in the OWASP Top 10 for LLMs and Generative AI Applications, are key to understanding which solutions are best suited to improve the security posture and combat a range of attacks.

**Tracking Emerging Solutions** – many existing security solutions are adapting to support LLM development workflows and use cases however given the nature of new threats and evolving technology and architectures new types of LLM-specific security solutions will be necessary.

## LLM Application Categories, Security Challenges

Organizations have been leveraging Machine Learning in applications for decades. This often required detailed expertise in Data Science and extensive model training. Generative AI has changed this. Specifically, Large Language Models (LLMs) have made machine learning technology widely accessible. The ability to dynamically interact in plain language has opened the door for the creation of a new class of data-driven applications and application integrations. Furthermore, usage is no longer limited to the highly skilled efforts of traditional developers and data scientists. Pre-trained models enable nearly anyone to perform complex computational tasks, regardless of prior exposure to programming or security. Organizations have been leveraging Machine Learning in applications for decades including Natural Language Processing (NLP) models that often require detailed expertise in Data Science and extensive model training.

With the advent of transformers technology enabling generative capabilities combined with the ease of access for pre-trained as-a-service models like ChatGPT and other as-a-service, Four major categories of LLM Application Architecture emerged; Prompt-centric, AI Agents, Plug-ins/extensions, and complex generative AI application where the LLM plays a key role in a larger application use case.

Prompt-centric	Agent Applications	LLM Plug-ins, Extensions	Complex Applications
<p><b>Key Attributes:</b></p> <ul style="list-style-type: none"><li>- Direct Model Interaction</li><li>- Rapid Prototyping / Experiments</li><li>- Simplicity and Accessibility</li></ul> <p><b>Use Case Examples:</b></p> <ul style="list-style-type: none"><li>- Content Generation</li><li>- Question-Answering Systems</li><li>- Language Translation Tools</li></ul> <p><b>Top Security Challenges</b></p> <ul style="list-style-type: none"><li>- Prompt injection attacks</li><li>- Data leakage from poorly crafted prompts</li></ul>	<p><b>Key Attributes:</b></p> <ul style="list-style-type: none"><li>- Autonomy and Decision-Making</li><li>- Interaction w/ External Systems</li><li>- Complex Workflow Automation</li></ul> <p><b>Use Case Examples:</b></p> <ul style="list-style-type: none"><li>- Customer Support Bots</li><li>- Data Analysis and Reporting</li><li>- Process Automation</li></ul> <p><b>Top Security Challenges</b></p> <ul style="list-style-type: none"><li>- Unauthorized access</li><li>- Confidentiality</li><li>- Increased exploitation risks</li></ul>	<p><b>Key Attributes:</b></p> <ul style="list-style-type: none"><li>- Task Specific Focus</li><li>- Bridge between the LLM and App</li><li>- Provide enhancements to LLM functionality</li></ul> <p><b>Use Case Examples:</b></p> <ul style="list-style-type: none"><li>- Content Generation Tools</li><li>- Text Summarization</li></ul> <p><b>Top Security Challenges</b></p> <ul style="list-style-type: none"><li>- Data breaches</li><li>- Introduce vulnerabilities</li><li>- Unauthorized access</li></ul>	<p><b>Key Attributes:</b></p> <ul style="list-style-type: none"><li>- Multi-Component Architecture</li><li>- Multiple Integrations</li><li>- Advanced Features, Scalability</li></ul> <p><b>Use Case Examples:</b></p> <ul style="list-style-type: none"><li>- Automated Financial Reporting</li><li>- Legal Document Analysis</li><li>- Healthcare Diagnostics</li></ul> <p><b>Top Security Challenges</b></p> <ul style="list-style-type: none"><li>- Adversarial attacks</li><li>- Misconfigurations</li><li>- Data leakage and Loss</li></ul>

(figure: Application Categories & Summary Attributes)

Having a common view of typical LLM application architectures, including agents, models, LLMs, and the ML application stack, is crucial for defining and aligning the application stack, security model, and application offerings. Below, we have provided a short description of key characteristics, use cases, and security challenges for each application category.

## Static Prompt Augmentation Applications

These applications involve specific static natural language inputs to guide the behavior of a large language model (LLM) toward generating the desired output. This technique optimizes the interaction between the user and the model by fine-tuning the phrasing, context, and instructions given to the LLM. These applications allow users to accomplish a wide range of tasks by simply refining how they ask questions or provide instructions.

### Key Characteristics

- Human to model / model to human interaction and response
- Static prompt augmentation
- Flexibility and Creativity
- Simplicity and Accessibility
- Rapid Prototyping and Experimentation

### Use Case Examples

- Experimentation/Rapid Prototyping
- Content Generation Tools
- Text Summarization Applications
- Question-Answering Systems
- Language Translation Tools
- Chatbots and Virtual Assistants

### Security Challenges

- Prompt-based applications face security risks like prompt injection attacks and data leakage from poorly crafted prompts. Lack of context or state management can lead to unintended outputs, increasing misuse vulnerability. User-generated prompts may cause inconsistent or biased responses, risking compliance or ethical violations. Ensuring prompt integrity, robust input validation, and securing the LLM environment are crucial to mitigate these risks.

## Agentic Applications

These applications leverage Large Language Models (LLMs) to autonomously or semi-autonomously perform tasks, make decisions, and interact with users or other systems. These agents are designed to act on behalf of users, handling complex processes that often involve multiple steps, integrations, and real-time decision-making. They operate with a level of autonomy, allowing them to complete tasks without constant human intervention.

### Key Characteristics

- Autonomy and Decision-Making
- Interaction with External Systems
- State Management and Memory
- Complex Workflow Automation
- Human-Agent Collaboration

### Use Case Examples

- Virtual Assistants
- Customer Support Bots
- Process Automation Agents
- Data Analysis and Reporting Agents
- Intelligent Personalization Agents
- Security and Compliance Agents

### Security Challenges

- Agent applications, with their autonomy and access to various systems, must be carefully secured to prevent misuse. They face security challenges like unauthorized access, increased exploitation risks due to interaction with multiple systems, and vulnerabilities in decision-making processes. If someone gains control of an autonomous agent, the consequences could be severe, especially in critical systems. Ensuring robust access controls and encryption methods to protect against this is essential. Ensuring data integrity and confidentiality is critical, as agents often handle sensitive information it is important to secure data at all stages, including at -rest, in motion, and access through secured APIs. Their autonomy also poses risks of unintended or harmful decisions without oversight. Robust authentication, encryption, monitoring, and fail-safe mechanisms are essential to mitigate these security risks. Observability and Traceability solutions that monitor the entire lifecycle of the Agents (Design, Development, Deployment, and Visibility on decision-making) must be considered to ensure real-time corrections using a humans-in-the-loop process can be enforced.



## LLM Plug-ins, Extensions

Plug-ins are extensions or add-ons that integrate LLMs into existing applications or platforms, enabling them to provide enhanced or new functionalities. Plug-ins typically serve as a bridge between the LLM and the application, facilitating seamless integration, such as adding a language model to a word processor for grammar correction or integrating with customer relationship management (CRM) systems for automated email responses.

While it can be sometimes difficult to draw the line between Agents and plug-ins or extensions which are often components of larger applications, one measure is the way it is deployed and used. For example, a plug-in would be a pre-built agent designed for reuse that you call explicitly, through an API, or as part of an LLMs plugin or extension framework vs. custom code running in the background on a periodic basis.

### Key Characteristics

- Modularity and Flexibility
- Seamless Integration
- Task Specific Focus
- Ease of Deployment and Use
- Rapid Updates and Maintenance

### Use Case Examples

- Content Generation Tools
- Text Summarization Applications

### Security Challenges

- Plugins interacting with sensitive data or critical systems must be carefully vetted for security vulnerabilities. Poorly designed or malicious plugins can cause data breaches or unauthorized access. LLM plugins face challenges like compatibility issues, where updates can introduce vulnerabilities, and integration with sensitive systems increases the risk of data leaks. Ensuring secure API interactions, regular updates, and robust access controls is crucial. Resource-intensive plugins may degrade performance, risking exploitation.
-

## Complex Applications

Complex applications are sophisticated software systems that deeply integrate Large Language Models (LLMs) as a central component to provide advanced functionalities and solutions. These applications are characterized by their comprehensive scope, scalability, and the integration of multiple technologies and components. They are typically designed to solve intricate problems, often in enterprise environments, and require extensive development, engineering, and ongoing maintenance efforts.

### Key Characteristics

- Multi-component architectures are designed to process prompts from other non-human systems.
- Often use multiple integrations, including other models.
- Multi-Component Architecture
- Scalability and Performance
- Advanced Features and Customization
- End-to-End Workflow Automation

### Use Case Examples

- Legal Document Analysis Platforms
- Automated Financial Reporting Systems
- Customer Service Platforms
- Healthcare Diagnostics

### Security Challenges

- Complex LLM applications face major security challenges due to their integration with multiple systems and extensive data handling. These include API vulnerabilities, data breaches, and adversarial attacks. The complexity increases the risk of misconfigurations, leading to unauthorized access or data leaks. Managing compliance across components is also difficult. Robust encryption, access controls, regular security audits, and comprehensive monitoring are essential to protect these applications from sophisticated threats and ensure data security.

## LLM Development and Consumption Models

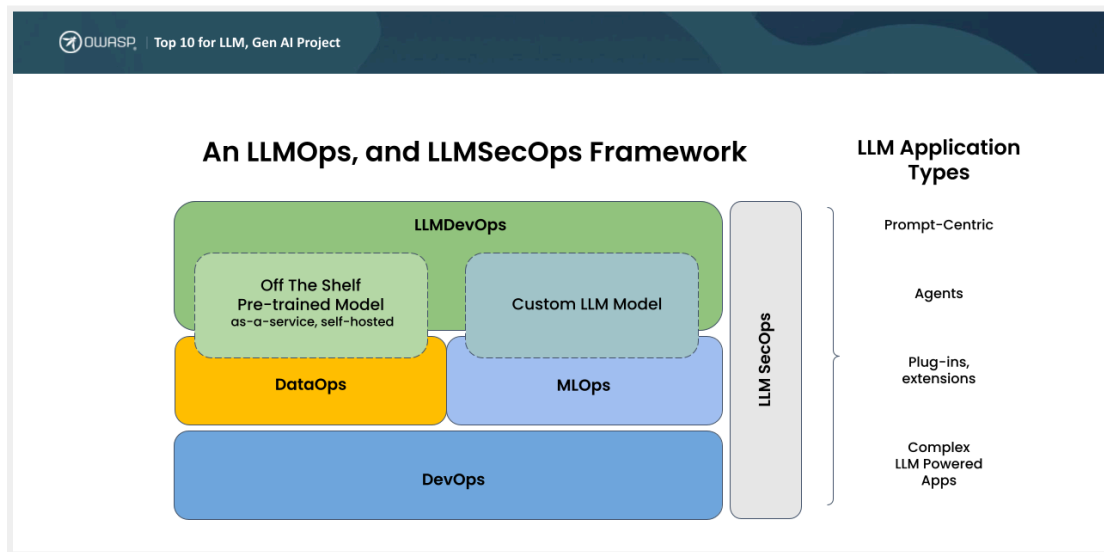
One of the first considerations for an organization is deciding upon the approach to leveraging LLM capabilities based on the type of application and goals for the project. Today, developers have a choice of two primary deployment models when implementing LLM-based applications and systems.

**Create a New Model:** The training process for custom LLMs is intensive, often involving domain-specific datasets and extensive fine-tuning to achieve desired performance levels. This approach is more akin to MLOps building ML models from the ground up, with detailed data analysis, collection formatting, cleaning, and labeling. One of the benefits of this approach is that you know the lineage and source of the data the model is built on and can attest directly to its validity and fit. However, a major downside is the resources, cost, and expertise necessary to build, train, and verify a model that meets the project objectives. Custom LLMs provide tailored solutions optimized for specific tasks and domains, offering higher accuracy and alignment with an organization's specific needs.

**Consume and Customize Existing Models:** Pre-trained (foundation) models, whether self-hosted or offered as a service, such as with ChatGPT, Bert and others on the other hand provide a more accessible entry point for organizations. These models can be quickly deployed via APIs, allowing for rapid solution validation and integration into existing systems. The LLMOps process in this scenario emphasizes customization through fine-tuning with specific datasets, ensuring the model meets the application's unique requirements, followed by robust deployment and monitoring to maintain performance and security.

## LLMOps and LLMSecOps Defined

Having a common view of typical LLM application architectures, including agents, models, LLMs, and the ML application stack, is crucial for defining and aligning the application stack and security model.



(figure: LLMOps related Operations Process for Data, Machine Learning and DevOps)

## A Quick Ops Primer – Foundation for LLMOps

DevOps, which emphasizes collaboration, automation, and continuous integration and deployment (CI/CD), has laid the groundwork for efficient software development and operations. By streamlining the software development lifecycle, DevOps enables rapid and reliable delivery of applications, fostering a culture of collaboration between development and operations teams.

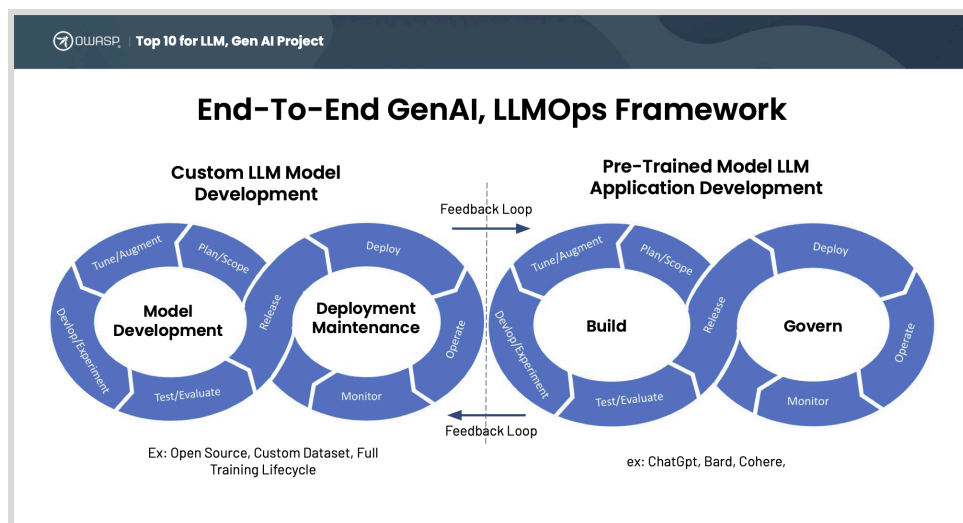
DataOps builds on DevOps, where data pipelines are managed with similar automation, version control, and continuous monitoring, ensuring data quality and compliance across the data lifecycle. MLOps also extends the DevOps principles to machine learning, focusing on the unique challenges of model development, training, deployment, and monitoring. Utilizing DevOps as a foundation ensures that both DataOps and MLOps inherit a robust infrastructure that prioritizes efficiency, scalability, security, and faster innovation in data-driven and machine learning applications.

MLOps and DataOps are foundational to LLMOps because they establish the critical processes and infrastructure needed for managing the lifecycle of large language models (LLMs). DataOps ensures that data pipelines are efficiently managed, from data collection and preparation to storage and retrieval, providing high-quality, consistent, and secure data that LLMs rely on for training and inference. MLOps extends these principles by automating and orchestrating the machine learning lifecycle, including model development, training, deployment, and monitoring.

LLMOps and MLOps, while rooted in the same foundational principles of lifecycle management, diverge significantly in their focus and requirements due to the specific demands of large language models (LLMs). LLMOps encompasses the complexities of training, deploying, and managing LLMs, which require substantial computational resources and sophisticated handling. LLMOps ensure that LLMs are efficiently integrated into production environments, monitored for performance and biases, and updated as needed to maintain their effectiveness. This holistic approach ensures that the deployment and operation of LLMs are streamlined, scalable, and secure, including considerations for data validation and provenance to ensure that the data used for training and fine-tuning LLMs is trustworthy and free from tampering. This can include techniques for data auditing and verification.

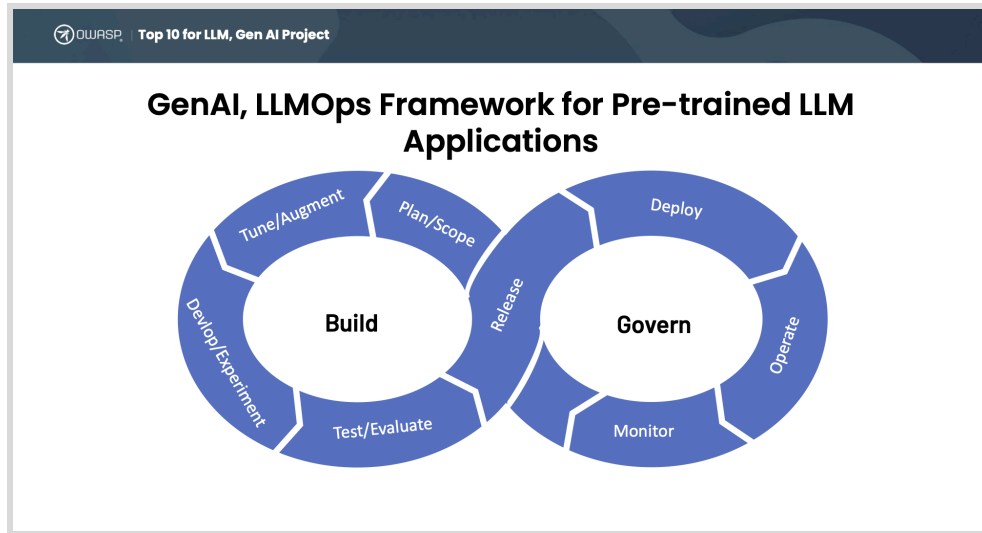
## LLMOps Life Cycle Stages – Foundation for LLMDevSecOps

As mentioned earlier in this document, to align security solutions for LLM applications for our solution guide we are using the LLMOps process to define the solution categories so that they align with the challenges developers are facing in developing and deploying LLM-based applications.



(figure: Combined LLM Custom and LLM Pre-Trained Image)

The LLMOps processes differ significantly between using pre-trained LLM models for application development and creating custom LLM models from scratch using open-source and custom datasets, which inherit more from MLOps practices with some additions. We first need to define the stages, the typical developer tasks, and the security steps at each stage of the life cycle.



(figure: LLMops Pre-Trained Process and Steps)

These phases we have defined include: Scope/Plan, Model Fine-Tuning/Data Augmentation, Test/Evaluate, Release, Deploy, Operate, Monitor, and Govern. Of course, this is an iterative approach, whether you are practicing waterfall, agile, or a hybrid approach each of these steps can be leveraged.

## Scoping/Planning

The focus is on defining the application's goals, understanding the specific needs the LLM will address, and determining how the pre-trained model will be integrated into the larger system. This stage involves gathering requirements, assessing potential ethical and compliance considerations, and setting clear objectives for performance, scalability, and user interaction. The outcome is a detailed project plan that outlines the scope, resources, and timelines needed to implement the LLM-powered application successfully.

Typical Activities:

LLMOps	LLMSecOps
<ul style="list-style-type: none"> <li>• Data Suitability</li> <li>• Model Selection</li> <li>• Requirements Gathering (business, technical, and data)</li> <li>• Task Identification</li> <li>• Task Suitability</li> </ul>	<ul style="list-style-type: none"> <li>• Access Control and Authentication Planning</li> <li>• Compliance and Regulatory Assessment</li> <li>• Data Privacy and Protection Strategy</li> <li>• Early Identification of Sensitive Data</li> <li>• Third-Party Risk Assessment (Model, Provider, etc.)</li> <li>• Threat Modeling</li> </ul>

## Data Augmentation and Fine-Tuning

The focus is on customizing the pre-trained model to better suit the specific application needs. This involves augmenting the original dataset with additional domain-specific data, enhancing the model's ability to generate accurate and contextually relevant responses. Fine-tuning is then conducted by retraining the LLM on this enriched dataset, optimizing its performance for the intended use case. This stage is critical for ensuring that the LLM adapts effectively to the unique challenges of the target domain, improving both accuracy and user experience with fewer instances of hallucination.

Typical Activities:

LLMOps	LLMSecOps
<ul style="list-style-type: none"><li>• Data Integration</li><li>• Retrieval Augmented Generation (RAG)</li><li>• Fine Tuning</li><li>• In-context Learning and Embeddings</li><li>• Reinforcement Learning with Human Feedback</li></ul>	<ul style="list-style-type: none"><li>• Data Source Validation</li><li>• Secure Data Handling</li><li>• Secure Output Handling</li><li>• Adversarial Robustness Testing</li><li>• Model Integrity Validation (ex: serialization scanning for malware)</li><li>• Vulnerability Assessment</li></ul>

## Application Development and Experimentation

The focus shifts to integrating the fine-tuned model into the application's architecture. This stage involves building the necessary interfaces, user interactions, and workflows that leverage the LLM's capabilities. Developers experiment with different configurations, testing the model's performance within the application and refining the integration based on user feedback and real-world scenarios. This iterative process is crucial for optimizing the user experience and ensuring the LLM functions effectively within the broader application context.

Typical Activities:

LLMOps	LLMSecOps
<ul style="list-style-type: none"><li>• Agent Development</li><li>• Experimentation, Iteration</li><li>• Prompt Engineering</li></ul>	<ul style="list-style-type: none"><li>• Access, Authentication, and Authorization (MFA)</li><li>• Experiment Tracking</li><li>• LLM &amp; App Vulnerability Scanning</li><li>• Model and Application Interaction Security</li><li>• SAST/DAST/ IAST</li><li>• Secure Coding Practices</li><li>• Secure Library/Code Repository</li><li>• Software Composition Analysis</li></ul>

## Test and Evaluation

At this stage in the LLM SDLC and Ops process, the focus is on rigorously assessing the application's performance, security, and reliability. This stage involves conducting comprehensive testing, including functional, security, and usability tests, to ensure the LLM integrates seamlessly with the application and meets all defined requirements. Evaluation metrics are used to measure the model's accuracy, response times, and user interactions, allowing for fine-tuning and adjustments. This phase is crucial for identifying and resolving any issues before the application is deployed to production, ensuring it operates effectively and securely in real-world environments.

Typical Activities:

LLMOps	LLMSecOps
<ul style="list-style-type: none"><li>• Evaluate the model on validation and test datasets.</li><li>• Integration Testing</li><li>• Perform bias and fairness checks.</li><li>• Stress / Performance Testing</li><li>• Use cross-validation and other techniques to ensure robustness.</li><li>• Validate the model's interpretability and explainability.</li></ul>	<ul style="list-style-type: none"><li>• Adversarial Testing</li><li>• Application Security Orchestration and Correlation</li><li>• Bias and Fairness Testing</li><li>• Final Security Audit</li><li>• Incident Simulation, Response Testing</li><li>• LLM Benchmarking</li><li>• Penetration Testing</li><li>• SAST/DAST/IAST</li><li>• Vulnerability Scanning</li></ul>

## Release

The focus shifts to deploying the finalized application to the production environment. This stage involves finalizing the deployment strategy, configuring the infrastructure for scalability and security, and ensuring that all components, including the LLM, are integrated and functioning as intended. Critical tasks include setting up monitoring and alerting systems, conducting a final security review, and preparing for user onboarding. The goal is to ensure a smooth and secure transition from development to production, making the application available to users with minimal risk and downtime.



## Typical Activities:

LLMOps	LLMSecOps
<ul style="list-style-type: none"><li>• Enable continuous delivery of model updates</li><li>• Integrate security checks and automated testing in the pipeline.</li><li>• Package the model for deployment (e.g., using Docker, Kubernetes).</li><li>• Set up CI/CD pipelines to automate application and model training, testing, and deployment.</li></ul>	<ul style="list-style-type: none"><li>• AI/ML Bill of Materials (BOM)</li><li>• Digital Model\Dataset Signing</li><li>• Model Security Posture Evaluation</li><li>• Secure CI/CD pipeline</li><li>• Secure Supply Chain Verification</li><li>• Static and Dynamic Code Analysis</li><li>• User Access Control Validation</li><li>• Model Serialization Defenses</li></ul>

## Deploy

The focus is on securely launching the LLM and its associated components into the production environment. This stage involves configuring the deployment infrastructure for scalability and reliability, ensuring that all security measures are in place, and validating the integration of the LLM with other application components. Key activities include setting up real-time monitoring, conducting final checks to prevent any vulnerabilities, and implementing fallback mechanisms to ensure continuous operation. The goal is to smoothly transition from development to live operation, ensuring that the application is ready to handle real-world usage.

## Typical Activities:

LLMOps	LLMSecOps
<ul style="list-style-type: none"><li>• Infrastructure Setup</li><li>• Integrate with existing systems or applications.</li><li>• Model and App Deployment</li><li>• Set up APIs or services for access</li><li>• User access and role management</li></ul>	<ul style="list-style-type: none"><li>• Compliance Verification</li><li>• Deployment Validation</li><li>• Digital Model\Dataset Signing Verification</li><li>• Encryption, Secrets management</li><li>• LLM Enabled Web Application Firewall</li><li>• Multi-factor Authentication</li><li>• Network Security Validation</li><li>• Secrets Management</li><li>• Secure API Access</li><li>• Secure Configuration</li><li>• User and Data Privacy Protections</li></ul>

## Operate

The focus at this stage in the LLM SDLC and Ops process is on managing and maintaining the application in a live production environment. This stage involves continuous monitoring of the application's performance, security, and user interactions to ensure it operates smoothly and securely. Key activities include responding to incidents, applying updates or patches, and refining the model based on real-world data and feedback. The goal is to maintain high availability, optimize performance, and ensure the application remains secure and effective over time.

### Typical Activities:

LLMOps	LLMSecOps
<ul style="list-style-type: none"><li>• Feedback Collection</li><li>• Iterative Enhancements</li><li>• Model Maintenance</li><li>• Performance Management</li><li>• Scalability and Infrastructure Management</li><li>• User Support and Issue Resolution</li></ul>	<ul style="list-style-type: none"><li>• Adversarial Attack Protection</li><li>• Automated Vulnerability Scanning</li><li>• Data Integrity and Encryption</li><li>• LLM Guardrails</li><li>• LLM Incident Detection and Response</li><li>• Patch Management</li><li>• Privacy, Data Leakage Protection</li><li>• Prompt Security</li><li>• Runtime Application Self-Protection</li><li>• Secure Output Handling</li></ul>

## Monitor

The focus at this stage is on continuously observing the application's performance, security, and user interactions in real-time. This stage involves tracking key metrics, detecting anomalies, and ensuring the LLM model and application components are functioning as expected. Monitoring also includes gathering data for ongoing improvement, identifying potential issues before they impact users, and maintaining compliance with security and operational standards. The goal is to ensure the application remains stable, secure, and efficient throughout its lifecycle.

### Typical Activities:

LLMOps	LLMSecOps
<ul style="list-style-type: none"><li>• Automate retraining processes based on new data.</li><li>• Detect and respond to model drift or degradation.</li><li>• Manage model versioning and rollback if necessary</li><li>• Monitor model performance (e.g., latency, accuracy, user interactions).</li></ul>	<ul style="list-style-type: none"><li>• Adversarial Input Detection</li><li>• Model Behavior Analysis</li><li>• AI/LLM Secure Posture Management</li><li>• Patch and Update Alerts</li><li>• Regulatory Compliance Tracking</li><li>• Security Alerting</li><li>• Security Metrics Collection</li><li>• User Activity Monitoring</li><li>• Observability</li><li>• Data Privacy and Protection</li><li>• Ethical Compliance</li></ul>

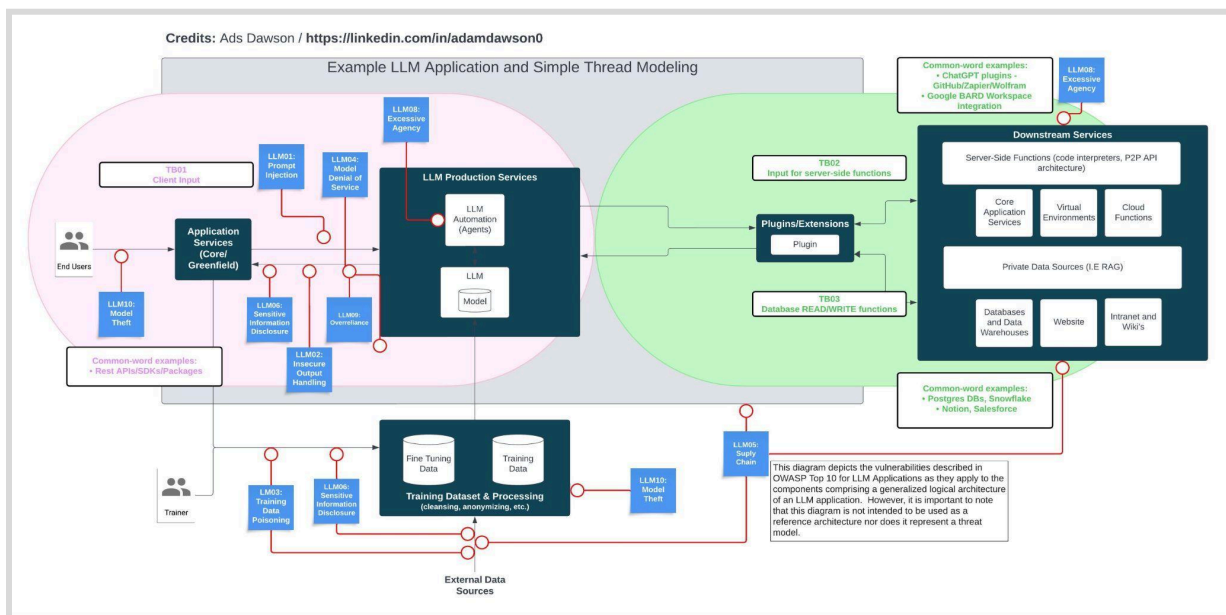
## Govern

At this stage in the LLMOps process, the focus is on establishing and enforcing policies, standards, and best practices to ensure the application operates securely and ethically throughout its lifecycle. This stage involves setting governance frameworks that oversee data usage, model management, compliance, and security controls. Key activities include auditing, risk management, and ensuring the application adheres to regulatory requirements and organizational policies.

Typical Activities:

LLMOps	LLMSecOps
<p>Conduct regular audits for compliance (e.g., GDPR, CCPA).</p> <p>Data Governance</p> <p>Document model decisions, datasets used, and model versions.</p> <p>Implement model governance frameworks.</p>	<p>Bias and Fairness Oversight</p> <p>Compliance Management</p> <p>Data Security Posture Management</p> <p>Incident Governance</p> <p>Risk Assessment and Management</p> <p>User/Machine Access audits</p>

## Mapping to the OWASP Top 10 for LLM Threat Model



(figure: OWASP Top 10 for LLM Application architecture and Threat Model)

Having a common view of typical LLM application architectures, including agents, models, LLMs, and the ML application stack, is crucial for defining and aligning the application stack and security model. By leveraging the application architecture from the OWASP Top 10 for LLMs, we can align appropriate security solutions with the specific risks and mitigation areas identified in the OWASP Top 10. This alignment ensures a comprehensive and cohesive approach to addressing the unique security challenges posed by LLM applications.

**Application Services**

An LLM application service uses large language models to process and generate human-like text for tasks like chatbots, translation, and content creation. It integrates with data agents, APIs, and security measures to ensure seamless, secure, and efficient AI-driven services, managing the model lifecycle from training to deployment.

**Production Services**

Production services deploy and manage large language models for real-time applications, ensuring high performance, scalability, and security. These services handle model training, versioning, and monitoring, integrating with APIs and security frameworks to deliver reliable apps like chatbots and translation services in a production environment.

**Training Datasets & Processing**

Training datasets consist of vast, diverse text sources, including books, articles, and web content. To ensure quality and consistency, these datasets undergo preprocessing steps like tokenization, cleaning, and normalization.

**Downstream Services**

Downstream services utilize the output of language models for applications such as chatbots, content generation, sentiment analysis, and automated translations. These services integrate LLM capabilities to enhance user interactions and data processing

**External data sources**

External data sources include web crawling through search engine APIs, remote datastores, and third-party APIs. They provide additional context and up-to-date information, enhancing the model's accuracy and relevance by supplementing the pre-trained data with real-time, domain-specific insights.

## OWASP Top 10 for LLMs Solutions Landscape

The LLM security solutions landscape leverages the LLMSecOps framework and integrates seamlessly with the LLMOps processes, encompassing Scope/Plan, Model Fine-Tuning/Data Augmentation, Test/Evaluate, Release, Deploy, Operate, Monitor, and Govern stages. This framework ensures that security is embedded at every phase of the LLM lifecycle, addressing unique challenges posed by LLM applications, including prompt-based interfaces, automation agents, LLM extensions, and complex LLM-driven applications.

The landscape includes both traditional security controls extended to support LLM Models, applications, and workloads, as well as specialized security solutions designed for LLM environments. While not intended to be a comprehensive list it provides a guiding framework for security professionals looking to integrate security controls and address the LLM Application Top 10 security risks as part of the LLM application and operations lifecycle.

## Emerging GenAI/LLM-Specific Security Solutions

The architecture and approaches for LLMs and Generative AI applications are still in their infancy, introducing new challenges that extend beyond the scope of traditional security and DevSecOps practices, often operating in unpredictable and dynamic environments where traditional security controls may fall short in addressing specific risks such as prompt injection, adversarial manipulation, and ethical biases.

We have begun to see new solutions emerging that address these security gaps and have attempted to capture them in the table below. We will continue to update our list as new solutions appear. These categories are typically early in development, but can have immediate benefits.

Security Solutions	Description
<b>LLM Firewall</b>	An LLM firewall is a security layer specifically designed to protect large language models (LLMs) from unauthorized access, malicious inputs, and potentially harmful outputs. This firewall monitors and filters interactions with the LLM, blocking suspicious or adversarial inputs that could manipulate the model's behavior. It also enforces predefined rules and policies, ensuring that the LLM only responds to legitimate requests within the defined ethical and functional boundaries. Additionally, the LLM firewall can prevent data exfiltration and safeguard sensitive information by controlling the flow of data in and out of the model.
<b>LLM Automated Benchmarking</b> (includes vulnerability scanning)	LLM-specific benchmarking tools are specialized tools designed to identify and assess security weaknesses unique to large language models (LLMs). These capabilities include detecting potential issues such as prompt injection attacks, data leakage, adversarial inputs, and model biases that malicious actors could exploit. The scanner evaluates the model's responses and behaviors in various scenarios, flagging vulnerabilities that traditional security tools might overlook.
<b>LLM Guardrails</b>	LLM guardrails are protective mechanisms designed to ensure that large language models (LLMs) operate within defined ethical, legal, and functional boundaries. These guardrails help prevent the model from generating harmful, biased, or inappropriate content by enforcing rules, constraints, and contextual guidelines during interaction. LLM guardrails can include content filtering, ethical guidelines, adversarial input detection, and user intent validation, ensuring that the LLM's outputs align with the intended use case and organizational policies.
<b>AI Security Posture Management</b>	AI-SPM has emerged as a new industry term promoted by vendors and analysts to capture the concept of a platform approach to security posture management for AI, including LLM and GenAI systems. AI-SPM focuses on the specific security needs of these advanced AI systems. Focused on the models themselves traditionally. The stated goal of this category is to cover the entire AI lifecycle—from training to deployment—helping to ensure models are resilient, trustworthy, and compliant with industry standards. AI-SPM typically provides monitoring and address vulnerabilities like data poisoning, model drift, adversarial attacks, and sensitive data leakage.

(table: List of New LLM Specific Security Solutions)

## LLM & Generative AI Security Solutions

The security solutions matrix below is based on the LLMSecOps lifecycle, and mapping it to the OWASP Top 10 for LLMs and Generative AI offers a targeted approach to assessing security controls. This matrix helps identify gaps by aligning security tools with OWASP's key risks at each stage, such as adversarial attacks and data leakage.

By cross-referencing existing security measures with the specific needs of LLM and Generative AI applications, organizations can ensure comprehensive coverage and strengthen their security posture across the entire development process.

### GEN AI SECURITY SOLUTIONS LANDSCAPE - ONLINE DIRECTORY

<https://genai.owasp.org/ai-security-solutions-landscape/>

Visit the online directory to see the latest solutions listing

The solution landscape of open source projects and proprietary offerings will be updated quarterly in this document to ensure the community maintains a reasonably updated reference list. We are also maintaining an on-line director on the project website to provide the most up to date listings. These listings are community and research sourced.

Solution listings may be submitted online by companies, projects or individuals. Submissions will be reviewed for accuracy before publishing. Below is an outline of the solution matrix maintained in the document with definitions for each area.

## Solution Landscape Matrix Definitions

EXAMPLE				
Solution (Project, Product, Service)	Type (Open Source, Proprietary)	Project, Company	Gen AI/LLMSecOps Category Coverage	Top 10 for LLM Risk Coverage
Project/Product Name Create hyperlink to the project/product	Open Source	Open Source Project Name, Company Name	List of covered security control categories provided within each stage	List of the LLM Top 10 Risks Covered by the solution. Use "LLM_All" for all categories.

## Landscape Solution Matrix

### SCOPING/PLANNING

Solution	Type	Project/Company	Gen AI/LLMSecOps	Top 10 for LLM Risk Coverage
<a href="#">StrideGPT</a>	Open Source	<a href="#">StrideGPT</a>	<ul style="list-style-type: none"> <li>Threat Modeling</li> </ul>	LLM_All
<a href="#">MitreAtlas</a>	Proprietary	<a href="#">Mitre</a>	<ul style="list-style-type: none"> <li>Threat Modeling</li> </ul>	LLM_All

### DATA AUGMENTATION AND FINE-TUNING

Solution	Type	Project/Company	Gen AI/LLMSecOps	Top 10 for LLM Risk Coverage
<a href="#">Cloaked AI</a>	Proprietary	<a href="#">IronCore Labs</a>	<ul style="list-style-type: none"> <li>Secure Data Handling</li> </ul>	LLM06
<a href="#">Unstructred.io</a>	Proprietary	<a href="#">Unstructured.io</a>	<ul style="list-style-type: none"> <li>Secure Data Handling</li> </ul>	LLM06

### DEVELOPMENT AND EXPERIMENTATION

Solution	Type	Project/Company	Gen AI/LLMSecOps	Top 10 for LLM Risk Coverage
<a href="#">Aqua Security</a>	Proprietary	<a href="#">Aqua Security</a>	<ul style="list-style-type: none"> <li>SAST, DAST &amp; IAST</li> <li>Secure Library/Code Repository</li> <li>Software Composition Analysis</li> <li>Secure Library/Code Repository</li> </ul>	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10
<a href="#">Cloaked AI</a>	Proprietary	<a href="#">IronCore Labs</a>	<ul style="list-style-type: none"> <li>Secure Data Handling</li> </ul>	LLM06
<a href="#">Fickling</a>	Open Source	<a href="#">Trail of Bits</a>	<ul style="list-style-type: none"> <li>Pickle Library</li> <li>Malicious Run-time File Detection</li> </ul>	LLM03
<a href="#">PrivacyRaven</a>	Open Source	<a href="#">Trail of Bits</a>	<ul style="list-style-type: none"> <li>Privacy testing library for AI models</li> <li>Malicious Run-time File Detection</li> </ul>	LLM03, LLM06

<a href="#">Pangea Sanitize</a>	Proprietary	<a href="#">Pangea</a>	<ul style="list-style-type: none"> <li>Model And Application Interaction Security</li> <li>Secure Coding Practices</li> </ul>	LLM02, LLM03, LLM05, LLM06
<a href="#">Pangea Authorization</a>	Proprietary	<a href="#">Pangea</a>	<ul style="list-style-type: none"> <li>Access, Authentication And Authorization (MFA)</li> <li>Model And Application Interaction Security</li> <li>Secure Coding Practices</li> </ul>	LLM04, LLM06, LLM07, LLM08, LLM10
<a href="#">Pangea Authentication</a>	Proprietary	<a href="#">Pangea</a>	<ul style="list-style-type: none"> <li>Access, Authentication And Authorization (MFA),</li> <li>Model And Application Interaction Security,</li> <li>Secure Coding Practices</li> </ul>	LLM04, LLM07, LLM10
<a href="#">Pangea Redact</a>	Proprietary	<a href="#">Pangea</a>	<ul style="list-style-type: none"> <li>Model And Application Interaction Security,</li> <li>Secure Coding Practices</li> </ul>	LLM04, LLM07, LLM10
<a href="#">PurpleLlama CodeShield</a>	Open Source	<a href="#">Meta-PurpleLlama</a>	<ul style="list-style-type: none"> <li>Insecure Code Generation</li> </ul>	LLM02
<a href="#">Pangea Data Guard</a>	Proprietary	<a href="#">Pangea</a>	<ul style="list-style-type: none"> <li>Model And Application Interaction Security,</li> <li>Secure Coding Practices</li> </ul>	LLM04, LLM07, LLM10
<a href="#">Pangea Prompt Guard</a>	Proprietary	<a href="#">Pangea</a>	<ul style="list-style-type: none"> <li>Model And Application Interaction Security,</li> <li>Secure Coding Practices</li> </ul>	LLM01, LLM03
<a href="#">Cisco AI Validation</a>	Proprietary	<a href="#">Cisco Systems</a>	<ul style="list-style-type: none"> <li>Adversarial Input Detection,</li> <li>AI/LLM Secure Posture Management</li> <li>Final Security Audit Incident Simulation</li> <li>LLM Benchmarking</li> </ul>	LLM01, LLM03, LLM04, LLM06, LLM09
<a href="#">Mend AI</a>	Proprietary	<a href="#">Mend.io</a>	<ul style="list-style-type: none"> <li>LLM &amp; App Vulnerability Scanning</li> <li>Model And Application Interaction Security</li> <li>SAST/DAST/IAST</li> <li>Secure Coding Practices</li> <li>Secure Library/Code Repository</li> <li>Software Composition Analysis</li> </ul>	LLM01, LLM02, LLM03, LLM04, LLM06, LLM07, LLM08, LLM09, LLM10



TEST AND EVALUATION				
Solution	Type	Project/Company	Gen AI/LLMSecOps	Top 10 for LLM Risk Coverage
<a href="#">LLM Vulnerability Scanner</a>	Open Source	<a href="#">Garak.AI</a>	<ul style="list-style-type: none"> <li>• LLM Vulnerability Scanning</li> </ul>	LLM01
<a href="#">Prompt Foo</a>	Open Source	<a href="#">Prompt Foo</a>	<ul style="list-style-type: none"> <li>• Adversarial Testing</li> <li>• Bias and Fairness Testing</li> <li>• Final Security Audit</li> <li>• LLM Benchmarking</li> <li>• Penetration Testing</li> <li>• SAST/DAST/IAST</li> <li>• Vulnerability Scanning</li> </ul>	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10
<a href="#">Modelscan</a>	Open Source	<a href="#">Protect AI</a>	<ul style="list-style-type: none"> <li>• Penetration Testing Vulnerability Scanning</li> </ul>	LLM03, LLM06, LLM10
<a href="#">CyberSecEval</a>	Open Source	<a href="#">Meta</a>	<ul style="list-style-type: none"> <li>• Adversarial Testing</li> <li>• LLM Benchmarking</li> <li>• Vulnerability Scanning</li> </ul>	LLM01, LLM02, LLM07, LLM08, LLM09, LLM10
<a href="#">Cisco AI Validation</a>	Proprietary	<a href="#">Cisco Systems</a>	<ul style="list-style-type: none"> <li>• Adversarial Input Detection,</li> <li>• AI/LLM Secure Posture Management</li> <li>• Final Security Audit</li> <li>• Incident Simulation</li> <li>• LLM Benchmarking</li> </ul>	LLM01, LLM03, LLM04, LLM06, LLM09
<a href="#">Enkrypt AI</a>	Proprietary	<a href="#">Enkrypt AI</a>	<ul style="list-style-type: none"> <li>• Adversarial Testing</li> <li>• Bias And Fairness Testing,</li> <li>• Final Security Audit</li> <li>• Incident Simulation</li> <li>• LLM Benchmarking</li> <li>• Penetration Testing</li> <li>• Response Testing</li> <li>• SAST/DAST/IAST</li> <li>• Vulnerability Scanning</li> </ul>	LLM01, LLM02, LLM03, LLM04, LLM06, LLM07, LLM08, LLM09, LLM10
<a href="#">Harmbench</a>	Open Source	<a href="#">Harmbench</a>	<ul style="list-style-type: none"> <li>• Adversarial Testing</li> <li>• Bias And Fairness Testing</li> <li>• Incident Simulation</li> <li>• LLM Benchmarking</li> <li>• Response Testing</li> <li>• Vulnerability Scanning</li> </ul>	LLM01, LLM02, LLM03, LLM06, LLM08, LLM09
<a href="#">Aqua Security</a>	Proprietary	<a href="#">Aqua Security</a>	<ul style="list-style-type: none"> <li>• Adversarial Attack Protection</li> <li>• SAST/DAST/IAST</li> <li>• Secure CI/CD Pipeline</li> <li>• Secure Library/Code Repository</li> </ul>	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10

			<ul style="list-style-type: none"> <li>• Software Composition Analysis</li> <li>• Vulnerability Scanning</li> </ul>	
<a href="#">Prompt Fuzzer</a>	Open Source	<a href="#">Prompt Security</a>	<ul style="list-style-type: none"> <li>• Adversarial Testing,</li> <li>• Bias And Fairness Testing,</li> <li>• Incident Simulation,</li> <li>• Response Testing</li> </ul>	LLM01, LLM02, LLM03, LLM06
<a href="#">Pillar Security</a>	Proprietary	<a href="#">Pillar Security</a>	<ul style="list-style-type: none"> <li>• Adversarial Testing,</li> <li>• LLM Benchmarking,</li> <li>• Penetration Testing</li> </ul>	LLM01, LLM02, LLM04, LLM06, LLM07, LLM08
<a href="#">ZenGuard AI</a>	Proprietary	<a href="#">ZenGuard AI</a>	<ul style="list-style-type: none"> <li>• Adversarial Attack Protection,</li> <li>• Adversarial Testing,</li> <li>• Automated Vulnerability Scanning,</li> <li>• Data Leakage Protection,</li> <li>• LLM Guardrails,</li> <li>• Penetration Testing,</li> <li>• Privacy, Prompt Security,</li> <li>• Secure Output Handling</li> </ul>	LLM01, LLM02, LLM04, LLM05, LLM06, LLM07, LLM08, LLM10
<a href="#">Giskard</a>	Open Source	<a href="#">Giskard</a>	<ul style="list-style-type: none"> <li>• Adversarial Testing,</li> <li>• Bias and Fairness Testing</li> <li>• LLM Benchmarking,</li> <li>• Vulnerability Scanning</li> </ul>	LLM01, LLM02, LLM06, LLM08, LLM09

RELEASE				
Solution	Type	Project/Company	Gen AI/LLMSecOps	Top 10 for LLM Risk Coverage
<a href="#">Cisco AI Validation</a>	Proprietary	<a href="#">Cisco Systems</a>	Model Security Posture Evaluation, Secure Supply Chain Verification	LLM01, LLM03, LLM04, LLM05, LLM06, LLM09
<a href="#">CycloneDX</a>	Open Source	<a href="#">CycloneDX</a>	LLM/ML BOM Generation	LLM05
<a href="#">Aqua Security</a>	Proprietary	<a href="#">Aqua Security</a>	<ul style="list-style-type: none"> <li>SAST, DAST &amp; IAST</li> <li>Secure Library/Code Repository</li> <li>Software Composition Analysis</li> <li>Secure Library/Code Repository</li> </ul>	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10
<a href="#">Legit Security - AI-SPM</a>	Proprietary	<a href="#">Legit Security</a>	AI Generated Code Detection	LLM05

DEPLOY				
Solution	Type	Project/Company	Gen AI/LLMSecOps	Top 10 for LLM Risk Coverage
<a href="#">Cisco AI Runtime</a>	Proprietary	<a href="#">Cisco Systems</a>	LLM Enabled Web Application Firewall, User and Data Privacy Protections	LLM01, LLM02, LLM04, LLM06, LLM07, LLM08, LLM09, LLM10
<a href="#">PurpleLlama CodeShield</a>	Open Source	Meta		LLM02

OPERATE				
Solution	Type	Project/Company	Gen AI/LLMSecOps	Top 10 for LLM Risk Coverage
<a href="#">LLM Guard</a>	Open Source	Protect AI	Privacy, Data Leakage Protection Prompt Security, Adversarial Attack Protection	

<a href="#">Aqua Security</a>	Proprietary	<a href="#">Aqua Security</a>	Adversarial Attack Protection, Adversarial Testing, Automated Vulnerability Scanning, Data Leakage Protection, LLM Guardrails, Penetration Testing, Privacy, Prompt Security, Secure Output Handling	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10
<a href="#">ZenGuard AI</a>	Proprietary	<a href="#">Zenguard.ai</a>	Adversarial Attack Protection, Automated Vulnerability Scanning, LLM Guardrails, Privacy Data Leakage Protection Prompt Security Secure Output Handling	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10, LLM_All
<a href="#">AI Blue Team</a>	Proprietary	<a href="#">NRI SecureTechnologies</a>	Adversarial Attack Protection, LLM Guardrails, LLM Incident Detection and Response, Privacy , Data Leakage Protection, Prompt Security, Secure Output Handling	LLM01, LLM02, LLM04, LLM06, LLM08, LLM09
Cisco AI Runtime		<a href="#">Cisco Systems</a>	Adversarial Attack Protection, LLM Guardrails, LLM Incident Detection and Response, Privacy , Data Leakage Protection, Prompt Security, Runtime Application Self-Protection, Secure Output Handling	LLM01, LLM02, LLM04, LLM06, LLM07, LLM08, LLM09, LLM10

MONITOR				
Solution	Type	Project/Company	Gen AI/LLMSecOps	Top 10 for LLM Risk Coverage
<a href="#">Cisco AI Validation</a>	Proprietary	<a href="#">Cisco Systems</a>	Adversarial Input Detection, Model Behavior Analysis, AI/LLM Secure Posture Management, Regulatory Compliance Tracking	LLM01, LLM03, LLM04, LLM05, LLM06, LLM09
<a href="#">AI-Sec Platform</a>	Proprietary	<a href="#">Hidden Layer</a>	Adversarial Input Detection, Model Behavior Analysis, AI/LLM Secure Posture Management, Regulatory Compliance Tracking, Security Alerting, User Activity Monitoring, Observability, Data Privacy and Protection	LLM01, LLM02, LLM04, LLM05, LLM06, LLM07, LLM08, LLM10
<a href="#">Aqua Security</a>	Proprietary	<a href="#">Aqua Security</a>	AI/LLM Secure Posture Management	LLM04, LLM06, LLM10
<a href="#">SPLX.AI</a>	Proprietary	<a href="#">Brand Engagement Networks</a>	Adversarial Input Detection, AI/LLM Secure Posture Management, Regulatory Compliance Tracking, Security Metrics Collection, Observability, Data Privacy and Protection, Ethical Compliance	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10
<a href="#">PromptGuard</a>	Open Source	<a href="#">Meta</a>	Model Behavior Analysis	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10
<a href="#">Lakera</a>	Proprietary	<a href="#">Lakera</a>	Adversarial Input Detection, Regulatory Compliance Tracking, Security Alerting, Security Metrics Collection, Data Privacy and Protection, Ethical Compliance	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10

GOVERN				
Solution	Type	Project/Company	Gen AI/LLMSecOps	Top 10 for LLM Risk Coverage
<a href="#">Lasso Secure Gateway for LLMs</a>	Proprietary	<a href="#">Lasso Security</a> (Silver Sponsor)	<ul style="list-style-type: none"> <li>LLM Secure Gateway</li> </ul>	LLM01, LLM02
<a href="#">AI Security &amp; Governance</a>	Proprietary	<a href="#">Securiti</a> (Silver Sponsor)	<ul style="list-style-type: none"> <li>Model Discovery</li> <li>Model Risk Management</li> </ul>	LLM03, LLM06, LLM09
<a href="#">Cisco AI Validation</a>	Proprietary	<a href="#">Cisco Systems</a>	<ul style="list-style-type: none"> <li>Compliance Management,</li> <li>Risk Assessment and Management</li> </ul>	LLM01, LLM03, LLM04, LLM06, LLM09
<a href="#">AI Verify</a>	Open Source	<a href="#">AI Verify Foundation</a>	<ul style="list-style-type: none"> <li>Bias and Fairness Oversight</li> <li>Risk Assessment and Management</li> </ul>	LLM03, LLM06, LLM09
<a href="#">Prompt Security</a>	Proprietary	<a href="#">Prompt Security</a>	Bias and Fairness Oversight, Compliance Management, Data Security Posture Management, Incident Governance, Risk Assessment and Management, User/Machine Access audits	LLM01, LLM02, LLM03, LLM04, LLM05, LLM06, LLM07, LLM08, LLM09, LLM10

# Acknowledgements

## Lead Authors

Scott Clinton  
Ads Dawson  
Jason Ross  
Heather Linn

## Contributors

Andy Smith  
Arun John  
Aurora Starita  
Bryan Nakayama  
Dennys Pereira  
Emmanuel Guilherme  
Fabrizio Cilli  
Garvin LeClaire  
Helen Oakley  
Ishan Anand  
Jason Ross  
Marcel Winandy  
Markus Hupfauer  
Migel Fernandes  
Mohit Yadav  
Rachel James  
Rico Komenda  
Talesh Seeparsan  
Teruhiro Tagomori  
Todd Hathaway  
Ron F. Del Rosario  
Vaibhav Malik

## Reviewers

Andy Smith  
Arun John  
Aurora Starita  
Blanca Rivera Campos  
Bryan Nakayama  
Dan Guido  
Dennys Pereira  
Emmanuel Guilherme  
Fabrizio Cilli  
Garvin LeClaire  
Heather Linn  
Helen Oakley  
Ishan Anand  
Jason Ross  
Joshua Berkoh  
Krishna Sankar  
Marcel Winandy  
Markus Hupfauer  
Migel Fernandes  
Mohit Yadav  
Rachel James  
Rammohan Thirupasur  
Rico Komenda  
Rammohan Thirupasur  
Talesh Seeparsan  
Teruhiro Tagomori  
Todd Hathaway  
Ron F. Del Rosario  
Vaibhav Malik

## OWASP Top 10 for LLM Project Sponsors

We appreciate our Project Sponsors, funding contributions to help support the objectives of the project and help to cover operational and outreach costs augmenting the resources the OWASP.org foundation provides. The OWASP Top 10 for LLM and Generative AI Project continues to maintain a vendor neutral and unbiased approach. Sponsors do not receive special governance considerations as part of their support. Sponsors do receive recognition for their contributions in our materials and web properties.

All materials the project generates are community developed, driven and released under open source and creative commons licenses. For more information on becoming a sponsor [Visit the Sponsorship Section on our Website](#) to learn more about helping to sustain the project through sponsorship.

### Silver Sponsors



Sponsor list, as of publication date. Find the full sponsor [list here](#).



## References

- Andreesen/Horowitz. (n.d.). Emerging architectures for LLMs. A16Z.  
<https://a16z.com/emerging-architectures-for-llm-applications/>
- Databricks. (n.d.). LLM architecture. Google Drive.  
[https://drive.google.com/file/d/166D\\_Pyt3iDu18xGl3qAoMza0cq-Y52AX/view?usp=drive\\_link](https://drive.google.com/file/d/166D_Pyt3iDu18xGl3qAoMza0cq-Y52AX/view?usp=drive_link)
- Protect AI. (n.d.). What is in AI Zeroday? Protect AI Blog.  
<https://protectai.com/blog/what-is-in-ai-zeroday>
- Insight Partners. (n.d.). LLMops & MLOps: What you need to know. Insight Partners.  
<https://www.insightpartners.com/ideas/llmops-mlops-what-you-need-to-know/>
- Software Engineering Institute. (n.d.). Application of large language models (LLMs) in software engineering: Overblown hype or disruptive change? SEI Insights.  
<https://insights.sei.cmu.edu/blog/application-of-large-language-models-llms-in-software-engineering-overblown-hype-or-disruptive-change/>
- Salesforce. (2023, August 3). SDLC for prompts: The next evolution in enterprise AI development. Salesforce DevOps.  
<https://salesforcedevops.net/index.php/2023/08/03/sdlc-for-prompts-the-next-evolution-in-enterprise-ai-development/>
- Valohai. (n.d.). LLMops: Everything you need to know. Valohai Blog.  
<https://valohai.com/blog/llmops/>
- Smart Bridge. (n.d.). AI done right: Streamline development & boost value with LLMops. Smart Bridge.  
<https://smartbridge.com/ai-done-right-streamline-development-boost-value-llmops/>
- Neptune AI. (n.d.). MLOps tools & platforms landscape. Neptune AI Blog.  
<https://neptune.ai/blog/mlops-tools-platforms-landscape>
- IBM. (n.d.). All the Ops: DevOps, DataOps, MLOps, and AIOps. IBM Developer.  
<https://developer.ibm.com/articles/all-the-ops-devops-dataops-mlops-and-aiops/>
- Arxiv. (2024). A comprehensive study on large language models and their security risks. Arxiv.  
<https://arxiv.org/abs/2406.10300>
- Cloud Security Alliance. (n.d.). CSA large language model (LLM) threats taxonomy. Cloud Security Alliance.  
<https://cloudsecurityalliance.org/artifacts/csa-large-language-model-llm-threats-taxonomy>
- Sapphire Ventures. (n.d.). GenAI infra startups. LinkedIn.  
[https://www.linkedin.com/posts/sapphirevc\\_genai-infra-startups-activity-7186724761400442883-Xt3D](https://www.linkedin.com/posts/sapphirevc_genai-infra-startups-activity-7186724761400442883-Xt3D)
- AIMultiple. (n.d.). LLM security tools. AIMultiple.  
<https://research.aimultiple.com/llm-security-tools/>

## Project Supporters

Project supporters lend their resources and expertise to support the goals of the project.

HADESS  
KLAVAN  
Precize  
AWS  
Snyk  
Astra Security  
AWARE7 GmbH  
iFood  
Kainos  
Aigos  
Cloud Security Podcast  
Trellix  
Coalfire  
HackerOne  
IBM  
Bearer  
Bit79  
Stackarmor  
Cohere  
Quiq  
Lakera  
Credal.ai  
Palosade  
Prompt Security  
NuBinary  
Baltix  
SAFE Security  
BeDisruptive  
Preamble  
Nexus

PromptArmor  
Exabeam  
Modus Create  
IronCore Labs  
Cloudsec.ai  
Layerup  
Mend.io  
Giskard  
BBVA  
RHITE  
Praetorian  
Cobalt  
Nightfall AI